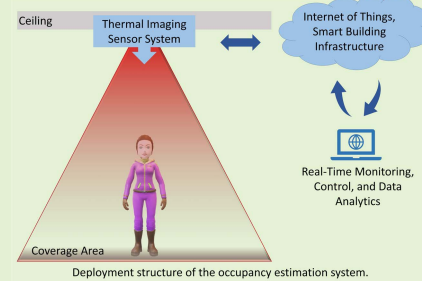


Occupancy Estimation Using Thermal Imaging Sensors and Machine Learning Algorithms

Veena Chidurala^{ID}, Graduate Student Member, IEEE, and Xinrong Li^{ID}, Senior Member, IEEE

Abstract—Occupancy estimation has a broad range of applications in security, surveillance, traffic and resource management in smart building environments. Low-resolution thermal imaging sensors can be used for real-time non-intrusive occupancy estimation. Such sensors have a resolution that is too low to identify occupants, but it may provide sufficient data for real-time occupancy estimation. In this paper, we present a systematic study of three thermal imaging sensors with different resolutions, with a focus on sensor characterization, estimation algorithms, and comparative analysis of occupancy estimation performance. A unified processing algorithms pipeline for occupancy estimation is presented and the performance of three sensors are compared side-by-side. A number of specific algorithms are proposed for pre-processing of sensor data, feature extraction, and fine-tuning of the occupancy estimation algorithms. Our results show that it is possible to achieve about 99% accuracy for occupancy estimation with our proposed approach, which might be sufficient for many practical smart building applications.

Index Terms—Classification, infrared array sensor, occupancy estimation, thermal imaging.



I. INTRODUCTION

OCCUPANCY estimation is an important research topic as it can be applied to a broad range of application domains such as security, surveillance, traffic and resource management, especially in smart building environments. For example, the heating, ventilation, and air conditioning (HVAC) systems can be controlled based on occupancy information to optimize the balance between occupants comfort and energy consumption. About 13% of all energy produced in the United States is used by the HVAC systems, with much of it being wasted when buildings are unoccupied or not fully occupied [1]. Real-time occupancy information and traffic patterns in large buildings such as libraries, museums, and business centers can also be used to optimize the usage efficiency of the space, shared facilities, and building resources.

Building occupancy has been traditionally monitored either through break-beam motion sensor-based door counters, which provide too little information to conduct further occupancy analysis, or through surveillance video cameras that invade

privacy. In this paper, we focus on low-resolution infrared (IR) array thermal imaging sensor-based techniques for real-time non-intrusive occupancy estimation. The low-resolution thermal imaging sensors have a resolution that is too low to identify occupants, but it may provide sufficient data for real-time occupancy estimation in certain application scenarios. A number of recent studies demonstrated promising results, see for example [2]–[11], [11], [13]–[15] and the references therein. The focuses of the existing studies vary in terms of the type of sensors employed, deployment structures, estimation and detection algorithms, and application scenarios. Some of the most relevant papers in the literature are reviewed briefly in Section II.

In this paper, we present a systematic study of three low-resolution thermal imaging sensors, including GridEye, MLX90640, and Lepton, with a focus on sensor characterization, estimation algorithms, and comparative analysis of occupancy estimation performance. To the best of our knowledge, this is the first time these sensors are studied side-by-side with a unified framework. The GridEye has a low resolution, making it impossible to detect identifiable features of human objects, hence it has no concerns of privacy invasion. The MLX90640 has a higher resolution than the GridEye, but still it is hard to distinguish the identifiable fine features of human objects. The Lepton sensor has the highest resolution among the three alternatives and it is possible to identify some body parts details such as hands,

Manuscript received December 12, 2020; accepted December 30, 2020. Date of publication January 5, 2021; date of current version February 17, 2021. The associate editor coordinating the review of this article and approving it for publication was Prof. Subhas C. Mukhopadhyay. (Corresponding author: Xinrong Li.)

The authors are with the Department of Electrical Engineering, University of North Texas, Denton, TX 75025 USA (e-mail: veenachidurala@my.unt.edu; xinrong.li@unt.edu).

Digital Object Identifier 10.1109/JSEN.2021.3049311

legs, and head as well as the profile outline of human objects.

A number of algorithms are proposed in this paper for sensor data pre-processing, feature extraction, and fine-tuning of occupancy estimation algorithms to address the issues that are resulted from the differences in sensor resolution and sensor characteristics. Contributions of this paper include:

- Proposed a unified processing algorithms pipeline for occupancy estimation using low-resolution thermal imaging sensors, which makes it possible to perform systematic analysis and comparison of three different sensors.
- Proposed a blob filtering algorithm to be used together with the connected components analysis and demonstrated that the proposed filtering algorithm works well even when two people stand very close to each other. The output of the filtering algorithm is then used as part of the feature vector when applying classification algorithms.
- Proposed a bilinear interpolation method to reduce noise in a noisy chess board pattern condition in the MLX90640 sensor data, which is resulted from the sensor's inherent design of reading patterns.
- Studied performance of the proposed algorithms using measurement data. Compared performance of four different feature vector designs and four different classification algorithms in detecting up to three human objects within the sensor coverage area. Demonstrated that the proposed feature vector design can effectively improve the classification performance as compared to other alternatives.

Our results show that it is possible to achieve about 99% accuracy for occupancy estimation with our proposed approach, which might be sufficient for many practical smart building applications. In addition, the non-intrusive nature of the low-resolution thermal imaging sensor-based techniques makes it possible to expand occupancy monitoring capabilities into private spaces, which might be extremely useful for applications in assisted living facilities, where occupants need to be monitored for emergency conditions at all times, including in the restroom or shower.

The rest of the paper is organized as follows. In Section II, we present a brief review of a number of most relevant papers in the literature. The overall system architecture, IR sensors and sensor system are discussed in Section III and Section IV, respectively. Then, the occupancy estimation algorithms are presented in detail in Section V, followed by the results and analysis in Section VI. Finally, the paper is closed with a summary of conclusions in Section VII and a discussion of limitations and future work in Section VIII.

II. RELATED WORKS

In the literature, a number of research papers presented occupancy estimation algorithms and systems using thermal imaging sensors. The existing studies mostly used the GridEye sensor while a few recent studies used the MLX90640 sensor, which became available in the market only in the past few years. For example, in one of the early studies presented in [3], the authors proposed an occupancy monitoring system that utilizes the GridEye and PIR motion sensors. They also presented an active pixel detection algorithm that is based

on background subtraction and simple thresholding methods. As discussed in [10], such an approach is ineffective in certain scenarios as it may over-filter a frame, leading to the loss of useful information and thus ineffective feature vectors. The features used in the paper include total active points, number of connected components, and size of the largest component. The classification algorithms used in the paper include k-nearest neighbors, linear regression and artificial neural networks.

The paper [8] used the same set of features for classification as the paper [3] and used image subtraction for active pixel detection to a different thermal imaging sensor MLX90620. Simple background subtraction is ineffective in filtering active pixels as they may still contain background noise. Hence, a threshold is added in our approach for better active pixel detection performance. The classification algorithms used in [8] include k-nearest neighbors, linear regression, and a multi-layer perceptron artificial neural network. According to the results therein, the accuracies achieved are at the 80%+ range. Both [3] and [8] showed challenges in identifying multiple human objects within one single image frame.

In the paper [10], the authors proposed an occupancy estimation technique based on entry and exit events through doors by placing the GridEye sensor on the doorway. The accuracy achieved therein is around 92%. The proposed solution was able to identify humans in a frame when there is a minimum of a pixel-level gap between the objects. However, considering the busy flow times through the doorways and when the people are walking or staying in close proximity, the low resolution sensor will misidentify multiple human objects as one single object. Sometimes, it is also hard to find the pixel-level gap between objects. In our paper, we have considered more features to overcome such false classification cases. In addition, through a series of performance comparisons, we demonstrate in this paper that the MLX90640 exhibits better performance than the GridEye.

In the paper [14], the authors proposed a dynamic way of updating the pixel-level background and the threshold level to filter active pixels from a frame, which may also be used to remove non-human thermal objects. In their application scenarios, the authors considered sensor deployment on the side wall facing the scene with a side-view. In such a deployment, there is a chance of losing the useful human object information while filtering active pixels in the presence of non-human thermal objects of higher temperatures. The authors explained the hollow phenomenon caused in such cases and proposed a uni-modal threshold segmentation technique to reconstruct the hollow phenomenon. Having the sensor installed with a side-view, sensor's coverage will be limited if any object blocks the sensor and creates the occlusion scenarios. In this paper, we adopt a different deployment structure where the thermal imaging sensors are installed on the ceiling while facing towards the floor. Such a deployment is useful in monitoring occupancy and traffic within a limited coverage area such as doorways, specific seating or standing spaces, etc.

In the paper [2], the authors proposed a people detection approach using a blob detection technique to find the active frame. A blob detection method is applied and then followed by area adjustment to extract occupancy information.

The authors also discussed two challenging scenarios: 1) when people are standing too close to each other, they appear as one single blob, and 2) when people raise their hands in contrary to the general scenario of hands kept down while moving, extra blobs may be detected to cause performance degradation. To address the issues in the first case, we propose more effective features to achieve better occupancy estimation performance. For the second case, we have considered finding the connected components instead of using the blob detection technique, which reduces the need of finding and joining the blobs that belong to the same human object.

III. OVERALL SYSTEM ARCHITECTURE

In this paper, we consider application scenarios where the thermal imaging sensor is deployed on the ceiling facing downward to monitor the occupancy within the sensor's coverage area on the floor as shown in the figure in the abstract. With such a deployment, there are less chances to have occlusion situations as compared to the sensor deployment on the side walls, where the view of the thermal imaging sensor could be easily blocked by the person standing next to it. Our deployment structure is suitable for small-area occupancy estimation such as people counting at the doorway, space utilization monitoring for seating areas in library, traffic monitoring at the exhibition areas in museums, etc. It is also possible to deploy multiple sensors in a mesh structure to expand the coverage of the system to a much larger space.

As discussed in Section IV, thermal imaging sensors can be easily integrated into a microcontroller- or microprocessor-based embedded sensor systems. With recent advances in sensor networks, cyber-physical systems, and Internet of Things, it is possible to integrate the thermal imaging sensor system into the smart building cyber-infrastructure system through a number of different wireless communication and networking technologies [16]–[20]. Then, the real-time monitoring, control, and data analytics capabilities offered by such a system can be applied to various smart building applications.

IV. THERMAL IMAGING SENSORS AND SENSOR SYSTEM

A. The GridEye Thermal Imaging Sensor

The GridEye (AMG8833) is a low-resolution thermal imaging sensor with a resolution of 8×8 and a field of vision (FOV) of 60° [22]. The temperature measurement range is from 0°C to 80°C at high gain and -20°C to 100°C at low gain. The operating temperature range is from 0°C to 80°C at high gain and from -20°C to 80°C at low gain. With the placement of sensor on the ceiling at a height of about 3 m from the ground, the sensor will have a FOV of approximately an area of $2.5\text{ m} \times 2.5\text{ m}$. It comes with the I2C interface and operates typically at 3.0 V voltage with high performance and high gain or 5.0 V with high performance and low gain. The typical frame rate of the GridEye is 10 FPS or 1 FPS.

The GridEye sensor, when powered on from the shutdown mode, the sensor requires some time to warm up in order to get stable readings. The time required for the sensor to

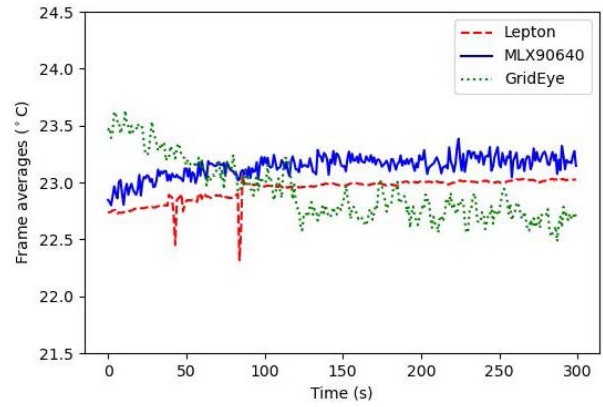


Fig. 1. Warmup time of three IR imaging sensors.

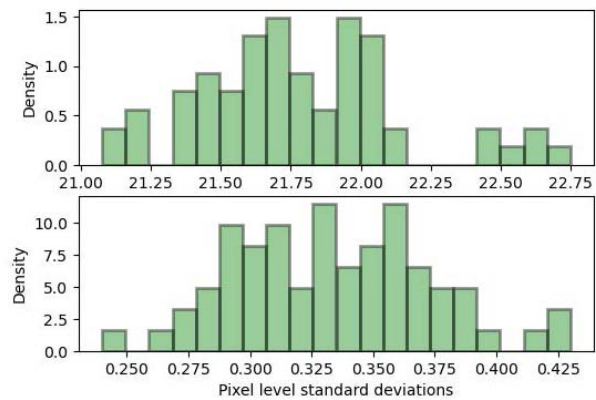


Fig. 2. Normalized histogram of GridEye pixel-level averages and standard deviations.

get stable is typically called as warmup time, setup time or settling time. It is important to let the sensor warm up for the given time to avoid false data detection. According to the datasheet, it takes 15 seconds for the sensor to get stabilized. However, an experiment is done in our lab to analyze the time required for its readings to get stable. The frames are captured continuously as soon as the sensor has been turned on for 300 seconds at a frame rate of 1 FPS. For this experiment the background was kept same and stable for all the measurement cases. The averages are calculated for each frame and is shown in Fig. 1. From the results, we can observe that it has taken approximately 1.5 minutes to see the readings from the sensor being stabilized, which is significantly longer than the number shown in the datasheet. So, it is a good practice to measure the warmup time before deploying the sensor in the field.

The temperature measurement accuracy of the GridEye is $\pm 2.5^\circ\text{C}$ according to its datasheet. A test was conducted to determine the noise levels of the GridEye. In the experiment, 100 frames were collected with a fixed and stable background during the measurements. The pixel-wise averages and standard deviations across all frames are calculated to observe the pixel-level noise. Fig. 2 shows the pixel-level averages and standard deviations of the sensor. From the figures, we can observe that the pixel-level noise of the GridEye lies within the range of $\pm 2.5^\circ\text{C}$ as stated in the datasheet.

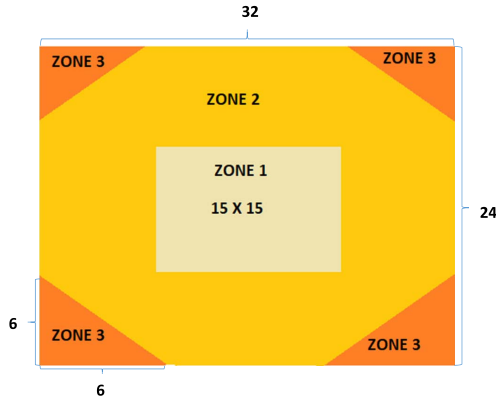


Fig. 3. MLX90640 subpage showing different measuring zones within a frame.

B. The MLX90640 Thermal Imaging Sensor

The MLX90640 is a thermal imaging sensor with a resolution of 32×24 and with two FOV options of $110^\circ \times 75^\circ$ and $55^\circ \times 35^\circ$ [23]. The sensor that is considered in this paper has a FOV of $110^\circ \times 75^\circ$. The operating temperatures are from -40°C to 85°C and the measuring temperatures are from -40°C to 300°C . With the sensor placement at a height of about 3 m on the ceiling facing the floor, the sensor FOV covers approximately an area of $4.5 \text{ m} \times 4 \text{ m}$. The sensor is compatible with I2C interface and operates with a supply voltage of 3.3 V. It comes with a programmable refresh rate from 0.5 Hz to 64 Hz. According to the datasheet, the warmup time to reach the accuracy in the sensor readings after it is electronically turned on is around 4 minutes. An experiment is performed to test the warmup time of the MLX90640 sensor. Once it is turned on from the shutdown mode, 300 frames were captured at a frame rate of 1 FPS. For each frame, the averages are calculated and plotted. Fig. 1 illustrates the trend of the sensor reading as it gets stabilized. Based on the observations, the warmup time of the sensor is about 4 minutes.

Each frame of the MLX90640 sensor measurement consists of two subpages, i.e., subpage 0 and subpage 1. Each subpage is divided into three zones as shown in Fig. 3 and the three zones have an accuracy of $\pm 1.5^\circ\text{C}$, $\pm 2^\circ\text{C}$, $\pm 3^\circ\text{C}$, respectively, with the Zone 1 to be the most reliable one and Zone 3 to have the worst performance.

According to the datasheet, the MLX90640 may suffer from long-term (years) drifts and thus it can experience an additional measurement deviation of $\pm 3^\circ\text{C}$ for the objects at the room temperature. An experiment is conducted to determine the pixel-level noise of the sensor, for which around 100 frames were captured and then the pixel-wise averages and standard deviations are obtained. Fig. 4 shows the pixel-wise averages and standard deviations of the sensor. Considering the maximum temperature noise levels that are possible in three zones, we can safely assume that the pixel-level noise of the MLX90640 lies within the range of $\pm 3^\circ\text{C}$.

C. FLIR Lepton Long Wave Infrared Imaging Sensor

We have used the Lepton 2.0 in this paper, it is a long-wave infrared thermal imaging sensor that comes with a resolution

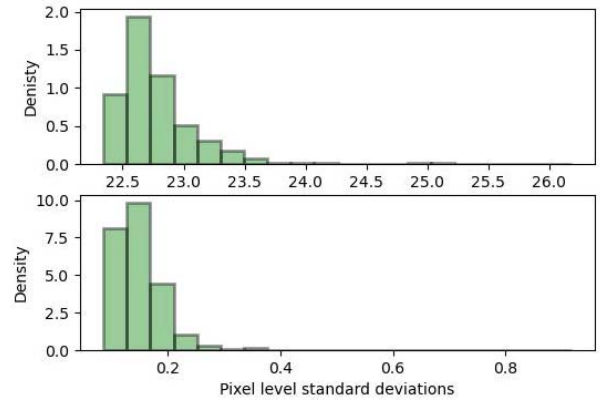


Fig. 4. Normalized histogram of MLX90640 pixel-level averages and standard deviations.

of 80×60 and the FOV of 51° and 60° diagonally [24]. The Lepton costs more than the GridEye and MLX90640. With the placement of sensor on the ceiling at a height of about 3 m while facing ground, it covers approximately an area of $2.5 \text{ m} \times 2.5 \text{ m}$. The sensor is compatible with both SPI and I2C interfaces. It is operated with a number of different power supplies, including 2.8 V to sensor, 1.2 V to digital core, and a flexible 2.5 V to 3.1 V to digital input and output. The frame rate of the sensor is less than 9 Hz. The raw sensor data is converted to the unit of degree Celsius with a linear conversion formula that is derived through calibration. There is no warmup time specified in the datasheet. To determine the sensor warmup time, 300 frames have been captured at a frame rate of 1 FPS and the average of each frame has been calculated. Fig. 1 shows the frame average versus time. From the results, we can observe that the warmup time required is approximately less than 2 minutes.

The sensor comes with high temperature coefficients of resistance and low noise, resulting in excellent thermal sensitivity and stable uniformity, but there are no specific noise levels mentioned for the sensor in the datasheet. To measure the noise levels of the sensor, 100 frames are captured using the sensor and then used to calculate pixel-level averages and standard deviations of all the frames. Fig. 5 shows the pixel-level averages and standard deviations of the sensor. The sensor's pixel-level noise appears to be very small.

D. Comparison of Thermal Imaging Sensors

The sensors considered in this paper are "uncooled" microbolometer type sensors, while many more accurate and more sensitive IR thermal imaging sensors are cooled, which are significantly more expensive and consume more power [29], [30]. Among the three thermal imaging sensors considered in this paper, the GridEye comes with the lowest resolution and it is impossible to use it to identify human objects. Choosing the GridEye for occupancy estimation solves the problem of privacy invasion. But because of its low resolution, whenever there are multiple human objects in the frame that are in close proximity to each other, they may appear as one blob, which will reduce the occupancy estimation accuracy. We will discuss two filtering techniques

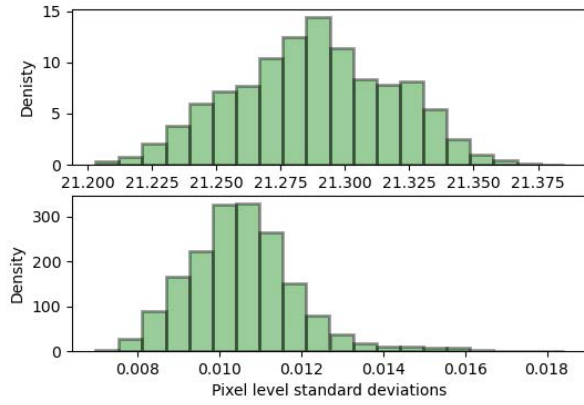


Fig. 5. Normalized histogram of Lepton pixel-level averages and standard deviations.

in Section V to obtain more accurate occupancy estimate performance when multiple human objects appear as one large blob in the data frame.

The MLX90640 sensor comes with a resolution that lies in between the resolutions of the GridEye and Lepton. It can identify human objects to some extent when it is held closer to the object. But it is impossible to identify the features of human object when it is installed on the ceiling that is 3 m above the floor. Compared to the other sensors, the MLX90640 is prone to more noise especially towards corners, which is partly due to the larger FOV and the sensor's reading patterns. Since the sensor showed undesirable noise characteristics during the measurement, it motivated us to perform further signal processing to reduce the noise. The following section contains a detailed discussion about a noise mitigation solution and its performance.

As compared to the GridEye and MLX90640 sensors, the Lepton has relatively higher resolution and it may not provide proper privacy protection. It is capable of identifying human body parts even when it is installed on the ceiling that is 3 m above the floor. When the sensor is held closer to the object, it can distinguish the features of human object and hence is able to identify the person to some extent. Working with Lepton has raised new concerns unlike the other two sensors. Because of its high resolution, the Lepton can capture the details of the human body such as limbs, head, face etc. It is observed from the measurement data that human body exhibits different temperatures patterns at different body parts. More discussions on this regard are presented in Section VI.

E. Noise Reduction Method for the MLX90640

The MLX90640 sensor has two types of frame reading patterns, i.e., chess pattern mode (factory default) and TV interleave mode [23]. The sensor is factory-calibrated in chess pattern mode, which results in better fixed pattern noise behavior. In both reading patterns, the array frame is divided into two subpages, i.e., subpage 0 and subpage 1, and the two subpages are read one after another in an alternating mode. Such a reading pattern results in a time delay between two subpages and thus a position offset of the human object detected by the sensor when the object is in motion.

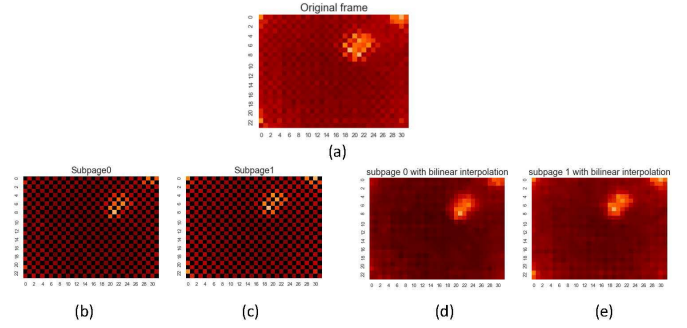


Fig. 6. Effects of bilinear interpolation to reduce the chess board pattern noise in the MLX90640 sensor data frame.

Fig. 6 illustrates the effect of the alternating subpage reading pattern. Specifically, Fig. 6a shows a data frame read from the sensor while the sensor is deployed on the ceiling facing downward to the floor. The light colored blob in the middle of the frame corresponds to a human object standing on the floor. We can see the chess board pattern inside the human object blob, which could degrade the performance of occupancy estimation algorithms. Fig. 6b and Fig. 6c are the subpage 0 and subpage 1 that are extracted from the original sensor data frame, respectively. Then, we applied the bilinear interpolation method to both subpages to interpolate the missing pixel values as shown in Fig. 6d and Fig. 6e. We can observe the difference between the two interpolated subpages in terms of the shape and position of the human object blob in the frame. Both interpolated subpages appear to be less noisy than the original data frame in Fig. 6a. So, in this research, we apply the bilinear interpolation method to subpages before further processing for occupancy estimation. The bilinear interpolation method can be implemented as a 2D image filter with the kernel

$$k_{BI} = \frac{1}{4} \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (1)$$

and the origin at the center of the kernel.

F. Sensor System Design

All three sensors described in this section provide convenient serial interfaces such as I2C and SPI. Hence, it is quite straightforward to integrate the sensors in any microcontroller or microprocessor system [17], [18]. In this research, we have used the Raspberry Pi 3 (RPI) as development platform. The RPI is a widely used low-cost single board computer that has sufficient computing power for many sensing, processing, and control applications [17], [20], [21]. RPI provides easy access to a 40-pin GPIO expansion header, which makes it easy to integrate various sensors into the system. In addition, RPI supports almost all standard networking technologies including Wi-Fi and Bluetooth. Therefore, the sensor system designed based on RPI can be either used as a standalone device or integrated easily into the cyber-infrastructure of a sensor network, cyber-physical system, or Internet of Things (IoT) through various standard networking technologies.

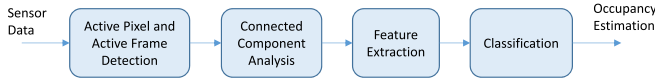


Fig. 7. Block diagram of occupancy estimation algorithm.

V. OCCUPANCY ESTIMATION ALGORITHMS

The workflow of occupancy estimation algorithms consists of four major functional blocks as shown in Fig. 7. The sensor data acquired from thermal imaging sensors are first used to detect active pixels and active frames. Active pixels are defined to be the pixels that are corresponding to human objects within a frame of sensor data. Active frame is defined to be the frame that has human objects captured. In contrast, a frame is defined as non-active background frame if all pixels on the frame correspond to background noise of the environment. A connected component analysis algorithm is then applied to the active frame to cluster active pixels into a number of blobs where each blob potentially only corresponds to one human object. Based on the results from the first two functional blocks, a set of features are extracted and used in classification algorithms to derive the final occupancy estimation. In the following, the occupancy estimation algorithms that we have designed are described in detail.

A. Active Pixel and Active Frame Detection

As the first step of the algorithm, each one of the pixels of a sensor data frame is classified as either active pixel or non-active background pixel through a binary thresholding process. To derive a suitable threshold for a particular sensor, we first collect N number of background-only data frames, $G^{(n)}$, $0 \leq n < N$, and then derive the pixel-wise threshold frame as

$$T_a(i, j) = \frac{1}{N} \sum_{n=0}^{N-1} G^{(n)}(i, j) + \delta_a, \quad (2)$$

where δ_a is a thresholding margin for active pixel detection and it is determined experimentally for each type of the thermal imaging sensors that we study in this research. In our experiments, we have typically set $N = 300$.

Once the threshold frame is determined, each of the pixels in a given sensor data frame is classified as an active pixel if its value is greater than the corresponding pixel value of the threshold frame T_a in (2); otherwise, it is classified as a non-active background pixel. The frame with active pixels is then classified as an active frame and it is passed to the connected components analysis algorithm for further processing; otherwise, it is classified as a non-active background frame and occupancy estimation algorithm terminates with an output of 0 as the estimated number of occupants.

B. Connected Components Analysis

Following the active pixel and active frame detection stage, an active frame is then binarized with the threshold in (2) and the connected components labeling is performed to group active pixels into a number of disconnected blobs. Two pixels

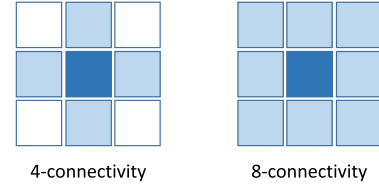


Fig. 8. Illustration of 4- and 8-connectivities in connected component analysis.

are considered connected if they are neighbors and have the same value. For a 2D image frame, there are two different types of connectivity that are commonly used, including 4-connectivity and 8-connectivity, as shown in Fig. 8. For the 4-connectivity, neighbors are found only in vertical and horizontal directions, while for the 8-connectivity, neighbors are also found in the diagonal directions. In this research, we have used 8-connectivity without loss of generality as we have not seen any noticeable performance differences when using these two options in the analysis of our sensor data. The connected components labeling-based blob extraction algorithm is listed in Algorithm 1. The connected components labeling algorithms are well studied in the literature and the ready-to-use implementations are available in many image processing software such as Matlab, OpenCV, and Python scikit-image.

Algorithm 1 Blob Extraction Algorithm

▷ F : data frame
 ▷ T : pixel-wise threshold frame
 1: **function** BLOBEXTRACTION(F, T)
 2: $F \leftarrow$ binarize F with the threshold frame T
 3: $L \leftarrow$ connected components labeling on F
 4: **return** L
 5: **end function**

When multiple human objects stand close to each other under the thermal imaging sensor, the active pixels corresponding to different human objects will be all connected into a large-sized blob. So, in this research, we designed a recursive blob filtering algorithm to further process a large-sized blob to split it into two or more smaller sized blobs. As demonstrated later in this paper, such a blob filtering algorithm helps to derive more effective features to achieve better performance at the classification stage for occupancy estimation. The recursive blob filtering algorithm is listed in Algorithm 2.

The initial input data to the blob filtering algorithm are the original sensor data frame F and the pixel-wise threshold frame T_a for active pixel detection that is defined in (2). Then, the blob extraction algorithm is first applied to the active frame to perform connected components labeling. Following that, large blobs are filtered recursively by applying the same blob filtering algorithm until the size of all blobs are within the range of $(s_{1,\min}, s_{1,\max})$, where $s_{1,\min}$ and $s_{1,\max}$ are the minimum and maximum blob sizes of one person, respectively; $s_{1,\min}$ and $s_{1,\max}$ are determined experimentally through extensive measurements for each of the thermal imaging sensors that we study in this research.

Algorithm 2 Blob Filtering Algorithm

```

▷  $F$ : data frame
▷  $T$ : pixel-wise threshold frame
1: function BLOBFILTERING( $F, T$ )
2:    $L \leftarrow \text{BLOBEXTRACTION}(F, T)$ 
3:   for each blob  $L^{(i)}$  in  $L$  do
4:     if blob size of  $L^{(i)} < s_{1,\min}$  then
5:        $L^{(i)}$  is ignored ▷ too small; treat as noise
6:     else if blob size of  $L^{(i)} > s_{1,\max}$  then
7:        $F_b \leftarrow$  data frame corresponding to blob  $L^{(i)}$ 
8:       ▷  $F_b$  is extracted from  $F$  with  $L^{(i)}$  as mask
9:        $T_b \leftarrow T + \delta_b$  ▷ increase the threshold
10:       $L_b \leftarrow \text{BLOBFILTERING}(F_b, T_b)$ 
11:      record all blobs in  $L_b$  as valid blobs
12:     else
13:       record blob  $L^{(i)}$  as a valid blob
14:     end if
15:   end for
16:   return valid blobs
17: end function

```

TABLE I

PARAMETERS USED IN THE BLOB FILTERING ALGORITHM

	δ_a	δ_b	$s_{1,\min}$	$s_{1,\max}$
Grid-EYE	0.75°C	0.25°C	4%	15%
MLX90640	1.00°C	0.25°C	2%	6%
Lepton	1.00°C	0.25°C	4%	15%

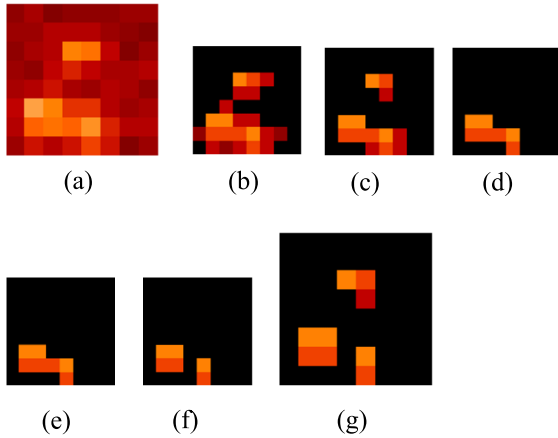


Fig. 9. GridEye stages of filtering with the original data frame in (a), the intermediate filtering results in (b)–(f), and the final output of the filtering algorithm in (g).

The blob filtering algorithm parameter values used in this research are summarized in Table I, where both $s_{1,\min}$ and $s_{1,\max}$ are given in percentage of the frame size, i.e., the total number of pixels of each thermal imaging sensor data frame. Fig. 9 demonstrates the intermediate stages of filtering process proposed in the paper, where frame (a) is the original sensor data frame, frame (b) is the active pixels frame using the proposed pixel level thresholding method, frame (c) is the result of the first level of filtering, with the threshold that is increased by δ_b , when applied to the active pixels frame, frame (d) is the result of the second level of filtering, with a

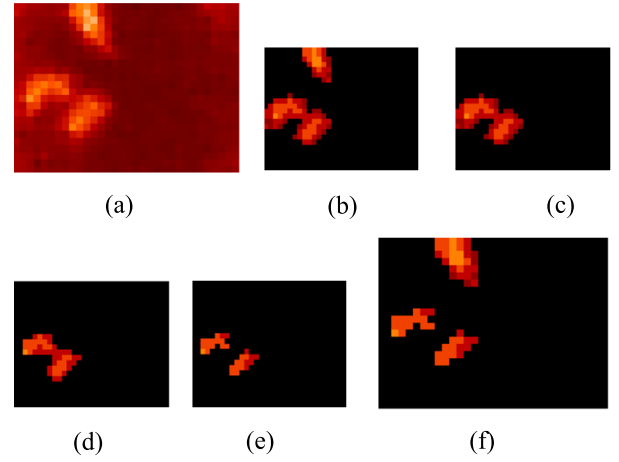


Fig. 10. MLX90640 stages of filtering with the original data frame in (a), the intermediate filtering results in (b)–(e), and the final output of the filtering algorithm in (f).

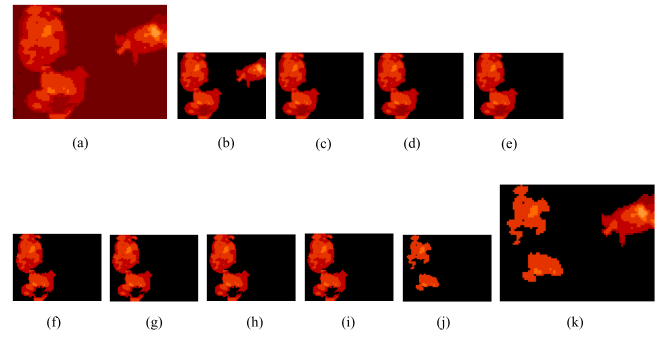


Fig. 11. Lepton stages of filtering with the original data frame in (a), the intermediate filtering results in (b)–(j), and the final output of the filtering algorithm in (k).

threshold that is increased by $2\delta_b$, when applied to the blobs that is larger than $s_{1,\max}$. Frames (e) and (f) are the results of the third and fourth levels of filtering with the thresholds that are increased by $3\delta_b$ and $4\delta_b$, respectively. We can clearly observe that in frame (f) the large blob from the previous stage has now been separated into two blobs, which accurately reflects the fact that there are two human objects standing close to each other, creating a large blob in the original data frame. Frame (g) shows all of the blobs that are detected using the recursive blob filtering algorithm, which is also the final output of the blob filtering algorithm.

Fig. 10 illustrates the sequence of blob filtering stages of the MLX90640. Frame (a) is the data frame that is derived using bilinear interpolation method on the original data frame while frame (b) is the active pixels frame using the proposed pixel-level thresholding method. Frame (c) is the frame showing the large blobs that require further filtering. Frame (d) is the result of the first level filtering with the threshold that is increased by δ_b . Frame (e) is the result of the second level filtering with a threshold that is increased by $2\delta_b$. We can clearly observe that the large blob from the previous stage is now separated into two smaller ones that correspond to two human objects. Frame (f) shows all of the blobs that are detected using the recursive blob filtering algorithm.

In Fig. 11, frame (a) shows the original sensor data frame obtained using the Lepton thermal imaging sensor. Frame (b) is the active pixels frame using the proposed pixel-level thresholding method. Frame (c) contains the large blob that requires further filtering. Frame (d) is the result of the first level of filtering with a threshold that is increased by δ_b . Frames (e)-(j) are a sequence of results of blob filtering with an increasing level of threshold, where frame (j) corresponds to the result of the seventh level of filtering with a threshold that is increased by $7\delta_b$. We can clearly observe that two human objects are now represented by two separate blobs in frame (j). Finally, frame (k) shows the final output of the recursive filtering algorithm.

Fig. 9 - Fig. 11 show the filtering stages of the data frames that are captured using three sensors in similar test scenarios, where the two people standing closer to each other are about 10 cm apart. We can observe that the proposed blob filtering algorithm can properly separate those two human objects. Additionally, since sensors are installed on a 3 m ceiling facing downward, human head is closer to sensors than shoulder and arms, which results in higher temperature values on the head and makes it possible to separate two closely located objects with a large gap after the filtering process as shown in Fig. 9g, Fig. 10f, and Fig. 11k. For the same reason, when using the MLX90640 and Lepton sensors, our blob filtering algorithm can work even when two people are lightly touching each other on the sides. However, for the GridEye sensor, due to its much lower resolution, it becomes much more challenging when two objects are less than 10 cm apart.

C. Feature Extraction

To apply classification algorithms to estimate the number of occupants from the thermal imaging sensor data frames, we have considered four types of features, including average of the pixel values m_d within a frame, standard deviation of the pixel values σ_d , total number of the active pixels n_a , and size of the three largest blobs $\{s(0), s(1), s(2)\}$. These features are defined in detail in the following in this section.

1) *Average and Standard Deviation of Pixel Values*: The thermal imaging sensors measure the temperature of the heat source objects. The pixel values of the sensor data frame directly correlate to the heat distribution in the environment that is being monitored. The more human objects stand under the sensor, the more pixels will have high values. Therefore, the average and standard deviation of the pixel values of the sensor data frame are good summative quantities for characterizing the distribution of pixel values within a data frame. In order to remove effects of the non-uniform pixel characteristics of the thermal imaging sensors, we have defined the features as the pixel-wise average and standard deviation of the differences of sensor data frame F and the threshold frame T_a given in (2); that is,

$$\begin{aligned} m_d &= \text{avg}\{F - T_a\}, \\ \sigma_d &= \text{std}\{F - T_a\}. \end{aligned} \quad (3)$$

2) *Total Number of Active Pixels*: A pixel in the data frame is classified as an active pixel if its value is greater than the

active pixel threshold that is defined in (2). The number of active pixels in a frame directly correlates to the number of human objects captured in a data frame; the more human objects captured in a frame, the higher the number of active pixels. So, the total number of active pixels n_a in a frame is selected as a feature to use in the classification algorithms to estimate the occupancy.

3) *Size of Three Largest Blobs*: Using the proposed recursive blob filtering technique in Algorithm 2, it is possible to split the large-sized blobs into smaller ones when multiple human objects are located in close proximity. In this research, we include the size of three largest blobs, i.e., $\{s(0), s(1), s(2)\}$, which are obtained after the blob filtering process, in the feature vector to use in the classification algorithms for occupancy estimation. When there are less than three blobs detected as valid blobs in Algorithm 2, the remaining blob size features are set to 0. In Section VI, we demonstrate the effectiveness of the proposed feature vector design through comparative analysis of the performance results that are obtained using different sets of features.

D. Classification Algorithms

Once we have defined the feature vector, it is quite straightforward to collect a set of training data using the thermal imaging sensors and then apply classification algorithms to the dataset. In this research, we have employed a number of machine learning algorithms that are commonly used in classification problems, including Gaussian naive Bayes, K-nearest neighbors, support vector machine, and random forests; for details of these algorithms see for example in the references [25]–[28]. Different machine learning models have their own advantages and disadvantages as well as different tuning requirements for the model hyperparameters. A brief description of these algorithms is presented in this section to provide a common basis for analysis of the performance results that are presented in the next section.

1) *Gaussian Naive Bayes*: Bayes classification algorithms are a group of supervised learning algorithms that are designed based on the Bayes theorem. With the Bayesian classification framework, the classifier first determines the posterior probabilities of the class labels, l_i , $0 \leq i \leq M - 1$, given some observed feature vector data x , that is, $P(l_i|x)$. Then, the label that has the largest posterior probability is determined as the predicted label of the observation data x . The Bayes theorem provides a way to compute the posterior probabilities for each of the class labels; that is,

$$P(l_i|x) = \frac{P(x|l_i)P(l_i)}{P(x)} = \frac{P(x|l_i)P(l_i)}{\sum_{i=0}^{M-1} P(x|l_i)P(l_i)}, \quad (4)$$

where $P(x|l_i)$ is a generative model (or the likelihood) of the data with the label l_i , $P(l_i)$ is the prior probability of the label l_i , and $P(x)$ is the probability of the observation data x . Such a Bayesian formulation naturally allows probabilistic classification and determination of classification uncertainty given the observation data.

Gaussian naive Bayes (GNB) classification method makes a simplistic assumption that the generative models $P(x|l_i)$ follow a multivariate Gaussian distribution and the feature

variables are conditionally independent. Thus, training of the classifier only involves finding the mean and variance in each dimension of the training data within each class. Given its simplicity and low requirements on computation and tuning, Gaussian naive Bayes method is often used as a baseline approach for many classification problems. It is also possible to extend the modeling capability of such a simplistic Bayes method by using more complex generative models such as Gaussian mixture models and kernel density estimation.

2) K-Nearest Neighbors: K-nearest neighbors (KNN) algorithm is another simple supervised machine learning method for classification problems. The underlying assumption of this algorithm is that similar things exist in close proximity. Hence, data points belonging to the same class stay close to each other, which means the unknown class label of the test data should be the same as that of its nearest training data points.

Training of the KNN model involves simply storing the training data. In prediction with test data, the algorithm finds k nearest neighbors of the test data in the training data, and then the majority class label among the k nearest neighbors are determined as the predicted class label of the test data. In the KNN algorithm, the Euclidean metric is commonly used as distance measure between data points when finding nearest neighbors. As a result of using Euclidean distance between multi-dimensional data points, the KNN algorithm is sensitive to scale differences along different dimensions. So, both training and test data need to be pre-processed to have similar scale along all dimensions of the feature vector.

3) Support Vector Machine: The support vector machine (SVM) is a discriminative classification method in the sense that it finds a decision boundary that is a line or curve in two dimensions or a manifold in multiple dimensions to divide the classes from each other. The training data points that are closest to the decision boundary are known as support vectors. The objective of training a SVM model is to find a decision boundary that maximizes the margin between support vectors and the decision boundary. In contrast to the KNN method, the SVM method works well with high dimensional data and it is very fast in prediction since the model is affected only by a small number of support vectors.

There are two approaches to enhance the modeling capability of SVM for complex classification problems. One approach is to use kernel transformation with, for example, radial basis function, to make it possible to derive a nonlinear decision boundary. Such an approach is often used in machine learning to turn fast linear methods into fast nonlinear methods. The other possible approach is to soften the margin to allow some training data to remain inside the margin if that improves prediction performance. Softening margin works as a regularization approach to avoid overfitting and to improve generalization performance. The softness of the margin can be tuned through cross-validation. One issue with the SVM is in that it does not provide probabilistic interpretation of the results as with the Bayes methods.

4) Random Forests: Random forests (RF) are ensemble methods that are built on a large number of randomized decision trees. Decision tree methods are widely used in classification problems. Building a decision tree essentially

means learning a hierarchy of binary if-else questions that leads to the true classification result. Decision tree methods are intuitive and easy to visualize and interpret, but they tend to easily overfit the data and provide poor generalization performance. Building random forests with decision trees provides an effective way to overcome these issues.

The fundamental idea behind random forests is that if many decision trees are built while all of them may overfit in different ways, it is possible to combine the decision trees by averaging or majority voting to reduce the effect of overfitting. The key to the effectiveness of such an ensemble method is in the mutual independence of the underlying decision trees. The trees in a random forest can be randomized by selecting a randomized subset of data points to build a tree and by randomizing the selection of the features in each binary split test. By training a random forest, it is also possible to determine feature importance by aggregating the feature importance over the trees in the forest. Random forests are currently among the most widely used machine learning methods in classification problems. They are very powerful, often work well without heavy tuning of the parameters, and don't require scaling of the data [28].

VI. RESULTS AND ANALYSIS

A. Experimental Setup and Performance Measures

The experimental setup used for data collection in this project is configured the same as the one shown in the abstract. That is, the IR imaging sensors are installed to face downward on the ceiling at a height of 3 m. Sensor readings have been collected for four different cases, including zero, one, two, and three human objects within the sensor coverage area. In each case, a total of 700 sensor data frames are collected at a frame rate of 1 FPS, which amounts to a total number of $3 \times 4 \times 700$ data for three sensors and four different cases per sensor. The test human objects include eight individuals of varying heights and sizes. When training and testing classification algorithms, 80% of the data are used as the training set while the remaining 20% of the data are used as the test set. Stratified sampling is used when splitting the data into the training and test sets. In the data pre-processing for MLX90640, raw data are used to derive bilinear-interpolated data, from which feature vector data are obtained. In addition, feature vector data are scaled (i.e., normalized) when working with the KNN and SVM classification algorithms, which is a necessary pre-processing step as mentioned in the previous section.

In this paper, performance of the classification algorithms has been measured and compared in terms of f_1 -scores and cross-validation scores [28]. The f_1 -score is the harmonic mean of precision and recall, which is defined as

$$f_1\text{-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (5)$$

Precision measures how many of the samples predicted as positive are actually positive and it is used when the goal is to limit the number of false positives. Recall measures how many of the positive samples are captured by the positive predictions and it is used when it is important to identify all positive samples and to avoid false negatives. Looking at any

TABLE II
F₁-SCORES FOR THE GRID EYE SENSOR

	FV1	FV2	FV3	FV4
GNB	0.8422	0.8124	0.9124	0.7015
KNN	0.8924	0.8812	0.9394	0.7069
SVM	0.8302	0.8375	0.9312	0.7131
RF	0.8857	0.9143	0.9518	0.9019

TABLE III
CROSS-VALIDATION SCORES FOR THE GRID EYE SENSOR

	FV1	FV2	FV3	FV4
GNB	0.8129	0.8014	0.9004	0.7071
KNN	0.8521	0.8914	0.9461	0.8825
SVM	0.8239	0.8354	0.9150	0.7671
RF	0.8454	0.8861	0.9436	0.8986

one of the measures of recall or precision individually will not provide a good overall performance assessment. The f_1 -score is a better measure to summarize both precision and recall, especially on imbalanced datasets.

Cross-validation is a statistical method of evaluating generalization performance which is more stable and thorough as compared to the typical accuracy measure on a single split of the available dataset to one training set and one test set. In such a method, data is split repeatedly and a model is trained and tested with each one of the splits. The most common cross-validation method is the k -fold cross-validation, where k is a number that is usually either 5 or 10. In the k -fold cross-validation, a series of models are trained and tested in k steps. At each step, one fold of data is held out as the test set and a model is trained with the remaining folds as the training set. In the end, accuracy scores are determined for each one of the k split cases. In this paper, we used 10-fold cross-validation, and the resulting 10 accuracy scores are averaged to summarize the cross-validation accuracies into one single performance measure for the ease of comparison.

B. Performance Comparison of Feature Vector Designs

In general, performance of classification algorithms depend heavily on the design of feature vector. As discussed in Section V-C, the feature vector proposed in this paper comprises four features, including the average and standard deviation of pixel values, the total number of active pixels, and the size of three largest blobs. To study the effectiveness of the proposed feature vector (FV), we compare the performance of four classification algorithms, including GNB, KNN, SVM, and RF, when using four different feature vector designs, from FV1 to FV4, as shown in Table II and Table III.

The first feature vector design (FV1) contains the average and standard deviation of pixel values. The second feature vector design (FV2) contains the average and standard deviation of pixel values, and the total number of active pixels. The third feature vector design (FV3) is the proposed feature vector that comprises four features. In addition, we have also implemented the feature vector (FV4) proposed in the reference [3], which comprises three features, including the total number of active

TABLE IV
COMPARISON OF F₁-SCORES

	GridEye	MLX90640	Lepton	Lepton-RR
GNB	0.9124	0.9839	0.8660	0.9014
KNN	0.9394	0.9911	0.9439	0.9570
SVM	0.9312	0.9696	0.9097	0.9505
RF	0.9518	0.9946	0.9572	0.9702

TABLE V
COMPARISON OF CROSS-VALIDATION SCORES

	GridEye	MLX90640	Lepton	Lepton-RR
GNB	0.9004	0.9704	0.8550	0.8693
KNN	0.9461	0.9850	0.9128	0.9429
SVM	0.9150	0.9671	0.9144	0.9300
RF	0.9436	0.9900	0.9561	0.9661

pixels, the number of connected components, and the size of the largest component. From the results shown in the tables, we can clearly observe that our proposed feature vector design FV3 has the best performance among the four alternatives. The feature vector design FV3 also shows the best performance on the data from the MLX90640 and Lepton sensors.

C. Performance Comparison of Classification Algorithms

To compare the performance of classification algorithms on the data from three different sensors, we have used the proposed feature vector FV3 and obtained the f_1 -scores and cross-validation scores for all combinations of classification algorithms and sensors as shown in Table IV and Table V. From the results in the first three columns of the tables, we can observe that the random forests method always performs better than the other methods when comparing the results for the same sensor. In addition, MLX90640 performs better than the other two sensors in our experimental scenarios. With the MLX90640 sensor and random forests method, the accuracy of occupancy estimation is about 99%, which might be sufficient for many practical smart building applications.

Performance differences among different sensors may be resulted from a number of factors, including differences in sensor noise level, sensitivity, field of vision, sensing coverage, and imaging resolution among possibly many others. Using our measurement data and performance results, we can briefly analyze the effects of different imaging resolutions on the classification performance. The imaging resolutions of GridEye, MLX90640 and Lepton are 8×8 , 32×24 , and 80×60 , respectively. From the images shown in Fig. 9, Fig. 10, and Fig. 11, we can observe that with the extremely low resolution of the GridEye sensor, it is impossible to recognize any identifiable human object features. However, with the increasing resolutions, more details of human objects can be captured using the MLX90640 and Lepton sensors. At the connected component analysis stage in the algorithm pipeline (i.e., Fig. 7), we have noticed that there is a possibility that different parts of a human object such as head, shoulder, and hands may be identified as different blobs by the blob filtering algorithm due to high variations in pixel values,

which may reduce the overall classification performance. It is also possible that when multiple human objects are in close proximity, parts of different human objects may be merged together and show up as one single blob.

To partly validate our conjecture, we derived a new set of reduced-resolution data from the raw Lepton measurement data by combining a block of four pixels into one pixel through averaging. The resolution of the new dataset is 40×30 , which is similar to that of MLX90640. The f_1 -scores and cross-validation scores of the new dataset, which is denoted as Lepton-RR, are shown in Table IV and Table V. By comparing the results, we can observe that the classification performance of the Lepton-RR data is better than that of the original Lepton data, and it is getting closer to that of MLX90640.

VII. SUMMARY AND CONCLUSION

This paper focuses on the comparative study of thermal imaging sensors for occupancy estimation. We first presented the warmup time and statistical noise level characteristics of the sensors based on measurements. For the MLX90640, we demonstrate that the sensor data frame may exhibit undesirable chess board patterns due to the two-subpage reading approach employed by the sensor, which may cause performance degradation when the objects are in motion. We employed a bilinear interpolation method to address the issue. One challenging issue in detecting the number of occupants is in distinguishing multiple human objects that are in close proximity to each other. We proposed a blob filtering algorithm to recursively split a blob when it is larger than the typical size of one single human object. A new feature vector design is proposed to use with the classification algorithms. We have demonstrated the effectiveness of the proposed methods by training and testing the algorithms with measurement data. Our performance results indicate that the proposed technique could be used in various smart building applications. It should be noted that due to the low-resolution and limited coverage area of the sensors, performance of the proposed technique depends on the specific deployment structure.

VIII. LIMITATIONS AND FUTURE WORK

This paper focuses on low-resolution thermal imaging sensors for occupancy estimation. However, in the literature, there are other sensing modalities that have also been considered for occupancy estimation. For example, the paper [31] presented a study on occupancy estimation using the CO₂ concentration level inside buildings. In addition to CO₂, the study presented in paper [4] also used temperature and ventilation actuation information that are commonly available in HVAC systems. In paper [5], a densely deployed sensor network of passive infrared (PIR) motion sensors was used to forecast building occupancy a short time (i.e., up to 60 minutes) into the future. The paper [11] used the number of WiFi devices, electrical energy demand and water consumption rate to estimate building occupancy. There are also studies on using acoustic measurements to locate, track, and estimate human objects such as [32]–[34], which could be useful for building occupancy estimation.

Different sensing modalities have their own advantages and limitations, and different sensors are suitable for different system configurations and application scenarios. The scenario that we have considered in this paper is the one where sensors are installed on the ceiling of one particular height, and no other types of sensors are integrated in deriving occupancy estimation. In practical implementation of system-level solutions, there are many issues that need to be addressed and many areas that could be optimized to meet application requirements. For example, it is highly desirable to integrate multiple sensing modalities and perform sensor fusion, employ multiple sensors at different deployment configurations (i.e., installing thermal imaging sensor on both ceiling and side walls), and deploy a distributed sensor network instead of a standalone sensor device, among many others. All of the aforementioned measures will make it possible to significantly improve overall system performance. One other area that we haven't touched upon in this paper is the potential impacts of various interfering heat-emitting objects such as personal electronic devices and pets. More in-depth studies and analyses are needed to fully characterize the impacts of such factors.

IR sensors are inherently susceptible to errors caused by thermal gradients and it is recommended not to subject thermal imaging sensors to heat transfer and especially transient conditions [23]. In real applications, however, such transient conditions may occur, e.g. because of opening windows and doors. So, the readers are cautioned that more studies are required to characterize the impacts of such conditions on system performance.

The classification algorithms that we have used in this paper are all among the more traditional machine learning algorithms. This research is a part of a larger scope project where we are exploring the broad field of occupancy estimation and smart building technologies where this research serves as a foundation. As a continuation of this research, we are currently working on more complex machine learning frameworks including convolutional and recurrent neural networks for more complex non-intrusive human behavior and scene analysis. Nevertheless, this paper could be useful for practitioners in the field as it provides a consistent framework to implement low-complexity occupancy estimation systems using a number of different low-resolution thermal imaging sensors.

REFERENCES

- [1] ARPA-E. (2017). *Saving Energy Nationwide in Structures With Occupancy Recognition*. Press Release, Department of Energy. [Online]. Available: <https://arpa-e.energy.gov/?q=arpa-e-programs/sensor>, 2017.
- [2] M. Berger and A. Armitage, "Room occupancy measurement using low-resolution infrared cameras," in *Proc. IET Irish Signals Syst. Conf. (ISSC)*, 2010, pp. 249–254.
- [3] A. Beltran, V. L. Erickson, and A. E. Cerpa, "ThermoSense: Occupancy thermal based sensing for HVAC control," in *Proc. 5th ACM Workshop Embedded Syst. Energy-Efficient Buildings - BuildSys*, 2013, pp. 1–8.
- [4] A. Ebadat, G. Bottegai, D. Varagnolo, B. Wahlberg, and K. H. Johansson, "Estimation of building occupancy levels through environmental signals deconvolution," in *Proc. 5th ACM Workshop Embedded Syst. Energy-Efficient Buildings - BuildSys*, 2013, pp. 1–8.
- [5] J. Howard and W. Hoff, "Forecasting building occupancy using sensor network data," in *Proc. 2nd Int. Workshop Big Data, Streams Heterogeneous Source Mining Algorithms, Syst., Program. Models Appl. BigMine*, 2013, pp. 87–94.

- [6] C. Basu and A. Rowe, "Tracking motion and proxemics using thermal-sensor array," 2015, *arXiv:1511.08166*. [Online]. Available: <http://arxiv.org/abs/1511.08166>
- [7] F. C. Sangoboye and M. B. Kjargaard, "PLCount: A probabilistic fusion algorithm for accurately estimating occupancy from 3D camera counts," in *Proc. 3rd ACM Int. Conf. Syst. Energy-Efficient Built Environ.*, Nov. 2016, pp. 147–156.
- [8] A. Tyndall, R. Cardell-Oliver, and A. Keating, "Occupancy estimation using a low-pixel count thermal imager," *IEEE Sensors J.*, vol. 16, no. 10, pp. 3784–3791, May 2016.
- [9] A. A. Trofimova, A. Masciadri, F. Veronese, and F. Salice, "Indoor human detection based on thermal array sensor data and adaptive background estimation," *J. Comput. Commun.*, vol. 5, no. 4, pp. 16–28, 2017.
- [10] H. Mohammadmoradi, S. Munir, O. Gnawali, and C. Shelton, "Measuring people-flow through doorways using easy-to-install IR array sensors," in *Proc. 13th Int. Conf. Distrib. Comput. Sensor Syst. (DCOSS)*, Jun. 2017, pp. 35–43.
- [11] A. K. Das, P. H. Pathak, J. Jee, C.-N. Chuah, and P. Mohapatra, "Non-intrusive multi-modal estimation of building occupancy," in *Proc. 15th ACM Conf. Embedded Netw. Sensor Syst.*, Nov. 2017, pp. 1–14.
- [12] G. Spasov, V. Tsvetkov, and G. Petrova, "Using IR array MLX90640 to build an IoT solution for ALL and security smart systems," in *Proc. IEEE 28th Int. Sci. Conf. Electron. (ET)*, Sep. 2019, pp. 1–4.
- [13] S. Savazzi, V. Rampa, S. Kianoush, A. Minora, and L. Costa, "Occupancy pattern recognition with infrared array sensors: A Bayesian approach to multi-body tracking," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019.
- [14] N. Gu, B. Yang, and T. Zhang, "Dynamic fuzzy background removal for indoor human target perception based on thermopile array sensor," *IEEE Sensors J.*, vol. 20, no. 1, pp. 67–76, Jan. 2020.
- [15] M. Maaspuro, "A low-resolution IR-array as a doorway occupancy counter in a smart building," *Int. J. Online Biomed. Eng. (iJOE)*, vol. 16, no. 6, p. 4, May 2020.
- [16] J. Yang, C. Zhang, X. Li, Y. Huang, S. Fu, and M. F. Acevedo, "Integration of wireless sensor networks in environmental monitoring cyber infrastructure," *Wireless Netw.*, vol. 16, no. 4, pp. 1091–1108, May 2010.
- [17] S. Ferdoush and X. Li, "Wireless sensor network system design using raspberry Pi and Arduino for environmental monitoring applications," *Procedia Comput. Sci.*, vol. 34, pp. 103–110, 2014.
- [18] S. Abraham and X. Li, "Design of a low-cost wireless indoor air quality sensor network system," *Int. J. Wireless Inf. Netw.*, vol. 23, no. 1, pp. 57–65, Mar. 2016.
- [19] J. P. Smith and X. Li, "AirSniffer: A smartphone-based sensor system for body area climate and air quality monitoring," in *Proc. 10th Int. Symp. Med. Inf. Commun. Technol. (ISMICT)*, Mar. 2016, pp. 1–5.
- [20] N. A. Ruprecht and X. Li, "Implementation of compressive sampling for wireless sensor network applications," *Int. J. Sensor Netw.*, vol. 31, no. 4, p. 226, 2019.
- [21] I. Charania and X. Li, "Smart farming: Agriculture's shift from a labor intensive to technology native industry," in *Internet Things*. Amsterdam, The Netherlands: Elsevier, 2020.
- [22] Panasonic Industry. *Infrared Array Sensor Grid-EYE*. Accessed: Aug. 2020. [Online]. Available: <https://industrial.panasonic.com>
- [23] Melexis. *Far Infrared Thermal Sensor Array*. Accessed: Aug. 2020. [Online]. Available: <https://www.melexis.com>
- [24] FLIR Systems. *Thermal by FLIR*. Accessed: Aug. 2020. [Online]. Available: <https://lepton.flir.com>
- [25] J. VanderPlas, *Python Data Science Handbook*. Newton, MA, USA: O'Reilly, 2016.
- [26] A. Geron, *Hands-On Machine Learning With Scikit-Learn & TensorFlow*. Newton, MA, USA: O'Reilly: O'Reilly, 2016.
- [27] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer, 2017.
- [28] A. C. Muller and S. Guido, *Introduction to Machine Learning With Python*. Newton, MA, USA: O'Reilly: O'Reilly, 2017.
- [29] K. Liddiard, "The active microbolometer: A new concept in infrared detection," *Proc. SPIE*, vol. 5274, pp. 227–238, Mar. 2004.
- [30] H. Wang, X. Yi, J. Lai, and Y. Li, "Fabricating microbolometer array on unplanar readout integrated circuit," *Int. J. Infr. Millim. Waves*, vol. 26, no. 5, pp. 751–762, May 2005.
- [31] A. Franco and F. Leccese, "Measurement of CO2 concentration for occupancy estimation in educational buildings with energy efficiency purposes," *J. Building Eng.*, vol. 32, Nov. 2020, Art. no. 101714.
- [32] J. George and L. M. Kaplan, "A finite point process approach to multi-target localization using transient measurements," *Inf. Fusion*, vol. 32, pp. 62–74, Nov. 2016.
- [33] T. Damarla, A. Mehmood, and J. Sabatier, "Detection of people and animals using non-imaging sensors," in *Proc. 14th Int. Conf. Inf. Fusion*, Chicago, IL, USA, Jul. 2011, pp. 1–8.
- [34] P. Chattopadhyay, A. Ray, and T. Damarla, "Simultaneous tracking and counting of targets in a sensor network," *J. Acoust. Soc. Amer.*, vol. 139, no. 4, p. 2108, Mar. 2016.



Veena Chidurala (Graduate Student Member, IEEE) received the Bachelor of Technology degree in electronics and communication engineering from Jawaharlal Nehru Technological University (JNTU), Hyderabad, India, in 2014, and the Master of Science degree in electrical engineering from the University of North Texas (UNT), Denton, TX, USA, in 2016, where she is currently pursuing the Ph.D. degree in electrical engineering. Her research interests include signal processing, computer vision, machine learning, and embedded systems.



Xinrong Li (Senior Member, IEEE) received the Bachelor of Engineering degree from the University of Science and Technology of China (USTC), Hefei, China, in 1995, the Master of Engineering degree from the National University of Singapore (NUS) in 1999, and the Ph.D. degree from the Worcester Polytechnic Institute (WPI), Worcester, MA, USA, in 2003, all in electrical engineering. From 2003 to 2004, he was a Postdoctoral Research Fellow with the Center for Wireless Information Network Studies, Worcester Polytechnic Institute. He has been working with the Department of Electrical Engineering, University of North Texas, Denton, TX, USA, as an Assistant Professor since 2004 and an Associate Professor since 2010. His research has been focused on signal processing, machine learning, embedded systems, sensor networks, and the Internet of Things.