

Battery-Free Camera Occupancy Detection System

Ali Saffari*
Sin Yong Tan*
saffaria@uw.edu
tsyong98@iastate.edu
University of Washington
Iowa State University

Mohamad Katanbaf
katanbaf@uw.edu
University of Washington

Homagni Saha
hsaha@iastate.edu
Iowa State University

Joshua R. Smith
jrs@cs.washington.edu
University of Washington

Soumik Sarkar
soumiks@iastate.edu
Iowa State University

ABSTRACT

Occupancy detection systems are commonly equipped with high-quality cameras and a processor with high computational power to run detection algorithms. This paper presents a human occupancy detection system that uses battery-free cameras and a deep learning model implemented on a low-cost hub to detect human presence. Our low-resolution camera harvests energy from ambient light and transmits data to the hub using backscatter communication. We implement the state-of-the-art YOLOv5 network detection algorithm that offers high detection accuracy and fast inferencing speed on a Raspberry Pi 4 Model B. We achieve an inferencing speed of $\sim 100ms$ per image and an overall detection accuracy of $>90\%$ with only 2GB CPU RAM on the Raspberry Pi. In the experimental results, we also demonstrate that the detection is robust to noise, illuminance, occlusion, and angle of depression.

CCS CONCEPTS

• Computing methodologies → Object detection; • Hardware → Wireless integrated network sensors.

KEYWORDS

Backscatter, Wireless Battery-Free Camera, Neural Networks, Occupancy Detection, Computer Vision

1 INTRODUCTION

Recent advances in sensors, wireless networks, and computer vision have enabled us to design smart devices that sense their surroundings and interact with humans and other intelligent objects. For instance, recent efforts in the development of smart surveillance systems have improved the accuracy of automated security systems wherein smart sensors are used to detect human presence. However, deployment of these automated systems is limited to scenarios where

plug-in sensors communicate with a powerful processor for occupancy detection. An ideal human occupancy detection system must satisfy two major requirements: first, the wireless sensors should be maintenance-free and do not require battery charging or replacement. Second, a low-cost processor should be able to detect human presence by using a fast and accurate algorithm.

Outdoor occupancy detection systems typically use high-resolution solar-powered cameras for human presence detection. However, these systems might require battery charging if the harvested energy could not support the operation of the camera [1]. Implementation of solar-powered cameras is even more challenging in indoor cases as the solar panels harvest less energy indoor. To address this challenge, researchers have studied other sensors such as carbon dioxide, temperature and humidity [30, 31], and passive infrared (PIR) sensors [2, 35] to replace the camera with a low-power sensor. However, these systems still need batteries for operation. Furthermore, environmental data is susceptible to abrupt pattern changes due to external factors such as an opened window or door and is generally restricted to an enclosed environment.

Image data, on the other hand, can provide a clear visual indication of a human presence. However, due to the limited computational resources on an embedded device, human detection commonly relies on shallow models such as Support Vector Machine [4, 6, 27] to perform real-time detection on images. Object detection and image classification models such as You Only Look Once (YOLO) networks [9, 19, 28, 33], Single Shot MultiBox Detector (SSD) [14], Few-shot models [25, 34] and RetinaNet [17] are generally too computation-intensive to perform real-time detection on an embedded system without GPU. In many cases, to perform inference or detection using a deep learning model, we require a computer with considerable computation capacity or an expensive device such as Jetson TX2 [28] to be able to handle the streaming images captured by the cameras. These examples show a contradicting trade-off between computation resources and detection performance. On the one hand, we want to reap the benefit of high detection accuracy from deep learning models. Still, ideally, on the other hand, we want to achieve that without expending excessive computation resources.

In this work, we design an occupancy detection system that uses a **lightweight yet accurate algorithm** implemented on a **low-cost processor** to analyze data from **battery-free cameras** and detect human presence. In this system, cameras capture low-resolution images and transmit them to a hub equipped with a low-cost processor for

*Both authors contributed equally to this research.



This work is licensed under a Creative Commons Attribution International 4.0 License.
EMDL '21, June 25, 2021, Virtual, WI, USA
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-8597-8/21/06.
<https://doi.org/10.1145/3469116.3470013>

occupancy detection. We use a low-power communication system called backscatter to send data to the hub in the implementation of battery-free cameras. Since the required energy for data transmission is low, a solar panel in an indoor setup can harvest enough energy from ambient light to support the camera’s operation.

The **contributions** of this paper are listed below:

- (1) Implementation of wireless battery-free cameras that harvest energy from ambient light.
- (2) Deployment of occupancy detection YOLOv5 model on an embedded system.
- (3) Demonstration of inferencing speed, detection accuracy, and reliability of the proposed system.

2 RELATED WORK

We review the prior efforts in battery-free sensing and detection algorithm in this section.

Battery-free sensing. To enable battery-free sensing, two conditions must be satisfied; first, operation of the sensor node, sensing and communicating, should be power efficient, and second, a harvester should provide enough energy to support this power-efficient operation. Backscatter communication technology has addressed the first challenge and has been used to design and implement battery-free sensors. In [26], a wireless sensing platform is developed using RFID readers. The reader in RFID technology uses backscatter communication to receive data from the battery-free sensors in the network. Researchers in [20–22], introduce a battery-free camera that captures still images and transmits them via RFID backscatter communication. Since the required energy for the operation of this camera is provided by harvesting energy from the RFID reader, the operating range is only limited to a few meters.

In [7, 24], a wireless battery-free camera is designed. This system uses an analog image sensor to stream video. The authors use Pulse Width Modulation (PWM) to convert the analog pixels value to digital for backscatter data transmission.

As mentioned, energy harvesting is a major obstacle in implementing battery-free sensors. To address this challenge, researchers have been trying to improve the efficiency of exiting harvesters and design novel boards capable of harvesting energy from multiple sources. In [24], a dual harvester board provides power for the camera’s operation by collecting energy from ambient light and RF signals simultaneously, allowing the camera to operate at longer distances. Researchers in [3] design the first battery-free personal gaming device powered by collected energy from sunlight and the user actions. Other sources of energy are used to power battery-free sensors such as thermal [18], vibration [13], and motion [29].

Detection algorithm. In the realm of object detection, or specifically, in human detection, many techniques and algorithms are proposed in previous literature. For example, the combination of feature extraction using Histogram of Gradient (HOG) and the Support Vector Machine (SVM) as the classifier [4, 6, 27] are frequently used. In these papers, HOG is used to effectively extract the human feature and contour from an input image, and these extracted features are then fed into an SVM for classification. On top of that, with the small model size and inferencing speed, it becomes a suitable candidate to be deployed on an embedded system to perform real-time detection.

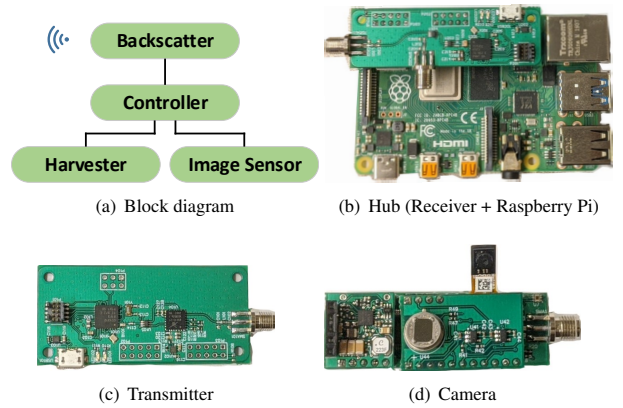


Figure 1: Prototype hardware.

In [4, 6], the detection is performed on relatively more straightforward human posture like standing or walking, and the whole human body is perfectly visible. In [27], the test cases presented in the paper are slightly more challenging. The images are captured in a certain angle of depression, and the humans’ figures appear smaller in the images. However, the performance of the detection algorithm under occlusion cases is still not discussed in the papers.

On the other hand, with the recent advances in deep learning, there are also papers that use deep learning models such as YOLO [19, 28, 33] to detect person for occupant density detection, human behavior, and motion analysis. In [19], the authors claim that the YOLOv3 model is performing better in cases of partial occlusion. In terms of inferencing speed, however, in [33], the authors mention that the algorithm would require a GPU to perform inferencing in a real-time fashion. An example of this is shown in [19], whereby using a Jetson TX2 and TensorRT optimization, the system can achieve 10 FPS of processing speed. While deep learning models offer a more robust detection performance, deploying a deep learning model on an embedded system for real-time detection remains a challenge.

3 SENSOR DESIGN

Our goal is to design a battery-free camera that captures low-resolution images and transmits them to a hub via wireless communication. To enable this, we use a low-power microcontroller to read images from the image sensor and send them to the receiver using backscatter communication technology. Our solar harvester stores enough energy to support the operation of the camera. Figure 1 shows the block diagram of our system and our camera prototype.

3.1 Image Sensor

We use a Himax HM01B0 low-power image sensor that outputs gray-scale video with a bit depth of 8 per pixel to capture 120×120 pictures. This image sensor has the lowest power consumption among the low-resolution sensors since it consumes only 1.1 mW to stream video at 30 frames per second. The process of capturing an image starts with enabling the image sensor. Next, the microcontroller unit (MCU) reads the image row by row and saves it in memory. Once the picture is recorded, the MCU disables the image sensor to keep the camera’s power consumption in sleep mode low.

To improve the image quality, we remove the static noise in the images. A white paper is placed in front of the camera’s lens, and hundreds of images are recorded to estimate this noise. Taking an average across these images provides an estimate of the static noise. We remove this offset by subtracting the per-pixel average from each pixel of the image that we record. Figure 2 shows a sample image before and after the noise removal process.

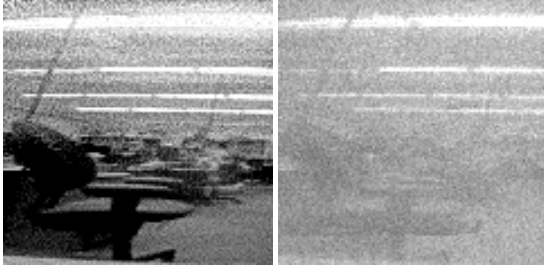


Figure 2: Image sample after (left) and before (right) enhancement.

3.2 Microcontroller and Energy Harvester

We use STM32L071C8, an ultra-low-power MCU with a clock frequency of 4 MHz, to initialize the image sensor, receive images from it, and transmit them in multiple packets to the hub using backscatter communication. In the following paragraphs, we explain the process of capturing an image and transmitting it to the receiver in more detail.

To save energy, we keep our system in sleep mode and wait for the hub to request an image. Our camera is equipped with the low-power Amplitude Shift Keying (ASK) receiver used in [10] to enable this down-link communication. The ASK receiver listens to the commands from the hub and wakes the microcontroller from sleep mode upon detection of the image request command.

Once the command is received, the microcontroller’s Phase-Locked Loop (PLL) generates a 3 MHz clock signal for the operation of the image sensor and initializes it via the I2C bus. Next, the image sensor captures an image and uses the SPI bus to transfer each row’s pixels value to the MCU. This process is finished when the MCU reads all of the 120 rows. Finally, to transmit the image to the hub, the MCU breaks down the image into 12 portions or 120 rows. The hub can request transmission of a portion or row by sending the dedicated command to the camera.

The MCU uses the backscatter block for image data transmission. We stream the raw data to the hub and do not perform any compression on the camera board because the power and time required for compression are higher than the needed time and energy to stream raw data [8]. Moreover, compression becomes more critical where in contrast to our system, the communication cost is high, so the data size must be as small as possible. To make our communication system more robust, the microcontroller computes the Cyclic Redundancy Check (CRC) of each communication packet and transmits it at the end of the packet. If this CRC does not match the CRC that the hub calculates, the hub requests re-transmission of the packet.

We use a TI BQ25570 solar energy harvesting Integrated Circuit (IC) to power the camera’s operation. The amount of energy that this IC stores in the supercapacitor depends on the size of the solar

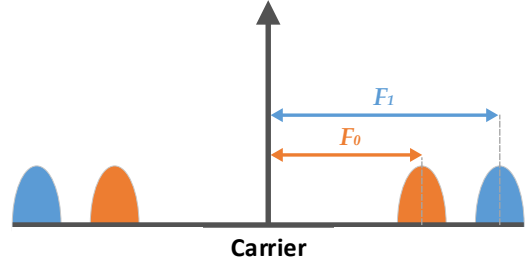


Figure 3: FSK backscatter in frequency domain.

panel and light intensity in the experiment room; using a larger solar panel or deployment in a room with higher light intensity results in a higher update rate.

3.3 Backscatter Communication

Backscatter communication is a low-power wireless technology wherein a transmitter unit generates a carrier signal, a sensor node reflects the carrier, and a receiver unit decodes the sensor node data [7, 11, 12, 16]. Since the sensor does not need to generate any signal to communicate with the receiver and only reflects the carrier, the power consumption for data transmission is low.

We implement a Frequency Shift Keying (FSK) backscatter unit to transmit the images to the hub at the data rate of 125 Kbps. In our design, the microcontroller changes the state of the RF switch connected to the antenna at two frequency rates to reflect the carrier. Figure 3 shows the carrier signal and bit 1, and bit 0 with frequency offset of f_1 and f_0 from the carrier, respectively. Our commutation system allows the camera to communicate with the receiver while there is up to 130 ft distance between them [23]. Our custom-designed receiver board equipped with a Raspberry Pi 4 for data processing and the transmitter prototype are shown in Figure 1.

4 ALGORITHM

This section discusses the algorithm that we use to perform occupancy detection on the embedded system.

4.1 YOLOv5 Network

To detect human presence, we implement a recent state-of-the-art detection algorithm, YOLOv5 [9]. Similar to most single-stage object detection models [14, 17], YOLOv5 consists of three components, each with an essential role in the model.

The first component of the model, known as the model backbone, is a feature extractor network that extracts rich and useful features from the input images. Many model backbones are developed over the years, such as Cross Stage Partial Network (CSPNet)[32] and Residual Network[5]. In YOLOv5, the former is used as the model backbone. The next component is the model neck, whose primary function is to generate feature pyramids. Feature pyramids are an essential block that allows the model to detect objects at different sizes and scales by constructing multi-scale feature maps. Similar to YOLOv4, YOLOv5 uses the Path Aggregation Network (PANet) [15] which combines the top-down and bottom structure as the model neck. The last component of the model is the model

head, which generates the predictions for bounding boxes and object classes.

The output tensor dimensions for YOLOv5 can be expressed as $d \times (b + n)$, where d is the number of detection layers, $b = (x, y, w, h, c)$ representing the predicted bounding box's coordinates, width, height and the prediction confidence, and n is the number of classes. The model has three detection layers in our application, and since we are only interested in a single class, human presence detection, the output tensor dimensions are updated to $3 \times (5+1) = 18$. The models are introduced with four different sizes; small (s), medium (m), large (l), and extra-large (x), with the model sizes sorted in ascending order. The model sizes are increased by increasing the model height and width multiplier, which will increase the number of layers and layer channels. Considering the small input image size and the limited computation resources on Raspberry Pi, we choose the s model with a 0.33 height multiplier and 0.5 width multiplier for our application. The resulting model weight is merely 14 MB with 7.25 M parameters and 16.8 GFLOPs in the model.

4.2 Training

Collected images by the cameras are labeled and annotated with bounding boxes around the occupants. We employ several data augmentation techniques such as brightness varying, image flipping, and mosaic data augmentations to generate a richer training image variation. These data augmentations procedures are essential to enhance the robustness of the trained model to various scenarios such as different illuminance levels and human position in the images. For this single-class detection implementation, we generate up to 4,308 images after augmentations(excluding mosaic augmentation), and we use 3,016 images for training, 430 images for validation, and 862 images for testing. In the next step, these images are fed into the YOLOv5 network for model training. The model is trained on Google Colab using a Tesla T4 GPU for 300 epochs. After training the model, the model weight and architecture are saved for inferencing purposes.

4.3 Deployment

The model weights are first converted into Tensorflow Lite (TF-Lite) format before deploying the saved model on Raspberry Pi. As a result, we can efficiently perform model inferencing on the Raspberry Pi using the TF-Lite Interpreter. TF-Lite Interpreter is a lightweight model interpreter library dedicated to model inferencing on various embedded systems. This essentially reduces the required memory size on the embedded system as it exempts the need to install a full-blown deep learning library to run the model.

5 EVALUATION

We perform several experiments to demonstrate the performance of our system. First, we measure the power consumption of the camera. Second, we deploy our camera in different locations to evaluate the accuracy of the model.

5.1 Camera Power Consumption

As mentioned in section 3.2, the MCU breaks down the image data into 12 portions or 120 rows to transmit data to the hub. Figure 4 shows the power that the camera consumes to send one portion of an

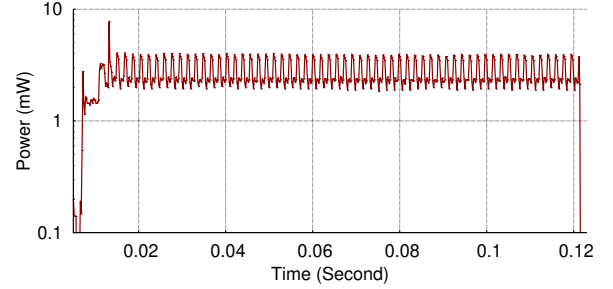


Figure 4: Camera's power consumption during transmission of a portion of the image.

Sleep Power (μ W)		30
Sensing Energy (mJ)		6.05
Communication Energy (mJ)		3.82
Solar-Powered Update Rate (sec.)	2 in^2 panel	280
	17 in^2 panel	17

Table 1: Power consumption measurements for camera.

image to the hub. Each portion is composed of 60 communication packets. The peaks in the power consumption plot are caused by the MCU computing the CRC of each packet. To measure the camera's power consumption, we use a 2.2 Ω resistor in series with the camera and measure the voltage of the resistor, which is 2.2x the current the camera and the resistor consume. We analyze the voltage data that the oscilloscope records to report the power numbers. Since the resistor value is small, we can assume that the camera consumes the total current. We list the energy that the camera needs to capture and transmit an image in Table 1.

To evaluate the harvester's performance, we use AM-1801 and AM-1816CA solar panels with areas of 2 in^2 and 17 in^2 , respectively, in an office with the light intensity of 300 lux for collecting energy. Table 1 lists the average update rates that we can achieve using these solar panels.

5.2 Experiment Setup

The system setup consists of two camera nodes, a transmitter, and a receiver attached to a Raspberry Pi. During the system testing, the cameras are installed in each zone to capture images from different directions and angles. The system is tested at two indoor locations, a residential unit, and a computer lab. For privacy concerns, the system is only set up in the common areas such as the kitchen and the living room in the residential unit. These selected testing locations are chosen to provide diverse indoor testing scenarios. For example, in the kitchen, occupants are often standing and walking as they are cooking or cleaning the dishes, while in the living room, occupants are usually sitting on the couch watching television or using their laptops. These scenarios produce images with different human postures, lighting conditions, and occupants' distance to the camera (human figure size in pictures), enabling a more comprehensive system performance evaluation.

In each location, we run the system for ~ 10 hours from sunrise to sunset, with 0 \sim 2 occupants in the zone throughout the experiment period. Moreover, to evaluate the reliability of the system, a final experiment is performed with an extended period, where we run

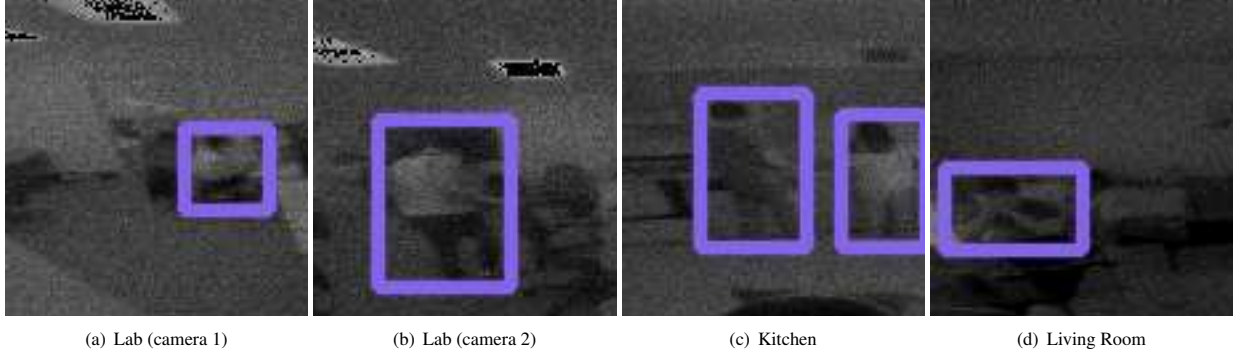


Figure 5: Figure shows some example detection results for each experiment zone.

the system uninterrupted for up to 5 days, with the camera nodes installed in two zones (kitchen and living room) simultaneously. This experimental setup allows the system to cover a large area in the residential unit, with up to a $\sim 30\text{ ft}$ euclidean distance between the receiver and camera sensor nodes.

5.3 Experiment Results

In this section, we discuss the performance of our system at locations mentioned in Section 5.2. Figure 5 shows four sample images wherein the YOLOv5 model detects occupants and annotates them with bounding boxes. Each example image shown in the figure depicts a different zone/background or posture. For example, in Figure 5(a), the image shows the person in a sitting position and far away from the camera. In contrast, another camera at the same location captures a picture of the person walking toward the door from a different angle. The model can detect the person in both scenarios despite the different posture and distance from the camera. Moreover, in Figure 5(c), the image shows that our model can detect multiple people in one image. Finally, in Figure 5(d), the image shows a scenario where the occupant is looking at the laptop in the living room with only the top half of the body visible. This example shows that the model can detect the person despite the occlusion of the lower half body and it is robust to a small angle of depression.

On top of the output illustrations, Table 2 also provides a summary of the outcome of the performed experiments. In each zone, the camera captures and infers up to $\sim 3,000$ images, and in the last experiment, the system records more than 30,000 images, with an inference time of only $\sim 100\text{ms}$ per image. To simplify results presentation, we use a 0.5 confidence threshold in the inferencing, and each frame with a detected person(s), will be labeled as *occupied*, and otherwise, as *unoccupied*. After evaluating these occupied and unoccupied frames, each experiment achieves an accuracy of $>90\%$, including the last experiment with two zones and an extended experiment period. This shows that the system can accurately detect human presence and operate reliably in a large area for a long duration.

Every system has its advantages and shortcomings. In our case, since our proposed detection system is vision-based, the images captured by the camera are inherently reliant on the surrounding illuminance. Under the low light condition, images captured appear much darker with small pixel intensity values, as shown in Figure 6. The red circle in Figure 6 indicates an occupant in the frame, but

Zones	Total Num. Images	Accuracy(%)
Lab	2,951	99.56
Kitchen	3,508	99.72
Living Room	3,082	92.53
Kitchen and Living Room (5 days)	32,914	92.72

Table 2: Table shows the detection accuracy rates for different testing locations and the total number of camera images collected using the developed system.

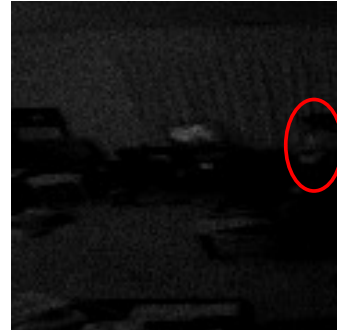


Figure 6: A sample image captured in a dark environment with the occupant circled in red.

the image captured does not contain enough contrast that shows the human figure in the image. The combination of low-resolution and dark images causes human detection to become a challenging task. This situation can potentially be remedied by equipping the detection system with simple motion detection and image enhancement algorithms. In our future work, the adaptation of this system in low light conditions will be further addressed in more detail.

6 CONCLUSION

This paper proposed a power-efficient and high-performing occupancy detection system deployed on an embedded system with a low-cost processor. Using a solar energy harvester and backscatter communication, we successfully powered the low-power camera sensor and transmitted the captured images to the receiver. The power consumption of the camera for sensing and transmitting an image was discussed in detail. For the detection algorithm, we deployed a

state-of-the-art detection model, the YOLOv5 network, on the Raspberry Pi to detect occupants accurately. Multiple indoor experiments were performed to demonstrate the performance of the proposed system. The experiment results showed promising results with >90% detection accuracy and a fast inferencing time of ~100ms.

ACKNOWLEDGMENT

We thank the anonymous reviewers for their helpful feedback on the paper. This work was funded in part by ARPA-E (DE-AR0000938) and NSF CRI award CNS-1823148. Any opinions, findings and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

REFERENCES

- [1] Solar-powered hd camera, <https://www.amazon.com/powerd-wireless-surveillance-outdoor-business/dp/b07wv49x9y>, 2021.
- [2] Jack Andrews, Meghana Kowsika, Asad Vakil, and Jia Li. A motion induced passive infrared (pir) sensor for stationary human occupancy detection. In *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pages 1295–1304, 2020.
- [3] Jasper de Winkel, Vito Kortbeek, Josiah Hester, and Przemyslaw Pawelczak. Battery-free game boy. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–34, sep 2020.
- [4] Oscar Arturo González González, Alina Mariana Pérez Soberanes, Victor Hugo García Ortega, and Julio César Sosa Savedra. Embedded system for human detection applied to domotics. *Research in Computing Science*, 148:103–115, 2019.
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [6] Li He, Youlu Wang, Senem Velipasalar, and M Cenk Gursay. Human detection using mobile embedded smart cameras. In *2011 Fifth ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–6. IEEE, 2011.
- [7] Mehrdad Hessar, Saman Naderiparizi, Ye Wang, Ali Saffari, Shyamnath Gollakota, and Joshua R. Smith. Wireless video streaming for ultra-low-power cameras. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, MobiSys '18*, page 536, New York, NY, USA, 2018. Association for Computing Machinery.
- [8] Vikram Iyer, Ali Najafi, Johannes James, Sawyer Fuller, and Shyamnath Gollakota. Wireless steerable vision for live insects and insect-scale robots. *Science Robotics*, 5(44), 2020.
- [9] Glenn Jocher, Alex Stoken, Jirka Borovec, NanoCode012, Ayush Chaurasia, TaoXie, Liu Changyu, Abhiram V, Laughing, tkianai, yxNONG, Adam Hogan, lorenzomamma, AlexWang1900, Jan Hajek, Laurentiu Diaconu, Marc, Yonghye Kwon, oleg, wanghaoyang0106, Yann Defretin, Aditya Lohia, ml5ah, Ben Milanko, Benjamin Fineran, Daniel Khromov, Ding Yiwei, Doug, Durgesh, and Francisco Ingham. ultralytics/yolov5: v5.0 - YOLOv5-P6 1280 models, AWS, Supervise.ly and YouTube integrations, Apr. 2021.
- [10] Mohamad Katanbaf, Vivek Jain, and Joshua R. Smith. Relacks: Reliable backscatter communication in indoor environments. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(2):1–24, jun 2020.
- [11] M. Katanbaf, A. Saffari, and J. R. Smith. Receiver selectivity limits on bistatic backscatter range. In *2020 IEEE International Conference on RFID (RFID)*, pages 1–8, Sep. 2020.
- [12] Mohamad Katanbaf, Anthony Weinand, and Vamsi Talla. Simplifying backscatter deployment: Full-duplex lora backscatter. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 955–972. USENIX Association, Apr. 2021.
- [13] Xin Li, Li Teng, Hong Tang, Jingying Chen, Haoyu Wang, Yu Liu, Minfan Fu, and Junrui Liang. Vipsn: A vibration-powered iot platform. *IEEE Internet of Things Journal*, 2020.
- [14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.
- [15] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation, 2018.
- [16] Vincent Liu, Aaron Parks, Vamsi Talla, Shyamnath Gollakota, David Wetherall, and Joshua R. Smith. Ambient backscatter: Wireless communication out of thin air. *SIGCOMM '13*.
- [17] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21–37, 2016.
- [18] Michele Magno, Xiaying Wang, Manuel Eggimann, Lukas Cavigelli, and Luca Benini. InfiniWolf: Energy efficient smart bracelet for edge computing with dual source energy harvesting. In *2020 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, mar 2020.
- [19] Ivan Mutis, Abhijeet Ambekar, and Virat Joshi. Real-time space occupancy sensing and human motion analysis using deep learning for indoor air quality control. *Automation in Construction*, 116:103237, 2020.
- [20] Saman Naderiparizi, Zerina Kapetanovic, and Joshua R. Smith. Wispcam: An rf-powered smart camera for machine vision applications. In *Proceedings of the 4th International Workshop on Energy Harvesting and Energy-Neutral Sensing Systems*, pages 19–22. ACM, 2016.
- [21] Saman Naderiparizi, Aaron N. Parks, Zerina Kapetanovic, Benjamin Ransford, and Joshua R. Smith. WISPCam: A battery-free RFID camera. In *2015 IEEE International Conference on RFID (RFID)*. IEEE, apr 2015.
- [22] Saman Naderiparizi, Yi Zhao, James Youngquist, Alanson P Sample, and Joshua R. Smith. Self-localizing battery-free cameras. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 445–449. ACM, 2015.
- [23] Ali Saffari. Widescatter: Toward wide area battery-free wireless sensor networks. Master's thesis, 2021. Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-05-13.
- [24] A. Saffari, M. Hessar, S. Naderiparizi, and J. R. Smith. Battery-free wireless video streaming camera system. In *in Proc. IEEE RFID*, 2019.
- [25] Homagni Saha, Sin Yong Tan, Ali Saffari, Mohamad Katanbaf, Joshua R. Smith, and Soumik Sarkar. Few shot clustering for indoor occupancy detection with extremely low-quality images from battery free cameras, 2020.
- [26] A.P. Sample, D.J. Yeager, P.S. Powlledge, A.V. Mamishev, and J.R. Smith. Design of an RFID-based battery-free programmable sensing platform. *IEEE Transactions on Instrumentation and Measurement*, 57(11):2608–2615, nov 2008.
- [27] Savath Saypadith, Watchara Ruangsang, and Supavadee Aramvith. Optimized human detection on the embedded computer vision system. In *2017 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pages 1707–1711. IEEE, 2017.
- [28] Robin Schrijvers, Steven Puttemans, Timothy Callemeyn, and Toon Goedemé. Real-time embedded person detection and tracking for shopping behaviour analysis. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 541–553. Springer, 2020.
- [29] Yu Song, Jihong Min, You Yu, Haobin Wang, Yiran Yang, Haixia Zhang, and Wei Gao. Wireless battery-free wearable sweat sensor powered by human motion. *Science Advances*, 6(40):eaay9842, sep 2020.
- [30] Sin Yong Tan, Homagni Saha, Anthony R Florita, Gregor P Henze, and Soumik Sarkar. A flexible framework for building occupancy detection using spatiotemporal pattern networks. In *2019 American Control Conference (ACC)*, pages 5884–5889. IEEE, 2019.
- [31] Sin Yong Tan, Homagni Saha, Margarite Jacoby, Anthony R Florita, Gregor P Henze, and Soumik Sarkar. Granger causality-based hierarchical time series clustering for state estimation. *IFAC-PapersOnLine*, 51(1), 2020.
- [32] Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, and Jun-Wei Hsieh. Cspnet: A new backbone that can enhance learning capability of cnn, 2019.
- [33] Junqi Wang, Jingjing Huang, Zhuangbo Feng, Shi-Jie Cao, and Fariborz Haghighat. Occupant-density-detection based energy efficient ventilation system: Prevention of infection transmission. *Energy and Buildings*, 240:110883, 2021.
- [34] Yaqing Wang, Quanming Yao, James Kwok, and Lionel M. Ni. Generalizing from a few examples: A survey on few-shot learning, 2020.
- [35] Libo Wu and Ya Wang. A low-power electric-mechanical driving approach for true occupancy detection using a shuttered passive infrared sensor. *IEEE Sensors Journal*, 19(1):47–57, 2018.