Low-cost Smart Raven Deterrent System with Tiny Machine Learning for Smart Agriculture

Seonyeong Heo*, Nicolas Baumann, Carla Margelisch, Marco Giordano, Michele Magno
Department of Information Technology and Electrical Engineering
ETH Zürich
Zürich, Switzerland
{seonyeong.heo, nicolas.baumann}@pbl.ee.eth.ch, carlama@ethz.ch, {marco.giordano, michele.magno}@pbl.ee.ethz.ch

***Abstract*—Smart farming is a promising application domain for the Internet of Things, which helps farmers to reduce operating costs and increase their profit with automation. Bird or raven deterrence is one example of the potential use of the Internet of Things paradigm leading to smart farming applications. Recently, drone-based approaches have demonstrated efficacy to detect and expel birds that reduce crop yield, but on the other side, those solutions are relatively high-cost options for farmers because they require expensive devices such as desktop-level computers and autonomous drones. This paper proposes a maintenance-free, energy-efficient, and low-cost smart raven deterrent system with edge computing, which runs completely standalone and self-sustainable. In particular, a tiny convolutional neural network has been proposed and optimized for multi-core microcontrollers. To demonstrate the effectiveness of the system and the neural network, the paper presents a developed prototype system with a novel hexa-core ARM Cortex-M4F platform, namely Spresense from Sony. The evaluation results show that the prototype system obtains a detection accuracy of 77% for test samples and consumes an average power of 85.1 mW.**

***Index Terms*—Smart agriculture, edge computing, tiny machine learning, embedded systems**

I. Introduction

With the growth of the Internet of Things (IoT) market, smart farming has been becoming a more promising option for large-scale farms [1]–[3]. Smart farming can help to increase crop yield and reduce operating costs by automating expensive and tedious tasks. One application of smart farming is a bird deterrent system, which is to expel birds such as ravens that may damage crops in large fields. Birds, especially ravens in Switzerland [4], are known to be troublesome in farms because they can cause a substantial reduction in crop yield [4]–[7]. Without a smart automation system, farmers have to continuously monitor their fields and manually deter birds from eating the crops.

Recently, drone-based approaches have been proposed to expel birds from areas that need protection [8]–[10]. For example, Schiano et al. [8] present a pigeon deterrent system, which detects birds using computer vision and directs a drone to the location of the birds. The existing drone-based approaches have shown that a bird deterrent system can materialize in a more intelligent way with autonomous drones. However, the existing approaches are relatively costly for farmers to install and maintain the system because they require autonomous drones controlled by a base station, not easily manageable without the help of experts.

For a smart farming system to be attractive, the system should be low-cost and low-maintenance without loosing its intelligence. Tiny machine learning, which aims to process machine learning algorithms on small embedded devices, can be one applicable technology for the objective. With tiny machine learning techniques [11]–[13], the system can run a smart application without offloading computations to a remote machine. Furthermore, energy harvesting, the technology to gather energy from environmental sources, can enable a maintenance-free system [14] by removing the necessity for battery replacement or the possibility of disruption.

In this paper, we propose a low-cost and low-power smart raven deterrent system with tiny machine learning. First, the proposed system is completely standalone so it does not require neither a base station nor a remote server for computation. This paper proposes a tiny neural network model that exploits audio signals from a microphone for accurate raven detection. The model is trained with an open dataset and optimized for a low-power multi-core microcontroller. In addition, the proposed system is self-sustainable with energy harvesting, minimizing the maintenance cost. Thus, the system can be installed and maintained by non-experts, so the system is easily applicable for smart farming.

To demonstrate the effectiveness of the system, we designed and developed a prototype raven deterrent system with a novel low-power hexa-core microcontroller, Sony Spresense [15]. In the prototype system, a microphone is attached to the microcontroller for the tiny neural network model as well as a solar energy harvesting module. We evaluated the prototype system in terms of detection accuracy and analyzed the power consumption of the system. The evaluation results show that the prototype system can provide accurate audio-based raven detection with an average power consumption of $85.1\,\mathrm{mW}$.

The contributions of this work are:

- Hardware and software design of an edge node for energy efficient smart raven deterrence
- Optimization of deep neural network execution for efficient neural network inference on the edge
- Development and evaluation of the prototype of the raven deterrent system

* Seonyeong Heo is currently at Kyung Hee University, Yongin, Republic of Korea. Email: seonyeong.heo@khu.ac.kr

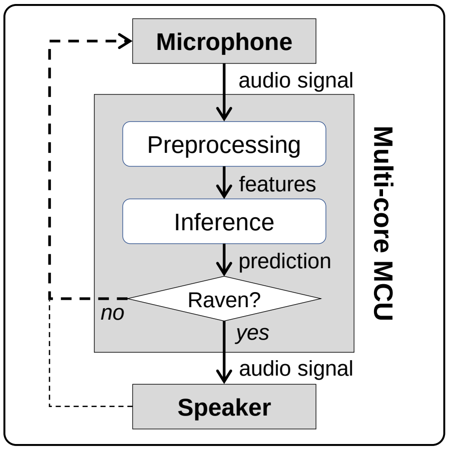


Fig. 1: Overview of the system.

II. System Overview

This paper presents a self-sustainable smart raven deterrent system with a multi-core microcontroller.

A. Hardware

The raven deterrent system consists of a multi-core microcontroller with a microphone for real-time raven detection. In the prototype of the raven deterrent system, we use the Sony Spresence development board (CXD5602) [15], which includes six ARM Cortex-M4F cores with 1.5 megabytes of main memory. By exploiting the multiple cores, the node can process audio signals and run a raven detection algorithm in real time. Additionally, the system uses a speaker to create a sound to expel ravens when they are detected. For self-sustainability, the raven deterrent system is equipped with an energy harvesting module (e.g., a solar charger with a solar panel) and charges its battery with the module.

B. Software

The raven deterrent system performs the following tasks at regular intervals as illustrated in Fig. 1:

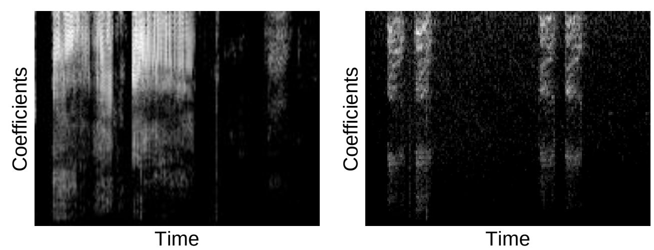
(i) **`Data Acquisition:`** Sample audio signals from its microphone over a defined sampling window.
(ii) **`Preprocessing:`** Perform Mel-frequency Cepstral Coefficient (MFCC) feature extraction on the raw audio data for neural network inference.
(iii) **`Inference:`** Execute the neural network model with the extracted features.
(iv) **`Actuation:`** Play an alarming sound with speakers to deter ravens if the model detects a raven sound from the audio signals.
(v) **`Sleep:`** Go into a sleep mode to preserve energy. The duration of sleep depends on the energy availability of the system.

III. Inference on the Edge

A. Feature Extraction

To facilitate the inference of the neural network, the system first processes raw audio signals from the microphone. Since raw audio signals are relatively large and sparse, it is inefficient to use the raw audio signals as the input of the neural network for both training and testing. Therefore, the system extracts distinct features from the raw audio signals and feeds them to the network for detecting the raven sound.

In the `preprocessing` step, the system calculates the mel-frequency cepstral coefficients (MFCC) [16] of raw audio signals. MFCC is one of the most widely-used representations of audio signals. However, MFCC extraction may be computationally expensive for embedded processors. For example, MFCC extraction requires Fourier transforms of audio frames with floating-point numbers. To reduce the computational burden, the system applies Fast Fourier transform (FFT) instead of Short Time Fourier transform (STFT), which is normally used in MFCC extraction.



(a) Non-raven audio (Frog) (b) Raven audio

Fig. 2: Visualization of MFCCs from sample sounds.

Fig. 2 visualizes the MFCCs of sample raven and non-raven sounds. Fig. 2b shows the pattern of the raven sound, which is distinguishable from the non-raven sound. With the distinct pattern, the neural network can predict whether the audio recorded with the microphone includes a raven call or not.

B. Network Architecture

The proposed system uses a tiny convolutional neural network for audio-based raven detection. Table I summarizes the architecture of the neural network model. Note that batch normalization is applied after every convolution. The input of the model is the features extracted from single-channel audio signals (i.e., MFCCs of the audio signals), of which shape is $(N_f, N_m, 1)$ where N_f is the number of audio frames and N_m is the number of cepstral coefficients. The output of the model is the probability that a raven call is in the audio, of which shape is (2).

The total number of parameters of the model is 19,796. When deployed with TensorFlow Lite Micro (TFLM) [17], the binary size of the model is 84.892 kilobytes and the peak memory footprint of the model is 14.336 kilobytes, with 32 audio frames and 13 cepstral coefficients.

TABLE I: Network Architecture (N: Batch Size)

Type	Kernel	Strides	Activation	Output Dim.
Convolution	3×3	1×1	ReLU	$(N, 32, 13, 3)$
Convolution	3×3	2×2	ReLU	$(N, 16, 7, 16)$
Max pooling	2×2	1×1	-	$(N, 8, 3, 16)$
Convolution	3×3	2×2	ReLU	$(N, 4, 2, 32)$
Max pooling	2×2	1×1	-	$(N, 2, 1, 32)$
Convolution	3×3	2×2	ReLU	$(N, 1, 1, 48)$
Dense	-	-	Linear	$(N, 8)$
Dense	-	-	Softmax	$(N, 2)$

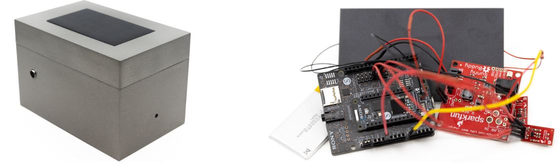
C. Optimizations

As an optimization, matrix multiplication algorithms are parallelized, which are the main computations of neural networks, to exploit multiple cores of the processing unit. For our prototype system, we modified the CMSIS-NN library [18], which provides the optimized neural network kernels for ARM processors, with DOALL parallelism. Additionally, model compression techniques such as quantization [11] are applied to further improve the performance of the network.

IV. Prototype of Raven Deterrent System

A. Hardware

The prototype raven deterrent system is built with a hexa-core Spresence development board (CXD5602) [15] from Sony as its main computational unit; an analog Electret microphone (100 Hz – 10 kHz) with a 60× microphone preamplifier BOB-12758 [19] for raw audio sampling; an ANYSOLAR IXOLAR solar module SM641K10L [20] as the main source of energy; a SparkFun Sunny Buddy solar charger [21] with the maximum power point tracking capability, to maximize the energy harvesting yield; a single cell 3.7 V Renata lithium-ion ICP303450PA-02 battery [22]. Fig. 3 shows the prototype raven deterrent system with a 3D-printed case.



(a) Exterior (b) Hardware Components

Fig. 3: Prototype of the raven deterrent system.

B. Software

Network Training: For the prototype system, the neural network described in Section III is trained with the *Xeno Canto* [23] database, which provides more than 700,000 bird sounds organized by species, location of recording, and etc. The train and test data is acquired by querying and downloading all the samples of Corvus with a web scraper. In total, 6,194 audio recordings were obtained for the raven class from the database. For the background class, 8,000 samples from the *warblrb10k* [24] dataset were taken. The dataset consists of various background noises such as weather sounds, human speech, and even bird imitations.

With the train and test data, the model is trained for 10 epochs with the Adam optimizer and the learning rate of 1e-4 on a desktop computer using TensorFlow [25]. During training, white noise is randomly mixed with original audio samples as a data augmentation. After training, the model is optimized and saved as an 8-bit integer TensorFlow Lite model using TensorFlow Lite Interpreter for the deployment to the target hardware platform.

Implementation: The entire software pipeline is implemented in Arduino programming language with Spresense and TensorFlow Lite Arduino libraries. In more detail, `Audio` API is used for audio recording and `LowPower` API is used for deep sleep, provided by the Spresense Arduino library. For example, the system can change its clock mode using the `LowPower.clockMode(mode)` method where the `mode` parameter can be either `CLOCK_MODE_156MHz`, `CLOCK_MODE_32MHz`, or `CLOCK_MODE_8MHz`.

Implementation of MFCC feature extraction is based on an open-source implementation from ARM [26]. The parameters of the original implementation are modified to generate the correct MFCCs for the trained network.

V. Evaluation

A. Experimental Setup

This paper evaluates the prototype raven deterrent system in terms of raven detection accuracy and power consumption. To evaluate the raven detection accuracy together with the hardware (e.g., microphone), test audio samples are played on a separate host computer. For 50 times, different raven sounds are played to check if the system can correctly identify ravens. Then for another 50 times, non-raven sounds are played (10 times with silence, 10 times with rain sounds, 20 times with sparrow sounds, and 10 times with farm background noises).

Regarding the self-sustainability evaluation, the prototype raven deterrent system is profiled without the solar cells so that the `inference` and `sleep` steps can be evaluated separately. Then, Nordic NRF-PPK2 power profiler is attached to the node instead of the battery of the raven deterrent system to measure the energy consumption within a power resolution of $\pm 5\,\mu$W at 3.7 V.

B. Raven Detection Accuracy and Latency

Table II shows the confusion matrix of the whole system on unforeseen test audio samples. The table shows that the system yields an overall accuracy of 77% for the 100 test samples. As well as in Fig. 4, the precision-recall (PR) curve on the left shows an average precision of 87% and the receiver operating characteristic (ROC) curve on the right shows an Area under the ROC Curve (AUC) of 84%.

The result demonstrates that the prototype system can detect raven sounds with robustness towards microphone noise as well as background environment noise. In addition, when the system detects a raven sound, the prediction can be trusted with the confidence of 87% even against other songbird

TABLE II: Confusion Matrix of Predictions

		Prediction	
	Total: 100	raven: 35	no raven: 65
Actual	raven: 50	**31**	19
	no raven: 50	4	**46**

sounds. Therefore, the low false positive rate can avoid a negative environmental impact on wildlife by not making alarm sounds unnecessarily.

Lastly, we measure the latency of the neural network on the multi-core microcontroller. The latency of the `inference` step is 57.8 ms on average, excluding the `preprocessing` step which takes 253.9 ms for 3-second long audio signal. Hence, the `inference` and `preprocessing` steps take 311.7 ms in total, as shown in Fig. 5a.

C. Power Profiling

The power consumption of the prototype system is profiled and analyzed with the three time-consecutive phases:

i) **Phase 1**: This phase includes data acquisition from the microphone, preprocessing with MFCC extraction, execution of the neural network, and actuation of the speaker. This phase is the most power intensive among the three phases. As shown in Fig. 5a, the prototype system has an average power consumption of $P_1 = 144.2\,\mathrm{mW}$ with a duration of $t_1 = 12.9\,\mathrm{s}$.

ii) **Phase 2**: In this phase, the system conserves energy by shutting down all the subsystems in a deep sleep mode. This phase is the least power intensive among the three phases. As shown in Fig. 5b, the prototype system has an average power consumption of $P_2 = 7.9\,\mathrm{mW}$ where t_2 is set to 30 s in the measurement, though t_2 can be set to an arbitrary time.

iii) **Phase 3**: When rebooting the node, the system has to restart and thus yields a power consumption to get the system fully up and running again. This phase has to be taken into account at every waking up from Phase 2 and is accompanied with Phase 1. As in Fig. 5c, the node has an average power consumption of $P_3 = 80.1\,\mathrm{mW}$ with a duration of $t_3 = 8.8\,\mathrm{s}$.

D. Self-Sustainability Analysis

To estimate the perpetual work of the proposed system, a brief discussion on the estimated solar power availability has to be held.

With the system being meant to be used during spring and summer in Switzerland, an estimate of the lowest accounted solar radiation is $60\,\mathrm{W\,m^{-2}}$. This is derived using the data provided by Swiss Meteo [27] in Zurich. Using the data of April to September of the last two years, one can see that the solar radiation threshold of $60\,\mathrm{W\,m^{-2}}$ is achieved nearly at all times. An average of $189.9\,\mathrm{W\,m^{-2}}$ solar radiation was measured. Further the mean hours of sunshine are determined

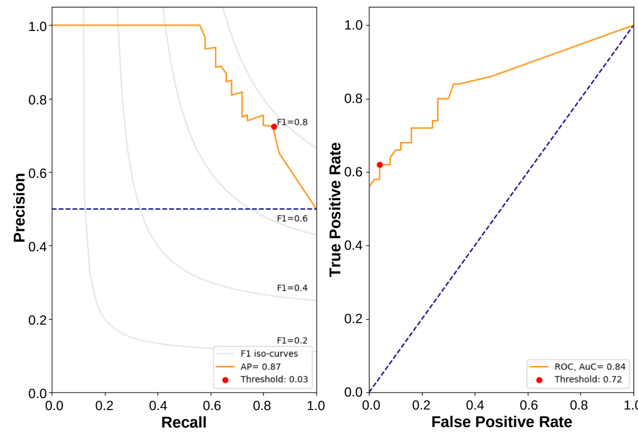


Fig. 4: Left: Precision-recall curve of the neural network model where the dashed line indicates the no-skill threshold. Right: ROC curve of the neural network model.

to be $7h$, this is derived by taking the mean daylight hours per day from the past 5 years of the months April to August. This results in a mean of $7.22h$ which we round down to $7h$ and denote as t_{day} [28].

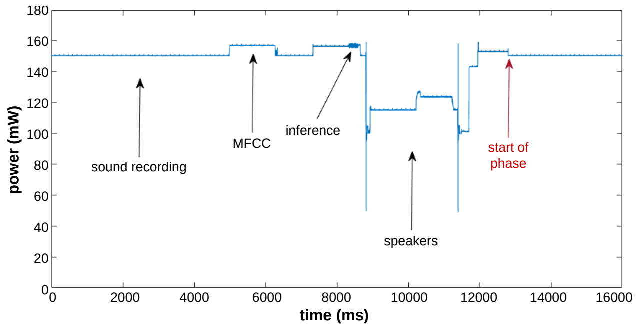
The utilized $0.11\,\mathrm{m} \times 0.06\,\mathrm{m}$ solar module has an efficiency of 25% and a surface area of $6.993 \times 10^{-3}\,\mathrm{m}^2$ [20]. With these values and the power consumption measured in the different modes in Fig. 5, the energy harvested in a day can be calculated as $E_{harv} = 2620.7\,\mathrm{J}$. During the night the system is in Phase 2, during at which it consumes $E_{night} = 482.3\,\mathrm{J}$. The resulting energy budget is therefore: $E_{day} = 2620.7 - 482.3 = 2138.4\,\mathrm{J}$. Lastly, to calculate the maximum possible duty cycle D of Phase 1 to Phase 2 within the energy envelope, can be inferred by defining the following definitions and conditions:

$$
\begin{aligned}
t_{on} &:= t_1 + t_3 \\
P_{on} &:= \frac{P_1 \cdot t_1 + P_3 \cdot t_3}{t_1 + t_3} = 118.2\,\mathrm{mW} \\
t_{tot} &:= t_{on} + t_2 \\
D &:= \frac{t_{on}}{t_{on} + t_2} = \frac{t_{on}}{t_{tot}}
\end{aligned}
$$

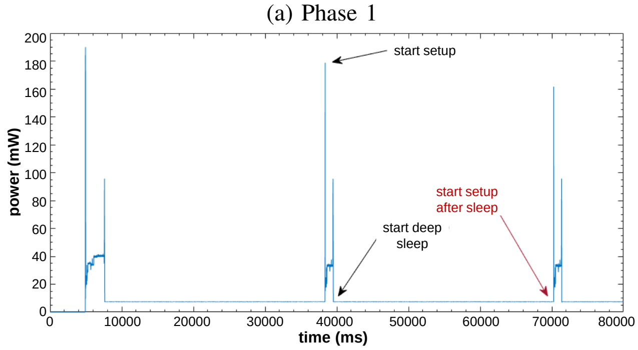
Lastly by solving for the duty cycle D:

$$
\begin{aligned}
E_{tot} &= t_{tot} \left\{ D \cdot P_{on} + (1 - D) \cdot P_2 \right\} \\
&= \frac{t_{on}}{D} \left\{ D \cdot P_{on} + (1 - D) \cdot P_2 \right\} \\
&\overset{!}{=} \frac{t_{on}}{D \cdot t_{day}} \cdot 2138.4\,\mathrm{J}
\end{aligned}
$$

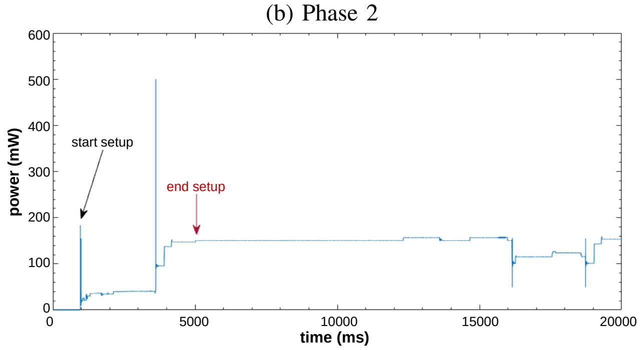
This results in $D = 0.70$ and consequently $t_2 = 9.3\,\mathrm{s}$. Hence, during the day, Phase 1 and Phase 3 can run at a duty cycle of 70% and a period of $t_1 + t_2 + t_3 = 31\,\mathrm{s}$ and be self-sustainable with the energy being harvested trough the solar cell and harvesting module. This results in an average power consumption of $85.1\,\mathrm{mW}$ during the day.



(a) Phase 1



(b) Phase 2



(c) Phase 3

Fig. 5: Power profiles of the prototype system for each phase.

VI. Conclusion

This paper presents a self-sustaining, fully embedded raven deterrent system for smart farming. The proposed system leverages tiny machine learning for model deployment to a multi-core microcontroller. By deploying the model to an embedded system, we derive a solution that enables low-cost, low-maintenance, and self-sustainability, which are the key attributes pursued in smart farming. The prototype system obtains 77% detection accuracy while consuming the average power of $85.1\,\mathrm{mW}$.

Future work on this topic could address a more effective deterrence mechanism. While the proposed system currently deters ravens with a loud audio signal, ravens could be intelligent enough to disassociate it with their actions of damaging crop, albeit the targeted action. A possible solution to this problem is to combine various existing bird deterrent methods with the proposed raven deterrent system.

Acknowledgments

We thank the anonymous reviewers for their valuable feedback. This research was supported by Sony's Sensing Solution University Program.

References

[1] M. S. Farooq, S. Riaz, A. Abid, K. Abid, and M. A. Naeem, "A survey on the role of iot in agriculture for the implementation of smart farming," *IEEE Access*, vol. 7, pp. 156 237–156 271, 2019.

[2] S. R. Prathibha, A. Hongal, and M. P. Jyothi, "IOT Based Monitoring System in Smart Agriculture," in *2017 International Conference on Recent Advances in Electronics and Communication Technology (ICRAECT)*, Mar. 2017, pp. 81–84.

[3] Y. Kalyani and R. Collier, "A Systematic Survey on the Role of Cloud, Fog, and Edge Computing Combination in Smart Agriculture," *Sensors*, vol. 21, no. 17, p. 5922, Jan. 2021, number: 17 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/21/17/5922

[4] "Agroscope the swiss federal competence centre for agricultural research," https://www.agroscope.admin.ch/agroscope/de/home/ueber-uns/agroscope.html, accessed: 2021-11-30.

[5] D. Adami, M. O. Ojo, and S. Giordano, "Design, Development and Evaluation of an Intelligent Animal Repelling System for Crop Protection Based on Embedded Edge-AI," *IEEE Access*, vol. 9, pp. 132 125–132 139, 2021, conference Name: IEEE Access.

[6] P. Gras, T. Tscharntke, B. Maas, A. Tjoa, A. Hafsah, and Y. Clough, "How ants, birds and bats affect crop yield along shade gradients in tropical cacao agroforestry," *Journal of Applied Ecology*, vol. 53, no. 3, pp. 953–963, Mar. 2016. [Online]. Available: https://doi.org/10.1111/1365-2664.12625

[7] A. T. Baxter and A. P. Robinson, "A comparison of scavenging bird deterrence techniques at uk landfill sites*," *International Journal of Pest Management*, vol. 53, no. 4, pp. 347–356, 2007. [Online]. Available: https://doi.org/10.1080/09670870701421444

[8] F. Schiano, D. Natter, D. Zambrano, and D. Floreano, "Autonomous detection and deterrence of pigeons on buildings by drones," *IEEE Access*, vol. 10, pp. 1745–1755, 2022.

[9] A. B. Mahmood, S. Gregori, J. Runciman, J. Warbick, H. Baskar, and M. Badr, "Uav based smart bird control using convolutional neural networks," in *2022 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 2022, pp. 89–94.

[10] Z. Wang, A. S. Griffin, A. Lucas, and K. Wong, "Psychological warfare in vineyard: Using drones and bird psychology to control bird damage to wine grapes," *Crop Protection*, vol. 120, pp. 163–170, 2019.

[11] D. Lin, S. Talathi, and S. Annapureddy, "Fixed point quantization of deep convolutional networks," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 2849–2858. [Online]. Available: https://proceedings.mlr.press/v48/linb16.html

[12] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," in *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2016. [Online]. Available: http://arxiv.org/abs/1510.00149

[13] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015. [Online]. Available: https://arxiv.org/abs/1503.02531

[14] M. Giordano, P. Mayer, and M. Magno, "A battery-free long-range wireless smart camera for face detection," in *Proceedings of the 8th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, 2020, pp. 29–35.

[15] Sony, "Spresense hardware documents," 2022, [Online; accessed 4-July-2022]. [Online]. Available: https://developer.sony.com/develop/spresense/docs/hw_docs_en.html

[16] M. Xu, L.-Y. Duan, J. Cai, L.-T. Chia, C. Xu, and Q. Tian, "Hmm-based audio keyword generation," in *Advances in Multimedia Information Processing - PCM 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 566–574. [Online]. Available: https://doi.org/10.1007/978-3-540-30543-9_71

[17] R. David, J. Duke, A. Jain, V. Janapa Reddi, N. Jeffries, J. Li, N. Kreeger, I. Nappier, M. Natraj, T. Wang, P. Warden, and R. Rhodes, “Tensorflow lite micro: Embedded machine learning for tinyml systems,” in *Proceedings of Machine Learning and Systems*, A. Smola, A. Dimakis, and I. Stoica, Eds., vol. 3, 2021, pp. 800–811. [Online]. Available: https://proceedings.mlsys.org/paper/2021/file/d2ddea18f00665ce8623e36bd4e3c7c5-Paper.pdf

[18] L. Lai, N. Suda, and V. Chandra, “Cmsis-nn: Efficient neural network kernels for arm cortex-m cpus,” 2018. [Online]. Available: https://arxiv.org/abs/1801.06601

[19] SparkFun. (2016, Mar.) SparkFun Electret Microphone Breakout. [Online]. Available: https://www.sparkfun.com/products/12758

[20] ANYSOLAR. (2021, Jan.) IXOLAR High Efficiency SolarMD - Data Sheet. [Online]. Available: https://waf-e.dubudisk.com/anysolar.dubuplus.com/techsupport@anysolar.biz/O18AzrW/DubuDisk/www/Gen3/SM641K10L%20DATA%20SHEET%2020210127.pdf

[21] SparkFun. (2014) SparkFun Sunny Buddy - MPPT Solar Charger. [Online]. Available: https://www.sparkfun.com/products/12885

[22] RenataSA. (2019, Aug.) 3V Li Battery. ICP303450PA-02. [Online]. Available: https://www.renata.com/en-ch/downloads/?product=icp303450pa-02&fileid=362eb77e217c713a5b0570752b

[23] Xeno-Canto, “Xeno-canto :: Vogelstimmen aus aller welt teilen,” 2022, [Online; accessed 13-June-2022]. [Online]. Available: https://xeno-canto.org/

[24] A. Mikołajczyk, “bird-recognition-review github,” 2022, [Online; accessed 13-June-2022]. [Online]. Available: https://github.com/AgaMiko/Bird-recognition-review/blob/master/README.md#Datasets

[25] M. Abadi et al., “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[26] “Keyword spotting for microcontrollers,” https://github.com/ARM-software/ML-KWS-for-MCU, accessed: 2023-02-23.

[27] MeteoSchweiz, “Messwerte,” 2018, [Online; accessed 4-July-2022]. [Online]. Available: https://www.meteoschweiz.admin.ch/home/messwerte.html?param=messwerte-globalstrahlung-10min&station=REH&chart=day

[28] WeatherOnline, “Sonnenstunden pro tag - zürich - klimarechner schweiz,” 1999, [Online; accessed 4-July-2022]. [Online]. Available: https://www.weatheronline.de/weather/maps/city?FMM=1&FYY=2016&LMM=12&LYY=2021&WMO=06660&CONT=euro&REGION=0001&LAND=SW&ART=SON&R=0&NOREGION=0&LEVEL=162&LANG=de&MOD=tab