RESEARCH ARTICLE

A Deep Learning-Based Embedded System for Pest Bird Sound Detection and Proximity Estimation

Euhid Aman[1,*] and Hwang-Cheng Wang[2]

ABSTRACT

Cultivating crops is vital for driving economies, and maintaining agricultural fields is crucial for sustaining food production. This initiative centers on addressing the issue of pest birds, specifically starlings, within vineyards. The proposed strategy employs sound signals to detect and distinguish starling birds within the vineyard environment. Through an analysis of audio inputs from the surroundings, the system can effectively recognize unique sound patterns associated with starling birds, utilizing deep learning techniques. Furthermore, this project incorporates ultrasonic sensors for distance estimation, enabling the calculation of the bird's proximity from a fixed point within the vineyard. All of these detection and estimation processes are executed on a RP2040 microcontroller, specifically the Cortex-M0+ 133 MHz variant. Following the detection phase, an autonomous vehicle equipped with red diode lasers can be dispatched to the designated location to deter the pest birds and safeguard the vineyards from unwanted disruptions and crop losses.

Keywords: Deep learning, embedded systems, quantized neural networks, region-based detection.

1. INTRODUCTION

Crop damage inflicted by birds persists as a perennial and daunting challenge within the realm of agriculture. Conventional bird deterrent methods, such as nets and loud noises, have gradually waned in effectiveness due to birds' adaptation, exacerbating this enduring issue. Addressing this multifaceted predicament necessitates the development of bird expulsion strategies that not only possess adaptability but also real-time bird detection capabilities.

This research proposes an economical real-time bird detection solution in vineyards, utilizing microcontrollers and artificial neural networks [1]. The focal point of this endeavor lies in creating a real-time bird detection system meticulously calibrated for pinpointing starling birds within grape fields [2].

The proposed system harnesses the capabilities of the RP2040 microcontroller, esteemed for its cost-efficiency and robustness—attributes essential for large-scale deployment, particularly considering the harsh environmental conditions and extreme temperatures that pervade vineyard landscapes.

This innovative system integrates an array of sensors, encompassing SparkFun MicroMod Machine Learning Carrier Board's sound sensors for sound detection and HC-SR04 ultrasonic sensors for precise distance estimation. By perpetually scrutinizing real-time sound signals from the surrounding environment, our system adeptly discerns the unique sound patterns attributed to starling birds, employing a neural network model. Upon bird detection, the system ascertains the avian intruder's exact location, promptly relaying this information to a central server system. This seamless communication mechanism facilitates the expeditious implementation of effective measures to repel the avian threat.

In the intricate ecosystem of vineyards, the consequences of bird infestations transcend the superficial realm. The interconnection among grape bunches renders them susceptible to an often-underestimated vulnerability. The initiation of decay within a single grape or its infection sets a chain reaction in motion, primarily due to its tightly packed nature and susceptibility to skin cracking. As a solitary grape deteriorates, it emits chemical compounds and moisture that can rapidly infiltrate neighboring grapes.

This swift decay escalation can culminate in the entire bunch becoming tainted and unsuitable for harvest [4]. Ultimately, the quality and marketability of an entire cluster become jeopardized, potentially resulting in substantial economic losses for vineyard proprietors [5].

In summary, this research proposes novel endeavors to tackle the persistent challenge of bird damage in agriculture, especially in vineyards (Fig. 1):

- Innovative Detection System: The proposed system utilizes the RP2040 microcontroller for cost-efficiency and robustness, essential for large-scale deployment in harsh vineyard environments [6].
- Multisensory Integration: The system integrates SparkFun MicroMod Machine Learning Carrier Board's sound sensors for sound detection and HC-SR04 ultrasonic sensors for precise distance estimation.
- Deep Learning for Avian Recognition: By analyzing real-time sound signals, the system discerns unique sound patterns of starling birds using a neural network model.
- Real-time Communication and Action: Upon bird detection, the system relays the avian intruder's location to a central server, facilitating the expeditious implementation of measures to repel the avian threat.

This paper is structured into several sections to provide a comprehensive understanding of our approach. Section 2 provides an overview of related works. The necessary hardware components are presented in Section 3, and the software aspects are discussed in Section 4. The methodology for creating the deep learning-based model for bird detection, from its development to deployment in the RP2040 microcontroller, along with an explanation of the distance calculation method to the bird, is elaborated upon in Section 5. Finally, the conclusion and future research are described in Section 6. Through this work, we aim not only to protect grapevines but also to safeguard the economic viability of vineyard harvests, offering a holistic solution to a pressing agricultural issue. This paper is organized to provide a comprehensive understanding of our approach and its potential impact on vineyard management.

2. LITERATURE REVIEW

Bird damage to crops has long been a significant challenge in the agricultural sector, necessitating effective bird control strategies. Traditional methods, such as nets and loud noises, have exhibited diminishing effectiveness over time due to birds' remarkable adaptability to these deterrents [7]. This persistent issue has underscored the need for innovative and adaptable solutions that can offer real-time bird detection capabilities, particularly in the context of vineyards. One emerging approach is the utilization of microcontrollers to create cost-effective and robust solutions. The RP2040 microcontroller variant, with its cost-efficiency and durability, has gained attention as a suitable platform for large-scale deployment in challenging agricultural settings [8]. This choice of hardware reflects the increasing trend toward employing accessible and adaptable technologies in precision agriculture.

Sensors play a pivotal role in enabling real-time bird detection systems. MicroMod Machine Learning board for sound detection and HC-SR04 ultrasonic sensors for accurate distance estimation have demonstrated their effectiveness in monitoring avian activities [9]. When integrated into a comprehensive system, these sensors enable the continuous analysis of sound patterns associated with

Fig. 1. Pest bird infestation to different crops [3].

specific bird species and facilitate real-time data collection for immediate action.

The integration of deep learning techniques in bird detection systems has shown promise in improving accuracy and efficiency [10]. The use of neural network models, coupled with sensor data, has enabled systems to differentiate between various bird species, contributing to more effective bird deterrence measures.

The economic implications of bird infestations in vineyards cannot be underestimated. The tightly packed nature of grape clusters and their susceptibility to rapid decay make vineyards particularly vulnerable to avian threats. The consequences can extend beyond immediate crop loss, affecting the overall quality and marketability of the harvest. Therefore, the development of comprehensive solutions that address both economic and agricultural concerns is of paramount importance.

In light of these challenges and opportunities, this research endeavors to provide an innovative solution that harnesses microcontroller technology, sensor integration, and deep learning techniques to create a real-time bird detection system tailored for vineyards. By addressing the multifaceted predicament of avian threats, this research aims to contribute to the preservation of grape quality and the economic sustainability of vineyard operations.

Recent research in agriculture and pest management has recognized the importance of real-time bird detection systems in safeguarding crops. These systems have shown promise in efficiently identifying and mitigating avian threats. However, the implementation of such systems remains a challenge, particularly in environments with demanding conditions like vineyards.

3. Hardware Design

3.1. SparkFun MicroMod RP2040 Processor

The SparkFun MicroMod RP2040 Processor, based on the RP2040 chipset, offers a robust microcontroller solution characterized by dual Cortex M0+ processors operating at up to 133 MHz [11]. It provides a substantial 264 kB of embedded SRAM distributed across six banks, fostering efficient memory management. This module is equipped with a comprehensive set of peripherals, including USB, UART, I2C, SPI, GPIO pins, analog inputs, and PWM channels, making it suitable for diverse applications. Its support for USB 1.1 Host/Device functionality enhances connectivity options. Moreover, it accommodates popular programming languages like MicroPython and C/C++, facilitating versatile software development. With additional features like real-time counters, timers, and status LEDs, the SparkFun MicroMod RP2040 processor is a compelling choice for embedded systems and IoT research and development.

3.2. SparkFun MicroMod Machine Learning Carrier Board

The SparkFun MicroMod Machine Learning Carrier Board is a versatile and feature-rich platform that supports machine learning and artificial intelligence applications [12]. Equipped with advanced components such as digital I2C MEMS microphones, a three-axis ST LIS2DH12TR accelerometer, and a Himax camera connector, this carrier board provides comprehensive sensor capabilities for audio and vision-based tasks. With USB-C connectivity, a Qwiic connector for seamless peripheral integration, and a MicroSD socket for data storage, it offers a robust and flexible solution for a wide range of machine-learning applications. Additionally, the inclusion of a lithium battery for real-time clock functionality ensures reliability in time-sensitive applications, making the SparkFun Micro-Mod Machine Learning Carrier Board an ideal choice for research and development in the field of machine learning.

3.3. Ultrasonic Distance Sensor-HC-SR04

The HC-SR04 is a widely used ultrasonic distance sensor module known for its simplicity and effectiveness in measuring distances [13]. This sensor operates by emitting ultrasonic pulses and then measuring the time it takes for the sound waves to bounce back after hitting an object. With its low cost and ease of use, the HC-SR04 is a popular choice in robotics, automation, and various areas, providing accurate distance measurements within a range of a few centimeters to several meters. Its versatility and reliability make it a valuable tool for applications such as obstacle detection, proximity sensing, and even water level measurement in certain contexts.

4. Software Design

4.1. TensorFlow Lite for Microcontrollers and Keras

TensorFlow Lite for Microcontrollers (TFLite Micro) is a compact framework created for microcontrollers and other devices with limited resources [14]. It offers a collection of tools and frameworks for deploying and optimizing TensorFlow models on compact hardware.

Using TensorFlow as its backend, Keras, a high-level neural network API written in Python, may be applied. For deep learning academics and practitioners, Keras offers an intuitive interface that streamlines the building, training, and deployment of neural networks. A version of Keras that is included in TFLite Micro enables programmers to create and train neural networks in Python before exporting them in a format that can be used by TFLite Micro for deployment on microcontrollers. This makes it possible for developers to create models and deploy them on devices with limited resources using the well-known Keras interface.

5. Pest Bird Detection

The sound detection methodology as shown in Fig. 2, follows a structured workflow to develop and deploy a custom deep-learning based sound classification model on an RP2040 microcontroller. It begins by setting up the necessary development environment, including installing required libraries, command-line tools, and ARM CMSIS software [15]. Python libraries and ARM CMSIS software are then installed to facilitate further development. Subsequently, the ESC-50 dataset is downloaded and processed,
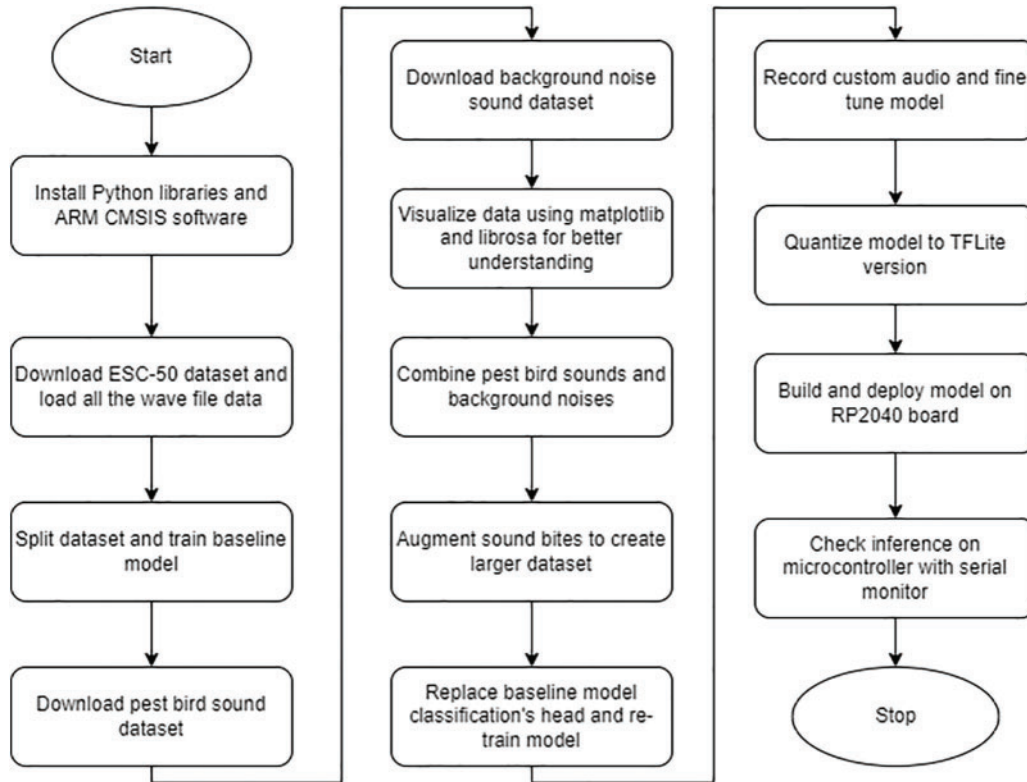
Fig. 2. Complete system with different components to monitor environmental factors, along with pest bird detection and expulsion.

and a baseline model is trained. To enhance model performance, additional datasets, including pest bird sounds and background noise, are downloaded, combined, and augmented to create a more comprehensive training dataset. The baseline model's classification head is replaced, and the model is fine-tuned to optimize its performance.

Once the model is refined, it is quantized to a TFLite version suitable for deployment on the RP2040 board. The final step involves building and deploying the model onto the microcontroller. Inference on the RP2040 is assessed through a serial monitor to ensure the model performs as expected in a resource-constrained environment. The study concludes with a thorough evaluation and validation of the deployed model's performance. This systematic methodology ensures a step-by-step approach to successfully develop, deploy, and test a custom sound classification model for real-world applications.

5.1. Sound Data Collection and Splitting

The initial baseline neural network model used for sound classification will undergo training with the ESC-50 dataset, comprising 2,000 environmental sound recordings categorized into 50 distinct classes [16]. These audio files, each lasting 5 seconds, are segmented into slices of 16,000 samples after filtering out any silent sections. Additionally, the original audio samples are strided every 4,000 samples to augment the dataset size and offer diverse samples for training. The ESC-50 dataset is organized into three subsets: training, validation, and test sets. In our methodology, we employ k-fold cross-validation. Specifically, entries with fold values less than 4 are designated for training, those with values equal to or greater than 4 but less than the total number of folds are allocated for validation, and

the remaining entries are reserved for testing. This strategic partitioning ensures a comprehensive assessment of our neural network's performance on distinct subsets of the dataset, contributing to its robustness and generalization capability. To visualize the data, as illustrated in Fig. 3, one can utilize matplotlib along with librosa's waveform and spectrogram functions [17].

5.2. Baseline Model Creation

Upon extracting the audio data features, we employ the TensorFlow Keras API to construct the model. This model is configured with accuracy as the metric, an Adam optimizer, and a loss function of sparse categorical cross-entropy. Furthermore, we have defined early stopping and a dynamic learning rate scheduler as callbacks during the training process.
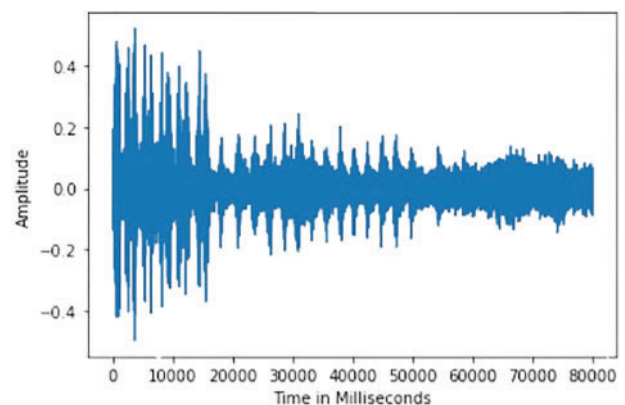

Fig. 3. Visualization of amplitude of sound wave of birds from the ESC-50 dataset over time in milliseconds.

```
Layer (type)                Output Shape       Param #
==============================================================
resizing (Resizing)         (None, 32, 32, 1)  0

normalization (Normalizatio (None, 32, 32, 1)  3
n)

conv2d (Conv2D)             (None, 13, 13, 8)   520

max_pooling2d (MaxPooling2D (None, 6, 6, 8)     0
)

flatten (Flatten)           (None, 288)         0

dropout (Dropout)           (None, 288)         0

dense (Dense)               (None, 50)          14450

==============================================================
Total params: 14,973
Trainable params: 14,970
Non-trainable params: 3
```

Fig. 4. Sequential 8-layer baseline model for audio classification.

With these settings of a total of 14,973 parameters, as depicted in Fig. 4, the model achieved a 39% loss and an accuracy of 24.44%. The low accuracy of the model, is due to the vastness of the ESC-500 dataset. The high loss and low accuracy of the baseline model does not have any impact on the overall project, as it will be refined further for sound categorization tasks.

5.3. Transfer Learning [18]

To ensure the model can accurately identify pest-bird sounds, a custom dataset containing mynah sounds needs to be utilized. This custom dataset should be augmented with background noises found in the TensorFlow speech commands directory (Fig. 5).

To enhance the existing mynah-sounds dataset, data augmentation techniques should be applied. This involves introducing white noise to the mynah sound clips, incorporating random periods of silence, and even merging two or more audio signals to expand the dataset. It is essential to segment the mynah sounds dataset into 1-second sound snippets and integrate them with the other background noises. This process not only increases the dataset's size but also enhances its consistency.
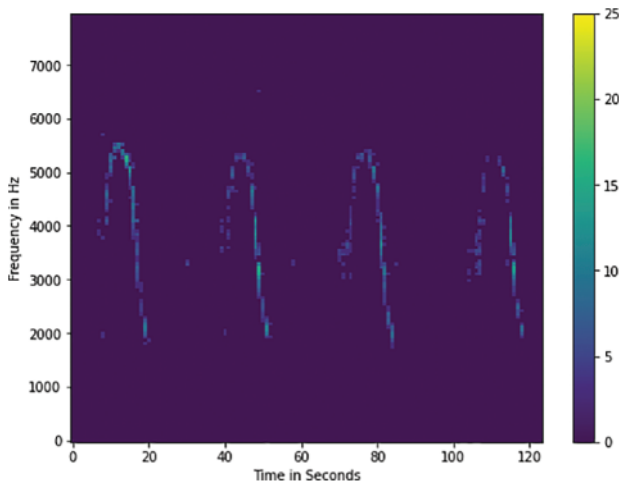
Once the dataset has been augmented and grown substantially, it should be partitioned again into training, validation, and testing subsets.

Finally, to create a binary classifier that exclusively identifies mynah sounds, the initial model's head and tail must be replaced. Subsequently, the model should be retrained using the newly augmented dataset.

5.4. Quantization-Aware Training and Model Compression

Following the completion of model training for mynah sound detection, the subsequent stage involves quantizing the model for execution on the RP2040 microcontroller. Quantization-aware training is employed to optimize the model's performance with reduced-precision computations, mirroring real deployment scenarios. This methodology results in a model capable of generating more streamlined, compact models, as shown in Fig. 6, with a total of 812 parameters.

The procedure outlined in (1) [19] initiates by ascertaining the required number of bits, denoted as m, for representing the unsigned integer part of the conversion from floating-point to fixed-point numbers:

$$m = 1 + \left\lfloor \log_2 \left(\max_{1 \leq i \leq N} |x_i| \right) \right\rfloor \tag{1}$$

where x_i represents an element within the floating-point vector x, which has a length of N. When m is a positive value, it signifies the requirement for m bits to the absolute value of the integer part. Conversely, when m is negative, it indicates that there are m leading unused bits in the fractional portion. Consequently, this approach enables the enhancement of precision for vectors containing values smaller than 21, as it permits the removal of redundant leading bits while introducing additional precision-enhancing trailing bits. This simplifies the calculation of the number of bits in the remaining fractional part, denoted as n, as follows:

$$n = w - m - 1 \tag{2}$$

where w denotes the width of the data type. When $n > 0$, it signifies that precision can be adequately conveyed using

Fig. 5. Spectrogram depicting pest bird audio augmented with white noise.

```
Layer (type)                Output Shape          Param #
=================================================================
input_1 (InputLayer)        [(None, 124, 129, 1)] 0

resizing (Resizing)         (None, 32, 32, 1)     0

normalization (Normalizatio (None, 32, 32, 1)     3
n)

conv2d (Conv2D)             (None, 13, 13, 8)     520

max_pooling2d (MaxPooling2D (None, 6, 6, 8)       0
)

flatten (Flatten)           (None, 288)           0

dropout (Dropout)           (None, 288)           0

dense_1 (Dense)             (None, 1)             289

=================================================================
Total params: 812
Trainable params: 809
Non-trainable params: 3
```

Fig. 6. Final quantized model for deployment on RP2040.

n bits. Conversely, if *n* is less than or equal to 0, it implies that the full accuracy of the integer cannot be faithfully represented.

5.5. Final Model Deployment and Inference on RP2040

The final stage involves transferring the generated model onto the RP2040 microcontroller through a process referred to as flashing. Prior to this, the model must undergo compilation and construction using the C/C++ SDK designed for the microcontroller [20]. This compilation and construction process encompasses tasks like configuring the board's LEDs for output, setting up the TFLite library and model for inference, and establishing the CMSIS-DSP based digital signal processing pipeline. Furthermore, the MicroMod Machine Learning carrier board microphone is activated to enable real-time audio input.

The deployed model yielded an accuracy of 69.07% on the validation dataset, which was deemed a commendable result for this particular task. Furthermore, the model successfully detected the distinctive sound of the mynah bird, even in the presence of background noise. This showcases the model's effectiveness in filtering out undesired ambient sounds and focusing on the intended target sound.

5.6. Distance Estimation

The proposed system employs a sound sensor for the detection of starlings or nuisance birds. Upon detecting a bird above a specified threshold, it triggers the activation of ultrasonic sensors to gauge the distance between the bird and the transmitter [21]. In our proposed system, we utilize the HC-SR04 ultrasonic sensor, which calculates the

distance by measuring the time taken for a sound wave to travel to the bird and return to the sensor.

The formula used for distance calculation is as follows:

$$s = \frac{t}{2} \times c \tag{3}$$

where *t* represents the time it takes for the sound wave to travel to the object and return to the sensor, and *c* denotes the speed of sound in the air (approximately 343 meters per second at room temperature).

To enhance accuracy, we employ multiple ultrasonic sensors positioned at various angles for precise distance estimation, as depicted in Fig. 7 [22]. To mitigate potential measurement errors, the distance estimation process is executed simultaneously multiple times, and the average of the results is computed to obtain the most precise distance to the pest bird [23].

Once the presence of a bird has been detected and its distance accurately determined, the entire dataset is transmitted to a central local server. This server then dispatches a bird interception vehicle equipped with a red diode designed to deter the birds.

6. CONCLUSION

The pest bird detection system, utilizing embedded systems with the SparkFun MicroMod RP2040 processor, MicroMod Machine Learning board, and HC-SR04 ultrasonic sensor, effectively identifies pest birds in grape fields. While leveraging continuous field sounds for detection, the system's trade-off between accuracy and rapid response due to limited RAM (264 KB) underscores the need for optimization. Real-time monitoring empowers farmers to swiftly address infestations, reducing reliance on harmful pesticides. This study highlights embedded systems' innovative potential, scalable to larger farms for diverse pest bird species detection.

Future work prioritizes enhancing detection accuracy through refined neural networks, additional sensor data, and algorithms to minimize false positives [24]. Broadening the system's scope to detect a wider range of pest bird species involves diverse sound datasets and techniques for species differentiation based on audio and movement patterns [25]. For larger operations, research into networked deployments, centralized monitoring, and remote management is essential. Energy efficiency investigations encompass low-power hardware, energy-efficient algorithms, and renewable sources [26].

Integrating with precision agriculture, real-time decision support, and long-term monitoring is crucial. User-friendly interfaces, environmental impact assessment, and cost optimization ensure sustainability and affordability for farmers [27]. This holistic approach aims to make the system a comprehensive, accessible, and sustainable solution for pest bird detection in agriculture.
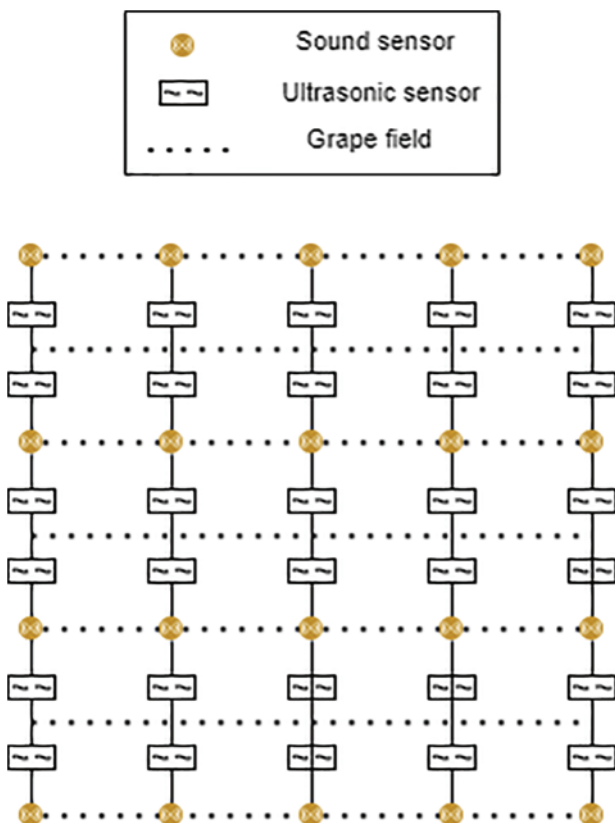
Fig. 7. Arrangement of sound and distance sensor arrays.

References

[1] Aman E, Jana S, Athikary KG, Suryanarayana RC. AI inspired ATC, based on ANN and using NLP (No. 2023-01-0985). SAE Technical Paper; 2023.

[2] Stevenson AB, Virgo BB. Damage by robins and starlings to grapes in Ontario. *Can J Plant Sci*. 1971;51(3):201–10.

[3] Bozzo F, Tarricone S, Petrontino A, Cagnetta P, Maringelli G, La Gioia G, *et al.* Quantification of the starling population, estimation and mapping of the damage to olive crops in the apulia region. *Animals*. 2021;11(4):1119.

[4] Palou L, Crisosto CH, Smilanick JL, Adaskaveg JE, Zoffoli JP. Effects of continuous 0.3 ppm ozone exposure on decay development and physiological responses of peaches and table grapes in cold storage. *Postharvest Biol Technol*. 2002;24(1):39–48.

[5] Scheck H, Vasquez S, Fogle D, Gubler W. Grape growers report losses to black-foot and grapevine decline. *Calif Agric*. 1998;52(4):19–23.

[6] Acevedo MA, Villanueva-Rivera LJ. From the field: using automated digital recording systems as effective tools for the monitoring of birds and amphibians. *Wildlife Soc Bull*. 2006;34(1):211–4.

[7] Avery ML. Birds in pest management. *Encyclopedia of Pest Management*. New York: Marcel Dekker; 2002, pp. 104–6.

[8] Boyce L, Anton DM, Sandy L. An economic analysis of bird damage in vineyards of the marlborough region. 1999.

[9] Hong SJ, Han Y, Kim SY, Lee AY, Kim G. Application of deep-learning methods to bird detection using unmanned aerial vehicle imagery. *Sens*. 2019;19(7):1651.

[10] Riya R, KR V, Sonamsi S, Jain D. Automated bird detection and repeller system using IoT devices: an insight from Indian agriculture perspective. *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*, 2020, March.

[11] Sharma PS. Programming the Pi Pico RP2040 I/O processor. Doctoral dissertation, Cornell University; 2021.

[12] Bakar A, Goel R, de Winkel J, Huang J, Ahmed S, Islam B, *et al.* Protean: an energy-efficient and heterogeneous platform for adaptive and hardware-accelerated battery-free computing. *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems*, 2022, November.

[13] Morgan EJ. HC-SR04 ultrasonic sensor. Nov. 2014.

[14] Warden P, Situnayake D. *TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers*. O'Reilly Media; 2019.

[15] Wickert MA. Using the ARM Cortex-M4 and the CMSIS-DSP library for teaching real-time DSP. *2015 IEEE Signal Processing and Signal Processing Education Workshop (SP/SPE)*. IEEE; 2015, August.

[16] Piczak KJ. ESC: dataset for environmental sound classification. *Proceedings of the 23rd ACM International Conference on Multimedia*, pp. 1015–8, 2015, October.

[17] McFee B, Raffel C, Liang D, Ellis DP, McVicar M, Battenberg E, *et al.* Librosa: audio and music signal analysis in python. *Proceedings of the 14th Python in Science Conference*, vol. 8, pp. 18–25, 2015, July.

[18] Ribani R, Marengoni M. A survey of transfer learning for convolutional neural networks. *2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T)*, pp. 47–57, IEEE; 2019, October.

[19] Novac PE, Boukli Hacene G, Pegatoquet A, Miramond B, Gripon V. Quantization and deployment of deep neural networks on microcontrollers. *Sens*. 2021;21(9):2984.

[20] Smith S, Smith S. Interacting with C and the SDK. In *RP2040 Assembly Language Programming: ARM Cortex-M0+ on the Raspberry Pi Pico*, Berkeley, CA: Apress, 2022, pp. 147–60.

[21] Mouritsen H, Heyers D, Güntürkün O. The neural basis of long-distance navigation in birds. *Annu Rev Physiol*. 2016;78:133–54.

[22] Dvorak JS, Stone ML, Self KP. Object detection for agricultural and construction environments using an ultrasonic sensor. *J Agric Saf Health*. 2016;22(2):107–19.

[23] Jiang B, Deghat M, Anderson BD. Simultaneous velocity and position estimation via distance-only measurements with application to multi-agent system control. *IEEE Trans Automat Control*. 2016;62(2):869–75.

[24] Gondchawar N, Kawitkar RS. IoT-based smart agriculture. *Int J Adv Res Comput Commun Eng*. 2016;5(6):838–42.

[25] Smith OM, Kennedy CM, Owen JP, Northfield TD, Latimer CE, Snyder WE. Highly diversified crop-livestock farming systems reshape wild bird communities. *Ecol Appl*. 2020;30(2):e02031.

[26] Pimentel D, Berardi G, Fast S. Energy efficiency of farming systems: organic and conventional agriculture. *Agric, Ecosyst Environ*. 1983;9(4):359–72.

[27] Kumar P, Nelson A, Kapetanovic Z, Chandra R. Affordable artificial intelligence–augmenting farmer knowledge with AI. 2023. arXiv preprint arXiv:2303.06049.