Recent Advances in Science and Engineering
Web page info: https://rase.yildiz.edu.tr
DOI: 10.14744/rase.2023.0002

Research Article

# Tiny machine learning model for obstacle detection with multi-zone time of flight sensors

**Sergen Erkan ARSLAN[1,*], Bülent BOLAT[2]**

[1]Arçelik A.Ş. R&D Center, Sensor Technologies, Çayırova, İstanbul, Türkiye
[2]Faculty of Electrical and Electronics Engineering Yıldız Technical University, İstanbul, Türkiye

## ARTICLE INFO

*Article history*
Received: November 17, 2023
Accepted: November 23, 2023

**Keywords:**
Time of Flight, multi zone, tiny ML, CNN, obstacle detection, autonomous robots.

## ABSTRACT

Obstacle detection is a critical research area for autonomous robots. It is especially important to detect small obstacles that can get entangled inside the robot. In addition, it can provide input to the movement and safety algorithm of autonomous devices by performing not only obstacle detection but also cliff detection. In this study, a tiny machine learning (TinyML) model that can run on a low-memory microcontroller and detect obstacles using multi-zone time-of-flight (ToF) sensors from STMicroelectronics is proposed. The proposed method applied on an ST ARM based development kit. The object detection model achieved a high accuracy of over 90% on 5 different locations and obstacle presences.

**Cite this article as:** Arslan SE, Bolat B. Tiny machine learning model for obstacle detection with multi-zone time of flight sensors. Recent Adv Sci Eng 2023;3:2:9–14.

## INTRODUCTION

The number and importance in our lives of autonomous robots with many sensors around us is increasing day by day. As the robot market grows, autonomous robot research is becoming more active than ever. Most autonomous robots use lidar, time-of-flight (ToF) sensor, ToF camera, depth camera, wide-angle camera, IR optical sensors and electromechanical switches to detect obstacles and perform navigation [1-5]. Camera based sensor systems require a great amount of processing power and memory. Therefore, such systems have greater costs. On the other hand, simpler applications have high error rates and less performance. Single point-based sensor structures are not suitable for detecting small obstacles. Especially cables and toys are very difficult to detect for these kind of sensors.

To overcome these issues, it is possible to use high-resolution sensors such as cameras to detect obstacles on the robot's driving route. However, in addition to processing power and cost constraints, cameras have privacy issues. For these reasons, the ToF sensors became popular in recent years [6].

In a real application, the robot must detect obstacles quickly and in real time during its movement. Depending on the location and distance of the detected obstacle, the robot must move away from the obstacle with a different route. By using traditional methods, there problems require a relatively large amount of computational power and resources. To overcome these, tiny machine learning (TinyML), which is optimized for structures with small memory and low processing-power microcontrollers systems, was used.

***Corresponding author.**
*E-mail address: sergenerkan.arslan@arcelik.com

Published by Yıldız Technical University Press, İstanbul, Türkiye

TinyML focuses on running compressed and optimized machine learning models on small platforms [7]. In recent years, the number of neural networks, platforms, and hardware it supports has increased [8]. TinyML has been successfully applied in various application areas such as healthcare, agriculture, industrial IoT, etc. [7].

This study was carried out with a data set of 18 obstacles of different sizes, with the presence of the obstacle and 5 positions (6 classes) during the straight motion of the robot. The proposed model achieved both a memory of less than 100kByte and an accuracy of approximately 90% for 6 classes. And it is optimized for ARM Cortex-M microcontroller (STM32F746NG evolution board [13]).

## LITERATURE REVIEW

There are many contactless activation, gesture recognition, content management and home applications specific to the multi-zone ToF sensor. Some of these application examples are urinal toilet flush, tap, dispensers, smart switch, public screen wake-up applications, gesture recognition, industrial robots, production lines, user friendly UI applications, content management, coffee machine liquid measurement, tanks, vending machines, ATM, lockers etc.

TinyML studies specifically designed for multi-zone ToF sensors are available in the literature. They are mostly gesture-based applications. For example, it is possible to perform gesture recognition using the small Temporal Convolutional Networks model [9]. There are motion modules provided by manufacturers [10] and applications to detect motion type [11]. In another work, an application for ground type detection was proposed. In this work, hard and soft ground output is produced using ToF sensors mounted on a robot to feed the neural network model [12]. Another study, there is an object detection study using a 64x32 high-speed single-photon ToF image sensor. In this study, the number of frames is very large and there is no memory constraint in the design [22]. Although ToF sensors widely used, publications on the use of ToF sensors on obstacle detection problem are limited.

## METHODOLOGY

Tiny Machine Learning (TinyML) is a rapidly expanding field of constrained hardware applications of machine learning technologies. It covers a wide range of components, including algorithms, hardware infrastructure, data analytics, direct in sensor-connect devices, low power consumption and superior software capabilities [14, 15]. Additionally, low-power machine learning in MCU-class hardware has the potential to increase efficiency and achieve significant reductions in carbon emissions in various industries [16].

There are many platforms available for developing TinyML applications such as edge impulse [17], uTensor (ARM), TensorFlowLite, STM32Cube.AI, NanoEdge AI Studio, emlearn. The hardware and framework supported by these platforms' examples may vary. In this study, edge impulse was determined as the platform. This platform is used Tensor Flow Lite Micro (TFLM) framework.

The workflow for TinyML on the Edge impulse platform is shown in Fig. 1. First, the collected data builds a model using standard tensor flow machine learning methods through training and testing. In the second part, the memory used by TensorFlow is pruned, quantized and optimized. The pruning method starts with training the network and then selecting key connections by positioning weights greater than a certain threshold. The weights below this level are eliminated, resulting in a clipped model. This new model may reduce the accuracy but reduces memory and model complexity. The quantization method is used to reduce the precision of the weights and activations from 32-bit floating-point numbers to 8-bit numbers. Symmetric and asymmetric quantization methods are used in this study [18]. In the asymmetric method, the scaled value ranges are variable. In the symmetric method, value ranges are fixed. These methods can reduce the memory space required to store network parameters by 20% to 30%. As the third step, the pruned and qualified model is converted from multidimensional arrays to one-dimensional arrays. Finally, the C/C++ model is created so that it can be embedded in the system containing the MCU.

For obstacle detection, the VL53L7 sensor from STMicroelectronics, which has an 8x8 area and a 90° viewing angle, was used as input data to the TinyML model. This sensor allows choosing between 4x4 or 8x8 individual zones for precise distance measurements and can measure up to 350cm. Additionally, the frame rate can be programmed in the range of 60Hz - 15Hz. Last, it gives range sigma, distance, reflectance, and SPAD signal values for 8x8(64) pixels, as shown in Fig 2.

This multi-featured sensor allows it to be used in different applications. In addition, this diversity of data provides a suitable infrastructure for machine learning. (Fig. 2.) In this work an STM32F746NG evolution board was used to collect data from the sensor [13]. During the measurements, sensor data was read at 30Hz frame rate by using the C library shared by ST company [21].


**Figure. 1.** Workflow for TinyML.

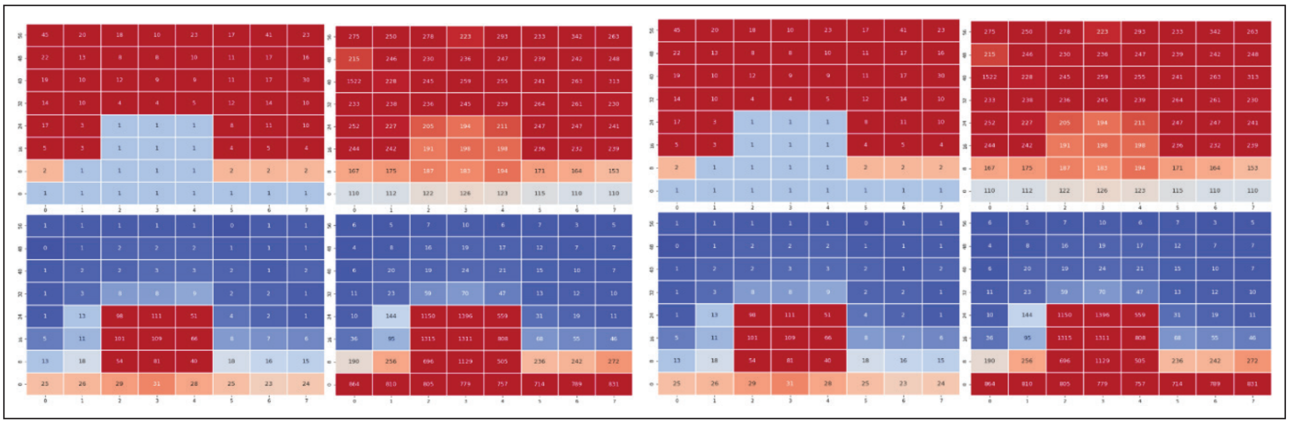

**Figure 2.** Color maps examples of range sigma, distance, reflection and SPAD values for 8x8 pixels for 50mm cube object.

**Obstacle Detection Setup with ToF sensors**

The aim of this study is to create and implement a model that can fit in low memory, produce high accuracy output, and execute in real time. To detect obstacles, the first step is to perform tests involving different obstacles and their locations. The tests were carried out in 5 different positions in front of the moving robot as shown in Figure 3. Additionally, obstacle-free tests were also performed. The data were data were collected multiple times at different locations with the mobile robot (figure 3) by using 18 different small obstacles seen in figure 4. The collected real-time data was prepared for training by labeling locations and obstacles.

**Tiny ML CNN model**

To build the TinyML model, low resolution ToF applications were examined [10]-[12] and it is seen that Convolutional Neural Network (CNN) was frequently used in the literature. Therefore, CNN was used in this study. First, a data set consisting of 6 outputs containing object detection and location via the mobile robot was collected. The data set contains 64 pixels (Fig. 5) where each pixel consisting of 4 features (range sigma, distance, reflectance, SPAD). There are 256 data in total for one frame. The

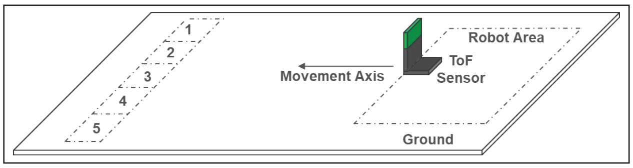


**Figure 3.** Position areas of objects for training data.

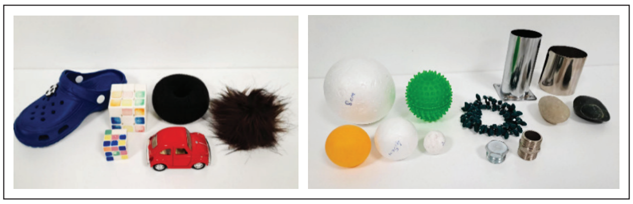

**Figure 4.** Obstacles included in the data set.

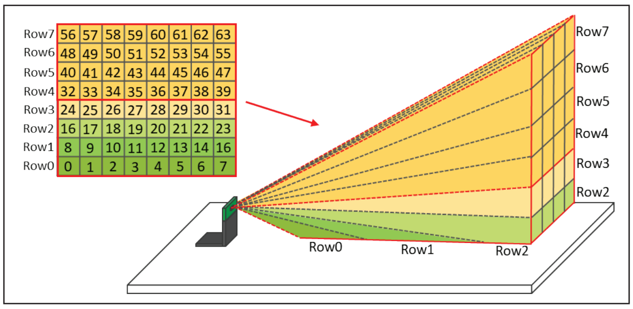


**Figure 5.** Sensor FoV pixel index and FoV for ground system.

sampling rate was determined as 30Hz. In summary, 7680 data can be collected in 1 second.

With the increase of pixels of collected data, required response time and memory increases. Hence, a reduction in data is imminent. During our experiments it was seen that the pixel values in the first 4 rows are more important for detecting the object depending on the location of ToF sensor (Fig. 5). Thus, a frame ToF data of the model is determined as 4 rows x 8 columns (32 pixels). Each pixel gives four features described above, and 10 consecutive frames forms one input sample. In this case, the data content is 32 pixels x 4 features x 10 frames. To reduce the computational complexity, which is an important limitation for small scale microprocessors, we decided to determine if there is any dominant feature which obviates the remaining. For this reason, in every trial the network fed by one feature, and the remaining features were eliminated. Hence, the total number of input data of the model was determined as 320. In the first step the input data was reshaped as a 40x8 matrix to feed the network. The second and third steps include 2D convolution and maxpooling layers. Finally, the output of maxpooling layer was flattened and applied to the dense layer. The final network architecture and code steps are shown in figure 6 and Figure 7.

To create the final model, the model with the highest accuracy rate was selected by changing the number of filters and kernel filter size of 2D convolution layers. (Fig. 6) Additionally, the learning rate was chosen as 0.0005 and

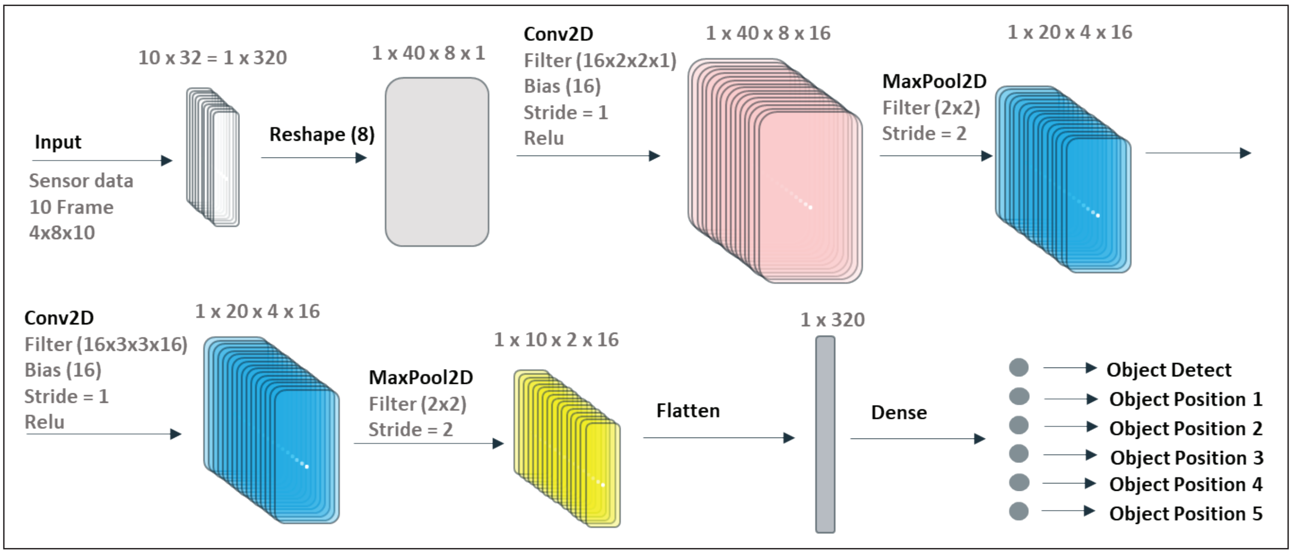


**Figure 6.** Final TinyML network architecture.

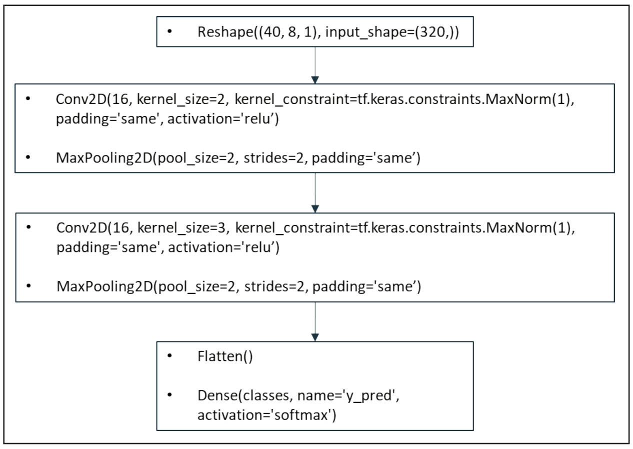


**Figure 7.** Final TinyML network code block with parameters.

epochs 30. The model was developed, trained and tested on "Edge Impulse" platform, which is a TinyML platform [19].

By using these parameters, the input features were applied to the network one by one, and the feature with the highest accuracy was determined. The final model was converted to a tiny model by using the TFLM framework. (Fig. 1)

**Table 1.** Result of features

| | Reflectance | R_Sigma | Distance | SPAD |
|---|---|---|---|---|
| Latency | | 2 ms | | |
| RAM | | 11.8Kb | | |
| Flash | | 53.5Kb | | |
| Accuracy (Int8) | 93.69% | 91.83% | 55.33% | 11.54% |
| Accuracy (Float) | 93.16% | 93.49% | 64.81% | 94.59% |

### Experimental Result

In this study, the codes were written with C/C++ programming language by using TensorFlow Lite Micro (TFLM) library on an ARM Cortex-M hardware [13]. The output of the model is the probability of 5 positions and obstacle states by the SoftMax activation function. Range sigma, distance, reflectance and SPAD values taken from ToF sensor were used as features as shown in Figure 2.

Table 1 shows the float (without quantization) and int8 (quantized) accuracy results for the features in the model shown in Fig. 2. The difference between float and int8 is the quantization process. Quantization works by reducing the precision of the model's weights, so there can often be some reduction in performance in contrast of some gain

**Table 2.** Confusion Matrix of Reflectance

| | Detect | Pos1 | Pos2 | Pos3 | Pos4 | Pos5 | Uncertain |
|---|---|---|---|---|---|---|---|
| **Detect** | 94% | 2.7% | 0.1% | 0.0% | 0.0% | 0.6% | 3.0% |
| **Pos1** | 14% | 83% | 0.0% | 0.0% | 0.0% | 0.5% | 2.8% |
| **Pos2** | 0.0% | 0.0% | 100% | 0.0% | 0.0% | 0.0% | 0.0% |
| **Pos3** | 4.1% | 0.0% | 0.0% | 95% | 0.5% | 0.0% | 0.9% |
| **Pos4** | 3.2% | 0.0% | 0.0% | 0.0% | 94% | 2.4% | 0.8% |
| **Pos5** | 0.0% | 0.0% | 0.0% | 0.0% | 1.3% | 95% | 3.3% |
| **F1_S.** | 0.96 | 0.78 | 0.98 | 0.97 | 0.96 | 0.92 | |

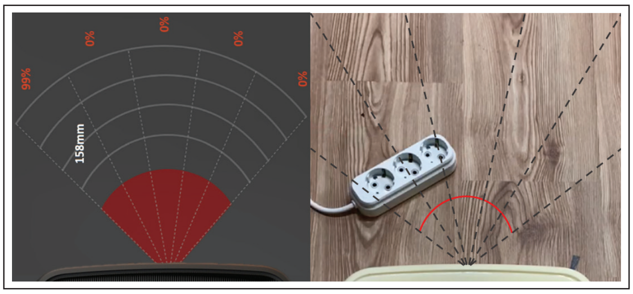


**Figure 8.** Final TinyML model realization test with robot.

in computational resources [20]. The most dramatic result seen on the table 1 is obtained by using quantized SPAD values. Natural form of SPAD values are float, therefore the quantization operation has a great amount of deterioration on information carried by SPAD. Table 2 shows the confusion matrix of the best result which is obtained by reflection feature. To help comparing the results by future works alternative performance metrics which are F1 scores and uncertain results for the 0.6 confidence threshold are also included in the Table 2.

The model was realized on a real robot platform. The model output is the position of the obstacle, but once the position information was detected, distance information can also be obtained from the sensor data. An obstacle test that was not used in model training was performed and displayed with an interface (Figure 8).

## CONCLUSION

In this paper, the model performance of the TinyML model, developed using data from the multi-zone ToF sensor while the autonomous robot is in motion, is presented on an embedded system. Model accuracies of four features taken from an 8x8 ToF sensor are compared to determine the best feature. Using the selected high-accuracy feature, the CNN model was adapted to the microcontroller with TensorFlowLite for TinyML implementation. The model is implemented on a low-power microcontroller with limited computing resources and memory to give an output in 300ms. And it has been shown that a 93% accuracy rate can be achieved. As a result, it has been shown that high accuracy obstacle detection can be implemented with a low resolution ToF sensor and a microcontroller without using a camera.

By using a similar model in the future, studies such as detecting small obstacles and creating maps can be carried out with multiple or higher resolution ToF sensors.

## AUTHORSHIP CONTRIBUTIONS

Authors equally contributed to this work.

## DATA AVAILABILITY STATEMENT

The published publication includes all graphics and data collected or developed during the study.

## CONFLICT OF INTEREST

The author declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

## ETHICS

There are no ethical issues with the publication of this manuscript.

## REFERENCES

[1] M.-C. Kang, K.-S. Kim, D.-K. Noh, J.-W. Han, and S.–J. Ko, "A robust obstacle detection method for robotic vacuum cleaners," IEEE Transactions on Consumer Electronics, Vol. 60(4), pp. 587–595, 2014. [CrossRef]

[2] K. M. Hasan, Abdullah-Al-Nahid, and K. J. Reza, "Path planning algorithm development for autonomous vacuum cleaner robots," 2014 International Conference on Informatics, Electronics & Vision (ICIEV), Dhaka, Bangladesh, 2014. [CrossRef]

[3] W. Yeon, Jeong, and K. Mu Lee, "CV-SLAM: a new ceiling vision-based SLAM technique," 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, Edmonton, pp. 3195–3200, AB, Canada, 2005.G.W. Marcy, Detection of Extrasolar Giant Planets, Astrophysics 1998. [CrossRef]

[4] W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1271–1278, Stockholm, Sweden, 2016. J. Devor, D. Charbonneau, Using MECI to Eclipsing Binaries from Photometric Exoplanet Surveys, Harvard, Semithsonian Center of Astrophysics. [CrossRef]

[5] H. A. S. H. Prayash, M. R. Shaharear, M. F. Islam, S. Islam, N. Hossain, and S. Datta, "Designing and Optimization of An Autonomous Vacuum Floor Cleaning Robot," 2019 IEEE International Conference on Robotics, Automation, Artificial-intelligence and Internet-of-Things (RAAICON), pp. 25-30, Dhaka, Bangladesh, 2019. [CrossRef]

[6] STMicroelectronics, "https://www.st.com/resource/en/datasheet/vl53l7cx.pdf" VL53L7CX datasheet, September 27, 2023.

[7] Y. Abadade, A. Temouden, H. Bamoumen, N. Benamar, Y. Chtouki, and A. S. Hafid, "A Comprehensive Survey on TinyML," in IEEE Access, Vol. 11, pp. 96892–96922, 2023. [CrossRef]

[8] V. Janapa Reddi, B. Plancher, S. Kennedy, L. Moroney, P. Warden, A. Agarwal, C. Banbury, M. Banzi, M. Bennett, B. Brown, S. Chitlangia, R. Ghosal, S. Grafman, R. Jaeger, S. Krishnan, M. Lam, D. Leiker, C. Mann, M. Mazumder, D. Pajak, D. Ramaprasad, J. Evan Smith, M. Stewart, D. Tingley, "Widening Access to Applied Machine Learning with TinyML," ArXiv, 2106.04008, 2021. [CrossRef]

[9] S. Boner, C. Vogt, and M. Magno, "Tiny TCN model for Gesture Recognition With Multi-point Low power ToF-Sensors," 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS), Incheon, Korea, Republic of, pp. 356–359, 2022. [CrossRef]

[10] Turnkey gesture recognition solution based on VL53L5CX, VL53L7CX and VL53L8CX multi-zone Time-of-Flight ranging sensors, https://www.st.com/en/embedded-software/stsw-img035.html" Accessed on Sep 27, 2023.

[11] 8x8 ToF Gesture Classification - Arduino RP2040 Connect, https://docs.edgeimpulse.com/experts/novel-sensor-projects/tof-gesture-classification-arduino-rp2040-connect Accessed on Sep 27, 2023.

[12] STM32. "Floor type detection for vacuum cleaners," https://stm32ai.st.com/use-case/floor-type-detection-for-vacuum-cleaners/ September 27, 2023.

[13] STMicroelectronics, "32F746GDISCOVERY," https://www.st.com/en/evaluation-tools/32f746g-discovery.html Accessed on September 27, 2023.

[14] M. Shafique, T. Theocharides, V. J. Reddy, and B. Murmann, "TinyML: Current Progress, Research Challenges, and Future Roadmap," 2021 58th ACM/IEEE Design Automation Conference (DAC), pp. 1303–1306, San Francisco, CA, USA, 2021. [CrossRef]

[15] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," Journal of King Saud University - Computer and Information Sciences, pp. 1595–1623, 2022. [CrossRef]

[16] S. Prakash, M. Stewart, C. Banbury, M. Mazumder, P. Warden, B. Plancher, and V. Janapa Reddi, "Is TinyML sustainable?" ArXiv, 2301.11899, 2023. [CrossRef]

[17] Edge Impulse, "Getting started," https://docs.edgeimpulse.com/docs/ Accessed on Sep 27, 2023.

[18] Tensor Flow Lite, "8-bit quantization specification," https://www.tensorflow.org/lite/performance/quantization_spec Accessed on Sep 27, 2023.

[19] S. Hymel, C. Banbury, D. Situnayake, A. Elium, C. Ward, M. Kelcey, M. Baaijens, M. Majchrzycki, J. Plunkett, D. Tischler, A. Grande, L. Moreau, D. Maslov, A. Beavis, J. Jongboom, and V. Janapa Reddi, "Edge Impulse: An MLOps platform for tiny machine learning," ArXiv, abs/2212.03332, 2022.

[20] Tenser Flow, "Post training quantization in tensor flow lite," https://www.tensorflow.org/lite/performance/post_training_quantization, Accessed on Oct 1, 2023.

[21] ST, "Ultra-Lite Driver (ULD) for VL53L5CX multi-zone sensor," https://www.st.com/en/embedded-software/stsw-img023.html#st-get-software Accessed on Oct 31, 2023.

[22] G. Mora-Martín, A. Turpin, A. Ruget, A. Halimi, R. Henderson, J. Leach, and I Gyongy, "High-speed object detection with a single-photon time-of-flight image sensor," Optics Express 29, 33184–33196, 2021. [CrossRef]