DATA PROCESSING FOR DEVICE-FREE FINE-GRAINED OCCUPANCY SENSING USING INFRARED SENSORS

A Dissertation

by

ZHANGJIE CHEN

Submitted to the Graduate and Professional School of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Chair of Committee,	Ya Wang
Committee Members,	Bryan Rasmussen
	Cynthia Hipwell
	Zhang O'Neill
	Xia Hu
Head of Department,	Guillermo Aguilar

December 2021

Major Subject: Mechanical Engineering

ABSTRACT

Fine-grained occupancy information plays an essential role for various emerging applications in smart homes, such as personalized thermal comfort control and human behavior analysis. Existing occupancy sensors, such as passive infrared (PIR) sensors generally provide limited coarse information such as motion. However, the detection of fine-grained occupancy information such as stationary presence, posture, identification, and activity tracking can be enabled with the advance of sensor technologies. Among these, infrared sensing is a low-cost, device-free, and privacy-preserving choice that detects the fluctuation (PIR sensors) or the thermal profiles (thermopile array sensors) from objects' infrared radiation. This work focuses on developing data processing models towards fine-grained occupancy sensing using the synchronized low-energy electronically chopped PIR (SLEEPIR) sensor or the thermopile array sensors.

The main contributions of this dissertation include: (1) creating and validating the mathematical model of the SLEEPIR sensor output towards stationary occupancy detection; (2) developing the SLEEPIR detection algorithm using statistical features and long-short term memory (LSTM) deep learning; (3) building machine learning framework for posture detection and activity tracking using thermopile array sensors; and (4) creating convolutional neural network (CNN) models for facing direction detection and identification using thermopile array sensors.

DEDICATION

To my parents

ACKNOWLEDGEMENTS

First, I would like to express my deep gratitude to my advisor, Dr. Ya Wang, for offering me this opportunity of research and for her guidance, support, and encouragement throughout this journey. I appreciate all the time, thoughts, and passion she has contributed to making my Ph.D. experience productive and stimulating.

In addition, I am grateful to Dr. Cynthia Hipwell, Dr. Zheng O'Neill, Dr. Bryan Rasmussen, and Dr. Xia Hu for taking the time to serve as my committee members and for sharing their breadth of knowledge and expertise.

Large parts of this work could not have been done without help from my lab colleagues. I would like to thank Dr. Libo Wu and Tianhao Yan for their help fabricate liquid crystal (LC) shutters. Much appreciation to Qijie Shen for his significant contribution to data-saving software development. Thanks also go to my other lab colleagues, Rui Hua and Jingfan Chen for sharing their comments, suggestions, and supports.

I am also thankful to all my friends, the department faculty and staff for making my time at Texas A&M University a great experience.

Finally, thanks to my parents for their love, encouragement, and selfless support during my study overseas.

CONTRIBUTORS AND FUNDING SOURCES

Contributors

This work was supervised by a dissertation committee consisting of Dr. Ya Wang, as the advisor, Dr. Cynthia Hipwell, Dr. Bryan Rasmussen, Dr. Zheng O'Neil of the J. Mike Walker '66 Department of Mechanical Engineering, and Dr. Xia Hu of the Department of Computer Science and Engineering at the Texas A&M University.

The LC shutters in Chapter 3 were fabricated by Dr. Libo Wu of the J. Mike Walker '66 Department of Mechanical Engineering, and Tianhao Yan of the Department of Chemistry at the Texas A&M University. The Bluetooth software of SLEEPIR sensor system for dataset collection in Chapter 4 was developed by Qijie Shen of the Department of Electrical and Computer Engineering at the Texas A&M University.

All other work conducted for the dissertation was completed by the student independently.

Funding Sources

This research was supported by the U.S. Department of Energy, Advanced Research Projects Agency–Energy (ARPA-E) under grants DE-AR0000531 (2015-2018) and DE-AR0000945 (2018-2022).

TABLE OF CONTENTS

Page

LIST OF FIGURES

Page

LIST OF TABLES

Page

1. INTRODUCTION

1.1. Background and Motivation

1.1.1. Fine-grained Occupancy Sensing

Occupancy sensing technology is rapidly increasing worldwide, especially for Heating, Ventilation, and Air Conditioning (HVAC) control [1, 2]. HVAC systems represent the largest energy use accounting for approximately 50% of the total energy required to operate residential and commercial buildings [3] [4]. While building occupants are often dissatisfied with their thermal comfort despite this vast energy consumption. Primary causes include the lack of a reliable occupancy presence detection system. For example, commercial occupancy sensors rely on passive infrared (PIR) or PIR – ultrasonic dual technology sensors as occupancy presence detectors. Nevertheless, false-negative detection happens frequently as they only respond to the motion of the occupants. This leads to significant comfort loss and low energy efficiency and thus limits their broader applications.

In addition, fine-grained occupancy information such as stationary presence, posture, identity, and activity tracking, is crucial for advanced building applications such as patient/staff monitoring in hospitals[5], elder monitoring in assisted living facilities [6], sleep monitoring, fall detection in residential apartments [7] [8], or personalized conditioning in office buildings [9]. For instance, occupancy identification supports the room temperature setpoint adjusted based on occupancy preference and activity level[10, 11]. Furthermore, in-bed postures are critical for sleep quality monitoring [12, 13].

1.1.2. Existing Device-free Fine-grained Occupancy Sensing Systems

Lots of existing device-free sensing systems have been researched to extract fine-grained occupancy information, such as PIR sensors [14-20], optical cameras [21-23], environmental sensors [24-26], thermal infrared array sensors [27-30], and radio frequency (RF) sensors [31-34]. However, different drawbacks exist, such as privacy invasion, high power consumption, high computing cost, and low robustness.

PIR sensors have been studied to extract various occupancy information, such as non-stationary presence detection [14-17], counting [18], walking direction detection [19, 20], localization [35-38], sleep movement detection [39], and activity recognition [40-42]. However, their incapability for stationary occupancy detection has not ever been addressed.

Optical cameras can provide rich occupancy information. Existing surveillance cameras can detect the occupancy presence of an open office [21]. Moreover, occupancy activities can be identified by analyzing the posture recorded by the camera [23]. Despite the fine-grained occupancy information provided by optical cameras, they require good light condition and raise concerns on privacy invasion at the same time. Infrared night vision cameras work in a dark environment. However, concerns regarding an invasion of privacy still exist as the night vision camera gives high-resolution greyscale images. The depth camera is a good alternative for privacy-preserving, while it is costly and also sensitive to the ambient light condition [43]. In addition, both night vision and depth cameras have high power consumption as their active sensing elements require continuous emission of infrared signals.

Environmental sensors can detect indoor air qualities such as CO_2 concentration, which correlates highly to occupancy state. Multiple environmental sensors are usually necessary for occupancy detection. For example, light, temperature, and humidity sensors are commonly fused with CO_2 sensors to detect the occupancy presence in an office room [26]. In addition, occupancy presence and counting within a residential home can be detected by measuring CO_2 and total volatile organic compounds [25]. One major drawback of using environmental sensors is that the indoor air quality in response to occupancy activity is very slow and also sensitive to factors other than occupancy activity, such as the air exchange caused by the ventilation system.

Extracting channel information from RF signals is another approach for occupancy sensing. For example, existing Wi-Fi devices, the most common RF infrastructure in buildings, can be used together with occupant-carried mobile devices for occupancy detection and counting on a whole building floor [31]. In addition, occupancy activities such as standing and sitting can be recognized using the channel state information based on two Wi-Fi access points around the desk [32]. On the other hand, research also points out that the performance of Wi-Fi-based occupancy sensing is highly dependent on the location and orientation of the Wi-Fi device, the floor plan, the furniture set up, and the movement of occupants[44]. For example, the occupant needs to be between a pair of Wi-Fi transmitters to achieve high accuracy of occupancy activity detection [45, 46]. Ultra-wideband (UWB) is another type of short-range RF technology for wireless communication. It transmits data between devices by sending short nanosecond pulse over an ultra-wide range of frequencies(3-10GHz). The UWB sensor has also been

leveraged for occupancy counting [33, 47] and motion classification[48]. However, similar to Wi-Fi, the performance for occupancy sensing using a UWB sensor is also sensitive to its position and the indoor setup.

Instead of just applying a single sensing mechanism, data from multiple types of sensors can be fused to improve the performance and reliability of occupancy detection [49-53]. A plug-and-play sensor node is developed for occupancy detection that contains single PIR, temperature humidity, VOC, and CO2 sensors [52]. Furthermore, a novel sensing prototype is proposed to obtain low-level counting consisting of one PIR, one light, one temperature, and three thermal infrared array sensors with different resolutions [50]. Environmental sensing and Wi-Fi data can also be fused to improve the performance of occupancy counting, and the result shows that the accuracy is increased by more than 3% compared to only using environmental sensors [49]. Moreover, the electricity load meter data can be fused with the PIR, Wi-Fi, and environmental sensing data for building-level occupancy counting [53]. Even though sensor fusion can perform better than single sensor technology, their application is restricted by data transmission reliability, computing time, power consumption, and retrofit cost.

1.2. Introduction of Infrared Sensors

1.2.1. The SLEEPIR Sensor for Stationary and Moving Occupancy Detection

Our lab has recently invented several chopped PIR sensors that enable the stationary presence detection of PIR sensors by adding a long-wave infrared (LWIR) chopper: mechanical choppers [54-56] and LC choppers [57-59]. In particular, the synchronized low-energy electronically chopped PIR (SLEEPIR) sensor can significantly

reduce the power consumption, weight, noise level, and volume compared to the mechanical mechanisms[54-56, 60, 61].

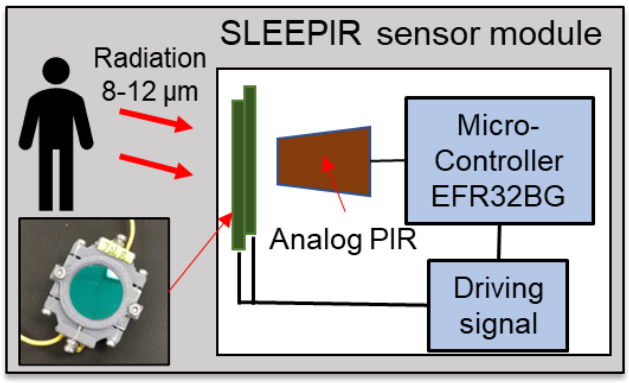


Figure 1.1 The working principle of the SLEEPIR sensor module.

Figure 1.1 shows the systematic working principle of the SLEEPIR sensor module. An LC shutter is installed in front of an analog PIR sensor (Panasonic AMN34211). The microcontroller can generate a two-state (ON-OFF) driving signal to modulate the transmission ratio of the LC shutter under the LWIR region (8-12 μm). The transmission ratio of the LC shutter is increased when the driving signal has the state 'ON'. The actuation period of the driving signal is 8s which contains 4s of 'ON' state and 4s of 'OFF' state. So there are two LC state changes during the actuation period, and output signal peaks are generated. The frequency of the driving signal actuation period is set as one time per 30 s. During the actuation period, the sensor data of SLEEPIR module is sampled at 20Hz and sent to a data-saving hub.

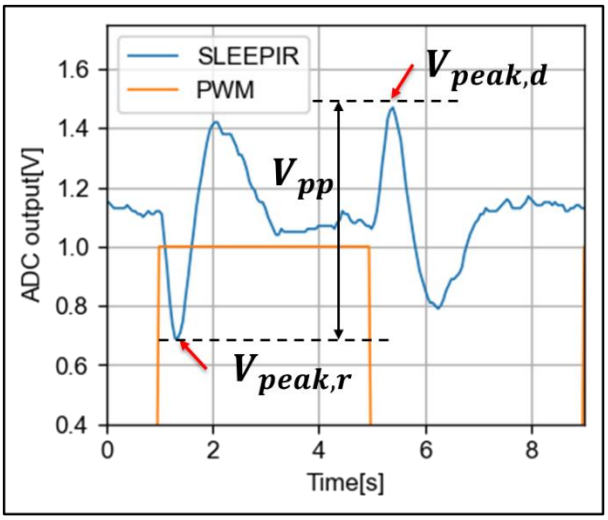


Figure 1.2 The SLEEPIR sensor module output data within one actuation period

Figure 1.2 shows the plotting of SLEEPIR module data within one actuation period. The orange line represents the ON and OFF of the driving signal, and the blue line shows the analog output of the SLEEPIR module. Signal peaks $V_{peak,r}$, $V_{peak,d}$ are generated during the switch time of the driving signal, and the peak-peak voltage V_{pp} can be calculated, which is a critical factor in determining the occupancy status. In addition, all the 8s of data in one actuation period are considered as one sample in the experiments.

During the previous work of the SLEEPIR sensor, a 'modulation' value calculated from the Fourier-transform infrared spectroscopy (FTIR) spectrum is proposed to evaluate the LC shutter quality. Furthermore, the fabrication of the LC shutter is optimized[59]. Preliminary lab experiments have also been performed to validate the ability of stationary occupants detection using the SLEEPIR sensor network and a threshold-based algorithm[58].

1.2.2. The Thermopile Array Sensors

The thermopile array sensor detects the object's surface temperature and produces low-resolution thermal images. It is a passive and privacy-preserving sensing technology. Figure 1.3 shows the working principal of thermopile and a sample 8×8 output data plotting of Grid-EYE sensor with the 'Jet' color scale.

Thermopile has a similar working principle as the thermocouple. The absorbed infrared radiation generates a temperature gradient ΔT, and it can be converted to electrical signal ΔV through the Seebeck effect of thermocouple thermoelectric material. The electrical signal is amplified by connecting thermocouples of micro-level in series.

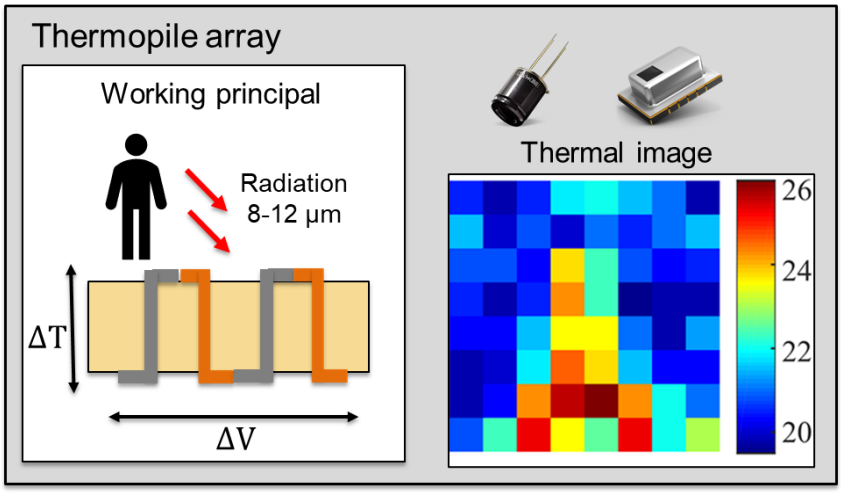


Figure 1.3 Thermopile array sensor working principal

The thermopile array sensor has been researched for occupancy presence detection [62], occupancy counting [28-30, 50] [63], fall detection [64, 65], posture

recognition [27, 66], occupancy localization [67-70], activity recognition [71], household activity monitoring [72], and personal thermal comfort monitoring[73].

Two different models of thermopile array sensor are used in this work: Grid-EYE AMG88(Grid-EYE) produced by Panasonic Corp. and MLX90640(MLX) produced by Melexis. N.V. Table 1.1 shows the specification of the two types of thermopile array sensors.

Table 1.1 Specification of thermopile array sensors

Model	**Grid-EYE**	**MLX90640**
Resolution	8×8	32×24
FOV [˚]	60×60	55×35
Range [˚C]	0 - 80	-40 - 300
Sensitivity [˚C]	0.25	0.1
Accuracy [˚C]	±2.5	±1.0
Frame rate [Hz]	10	8
Interface	I^2C	I^2C
Current [mA]	4.5	23

The Grid-EYE sensor has 8×8 pixels with ±2.5ºC temperature accuracy, a field of view (FOV) of 60º ×60º. It enables three working modes: working mode, sleep mode, and standby mode, with current consumption of 4.5 mA, 0.2 mA, and 0.8 mA, respectively. It outputs an 8×8-pixel frame at 10Hz, representing 64 temperature values in ˚C, with a sensitivity of 0.25 °C. The Grid-EYE sensor also has an onboard thermistor that can provide sensor internal temperature as a reference. The MLX90640 sensor has 32×24 pixels with ±1.0ºC temperature accuracy. A model with a FOV of 55º ×35º is selected. It

outputs 32×24-pixel data with a frame rate of 8Hz and a sensitivity of 0.1 °C. Both Grid-EYE and MLX sensor data can be read through the I^2C interface by microcontrollers.

1.2.3. The Proposed Sensing Platforms

1.2.3.1. The Wireless SLEEPIR Sensor System

Figure 1.4 shows the Bluetooth-enabled SLEEPIR sensor system. It is a wireless sensor network with star topology. Up to seven SLEEPIR sensor nodes can communicate to a Bluetooth central hub for data saving. Each SLEEPIR sensor node contains two SLEEPIR modules and one digital PIR sensor with binary output. Moreover, the SLEEPIR sensor node also has a digital temperature sensor (Model: si7021) on board to record the changing of the ambient temperature. The hub can be plugged into either PC or Raspberry Pi through the USB port for sensor data saving.

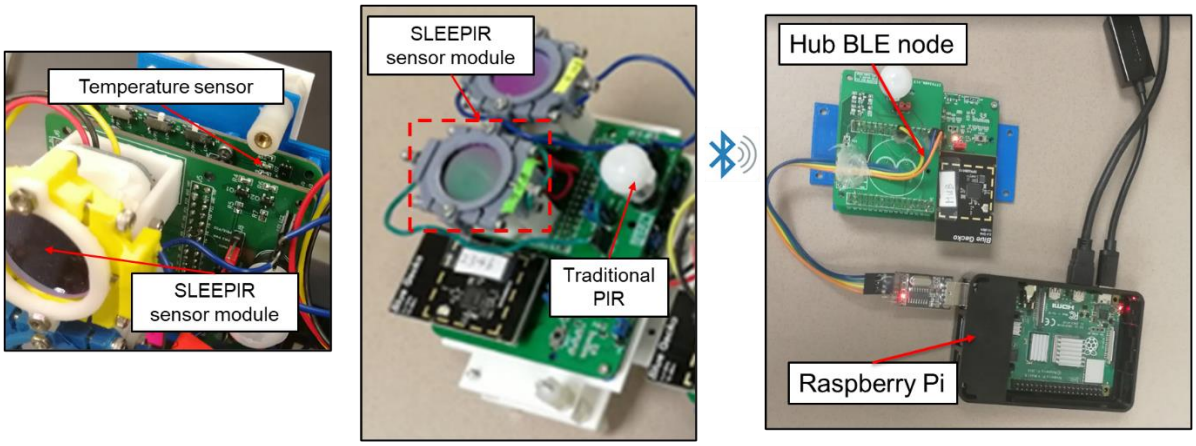


Figure 1.4 The Bluetooth enabled wireless SLEEPIR sensor node and data saving hub

1.2.3.2. The Thermopile Array Sensing Systems

Type A: This sensor node is developed for face direction detection and activity tracking. It integrates Grid-EYE with a time of flight (ToF) VL53L0X distance sensor. The VL53L0X distance detector can measure distance up to 2m with a sensitivity of 1 mm. It uses an active sensing approach but only consumes a current of 4mA. The type I sensor node can send data to the up-level computer through a Bluetooth 4.0 module. The sampling frequency is set as 10Hz by considering the highest sampling frequency of Grid-EYE is 10Hz. Furthermore, both sensors are installed on a rotational platform to track indoor occupants.

Type B: This ceiling-mounted sensor node consists of one Grid-EYE sensor and four VL53L0X distance sensors. It is developed for occupancy identification. The photographic representation of the sensor prototype is shown in Figure 1.5. The four distance sensors are circular distributed around the Grid-EYE sensor with an angle of 25° related to the vertical direction. An Arduino Nano board captures the data from all five sensors through the I^2C interface and sends it to the computer through a Bluetooth module. The sampling frequency of the Grid EYE and the VL53L0X sensor is set as 10Hz and 25Hz, respectively.

Type C: This is a sensor node developed based on an MLX90640 sensor for in-bed posture detection. The sensor node consists of an MLX90640 breakout board (SparkFun Electronics Inc.) and a Teensy 3.6 microcontroller. The sampling frequency is set at 8Hz. Data can be sent to the PC and saved through the serial port.

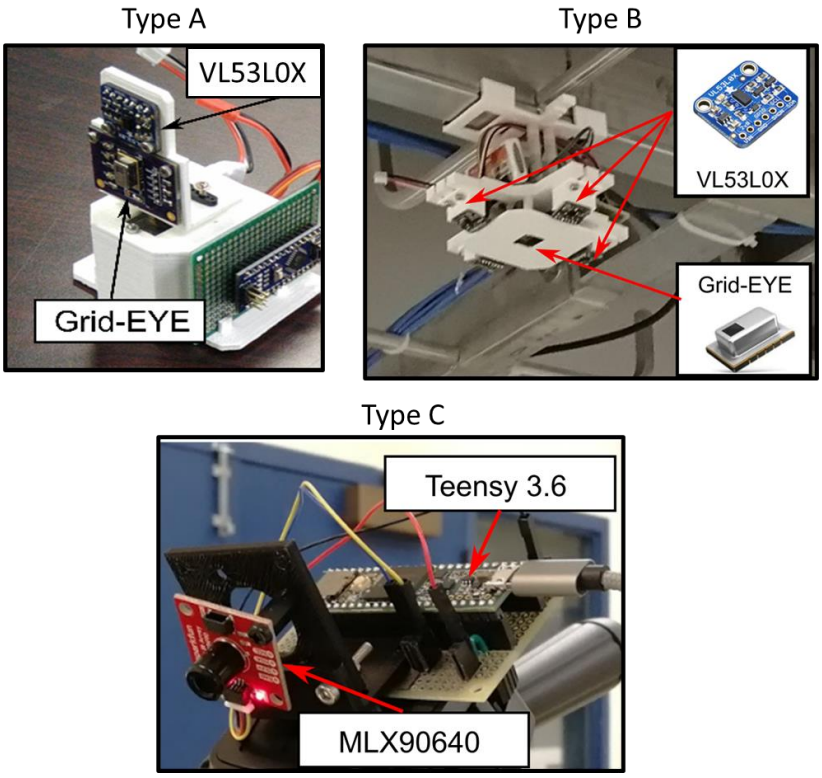


Figure 1.5 The thermopile array sensing systems (Type A, Type B, Type C)

1.3. Data Processing Methods for Occupancy Sensing

A variety of Data processing methods have been used for occupancy sensing [74, 75]. Occupancy presence and counting can be considered as classification problems. Thus, traditional machine learning classifiers have been widely used, such as support vector machine (SVM), Naive Bayesian (NB), K Nearest Number (KNN), Random Forest (RF), Decision Tree (DT), and Full connected feed-forward network (NN). DT, KNN, NB, SVM, and NN models are compared for presence detection and counting in the office using environmental sensors, and the DT model achieves the highest accuracy in both presence detection and counting tasks[76]. NN and SVM models are compared for occupancy

counting in a theater using CO2 sensors[77], and both models reach close accuracy of around 76%. The specific feature extraction method is required for different occupancy sensing objectives or using different sensor devices. And the performance of traditional machine learning methods highly depends on the features to be used.

The Hidden Markov Model (HMM) takes the occupancy state as a Markov chain with hidden states, and a future state can be predicted based on the current state. It has been proposed to be an effective method for occupancy presence detection and counting based on different sensing devices such as PIR sensors [14], smart meters [78], or environmental sensors [79]. However, a long-time uncontrolled dataset is required to get a reliable HMM model.

The computer vision domain contains lots of models for data processing data using cameras [80, 81], which has also been applied for occupancy sensing. For example, object detection algorithms can also work for occupant counting by detecting the number of human subjects in the FOV[82]. HOG is a traditional method for object detection [83]; it is used with an NN classifier for presence detection using combined data from optical and thermal cameras [22]. Moreover, the Haar-like feature, classical face detection features, is used for head detection and determining the walking direction of occupants [84].

Latest computer vision algorithms rely on deep learning models for object detection. For example, YOLO is a lightweight model for fast object detection, which could be directly used for occupancy counting.[85]. Instead of using existing models, a customized CNN model is also proposed[21]. On the other hand, deep learning models generally need a large dataset, while it may not be available for every objective. Then pre-

trained models such as AlexNet[86], ResNet[87], R-CNN[88] can be used as feature extractors and fine-tuned for the specific problem.

Deep learning techniques can also be used with other sensing devices other than optical cameras[89]. For example, a CDBLSTM network is proposed for occupancy counting using environmental sensors[24]. Furthermore, A CNN model is proposed for occupancy metabolic rate estimation using depth camera data [23]. Finally, an Autoencoder Long-term Recurrent Convolutional Network (AE-LRCN) model is proposed for occupancy activity detection using Wi-Fi CSI data[90].

1.4. Data Processing Methods using Infrared Sensors

Since the PIR sensor cannot sense stationary occupants and may frequently cause false detection, a common approach is introducing a prediction algorithm. The hidden Markov model(HMM), a probability-based method, can reduce false occupancy detection [14]. Furthermore, a handcrafted feature-based machine learning model is trained based on historical motion sensor data to predict occupancy patterns and improve occupancy detection accuracy[17]. In addition, different machine learning classifiers and sequential models are compared using motion data collected by residential thermostats[15]. These prediction models largely improve the accuracy of occupancy detection. However, the model only works when a relatively fixed occupancy pattern can be identified, which may require a large training dataset. In contrast to the previous work in which the PIR sensor is used as a digital sensor with binary output, features can also be extracted from the analog output of the PIR sensor. For example, a linear regression model has been proposed using data from a single analog PIR sensor for occupancy detection in a meeting room[18]. In

summary, the aforementioned research improves PIR sensing performance by increasing the sensor number and applying advanced detection algorithms. However, the limitation of the PIR sensor in stationary occupancy detection remains.

On the other hand, a few methods have been applied to identify stationary occupants using a chopped PIR sensor. For example, the fixed thresholding method is applied to the peak-to peak output of the mechanical chopped PIR sensor V_{pp} to classify the ‘occupied’ and ‘unoccupied’ conditions and to determine the FOV for stationary detection [54, 55]. Furthermore, handcrafted feature-based machine learning has been applied to a 10-hour presence detection with 99.12% using the SLEEPIR sensor[56, 58].

Machine learning techniques are popular data processing methods among thermopile array sensors. Handcrafted features and machine learning classification methods such as SVM, RF, and NN have been applied and compared to a single Grid-EYE sensor for counting[63] walking direction detection[91] and activity detection[65, 92]. Deep learning models such as CNN have been applied for yoga posture detection using three Grid-EYE sensors[66]. In addition, a few background removal and denoising methods have been proposed for presence detection and traffic monitoring using the MLX sensor[93, 94].

1.5. The Objective of This Dissertation

Two types of infrared sensing technologies, the SLEEPIR sensor, and the thermopile array sensor, have been applied for fine-grained occupancy sensing objectives such as true presence detection, posture detection, facing direction detection, and

occupancy identification. In addition, data processing models based on handcrafted features, traditional machine learning, and deep learning are proposed.

In Chapter 2, a mathematical model is proposed to predict SLEEPIR output for stationary occupancy detection under changing ambient temperature. The unoccupied $V_{pp,u}$ is found to be proportional to the temperature difference of sensor and background $V_{pp,u} = K(T_{floor}^4 - T_{sensor}^4)$. The linearity is validated, and the coefficient K_{BB} is characterized using a blackbody radiation source. The LC shutters' performance for stationary occupancy detection is evaluated by performing the occupant experiments to obtain ΔV_{pp} at different distances. The sensor noise is analyzed, and the LC shutters can be classified as 'Good','Fair', and 'Poor' based on metric ΔV_{pp}@1.5m. Moreover, K_{BB} is highly correlated with the ΔVpp@1.5m value, so it can help reduce the time of LC shutter classification vastly. Finally, SLEEPIR sensor nodes are prepared with 'Good' LC shutters to enable the maximum FOV of stationary occupancy detection.

In Chapter 3, datasets for occupancy presence detection using SLEEPIR sensor nodes are collected in a lab room and a room in a residential apartment. In addition, two edge cases experiments are also performed, including changing the room temperature and lying on a bed. Finally, different occupancy presence detection algorithms are proposed and compared. StateS, Stat. ML, LSTM, and Stat.LSTM algorithms are proposed. The different occupancy presence detection algorithms are compared by reporting their daily occupancy detection accuracy. The result shows that both LSTM and Stat. LSTM reaches reliable high accuracy >96.0% for both the lab room and residential apartment. Moreover, the Stat. LSTM model does not require the data from a digital PIR sensor. Furthermore,

the result of the edge case experiment indicates that the proposed model is a reliable occupancy presence detection method.

In Chapter 4, feature extraction methods are proposed for fall detection and in-bed posture detection. The fall detection relies on an 8×8 pixel Grid-EYE sensor and a distance sensor. Experimental studies include three non-fall activities – standing, sitting, and stooping, and two fall actions – forward falling and sideway falling to simulate elderly daily activities. Different handcrafted feature sets for the SVM-based machine learning algorithm are analyzed, and their impact on fall detection accuracy is evaluated and compared empirically with an overall accuracy above 90% achieved. The preprocessing method and feature extraction approach based on the HOG (HOG+PCA) is compared to the handcrafted feature for the in-bed posture detection. The user study shows that a total of 9 in-bed postures can be successfully classified with 5-fold cross-validation-accuracy over 99.8%. The cross-user-validation is also performed to evaluate the robustness of posture classification models. The HOG features show significantly higher accuracy than handcrafted features.

In Chapter 5, a pre-trained CNN model is proposed as a feature extractor for facing direction detection. A customized CNN is proposed to fuse the data from two types of sensors for occupancy identification. The CNN-based feature extraction demonstrates more reliable performance for facing direction detection compared to the handcrafted features regardless of the detection ranges. A non-intrusive sensor node is prototyped based on a thermopile array and four distance sensors for occupancy identification. The sensor node is installed near the lab entrance as the realistic setup to get the dataset, and

94.4% accuracy is achieved to identify eight different people. The proposed approach also shows a higher accuracy and lower power consumption compared to the state-of-the-art.

2. SLEEPIR SENSOR MODELING AND CHARACTERIZATION

2.1. Mathematic Model for Stationary Occupancy Detection

A mathematical model for stationary occupant detection is developed to describe the SLEEPIR module output under different ambient temperatures. The thermal power absorbed by the PIR sensor from an object can be calculated as $A\varepsilon\varepsilon_{sensor}\sigma(T^4 - T_{sensor}^4)$ [95], where A is the geometry factor. ε, ε_{sensor} are the emissivity of object and sensing element, respectively. σ is the Stefan-Boltzmann constant. T_{sensor}, T are the surface temperature of the sensor and object. Then the energy received by the SLEEPIR module W can be written as equation **(1)**

$$W = A_{floor}\varepsilon_{floor}\varepsilon_{sensor}\sigma(T_{floor}^4 - T_{sensor}^4) + A_{human}\varepsilon_{human}\varepsilon_{sensor}\sigma(T_{human}^4 - T_{sensor}^4) \quad (1)$$

Here A_{floor} and A_{human} are the geometry factor of the floor surface and the human body; $\varepsilon_{sensor}, \varepsilon_{floor}, \varepsilon_{human}$ are the emissivity of the sensor, floor surface, and the human body, respectively; $T_{sensor}, T_{floor}, T_{human}$ are the surface temperature of SLEEPIR sensing element, floor surface, and the human body, respectively. Thus, the energy received by the SLEEPIR model consists of two parts: the energy received by the human body and the background, e.g., the floor of the room.

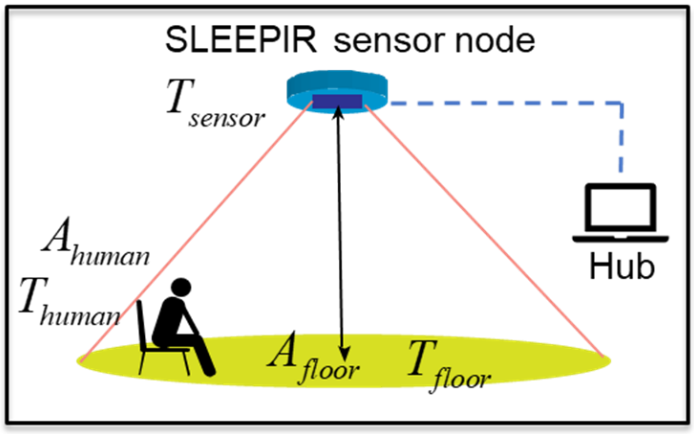


Figure 2.1 systematic of SLEEPIR sensor for stationary occupancy detection

The following equation (2) can express the output peak voltage generated during LC shutter-controlled transmission ratio change.[58]

$$V_{out}(t) = \frac{R_f \eta p' A_p \omega}{2G_T (1+\omega^2 \tau_T^2)^{1/2} (1+\omega^2 \tau_E^2)^{1/2}} W(t) \quad (2)$$

Here p is the perpendicular component of the pyroelectric coefficient; A_p is the area of the pyroelectric sensing element; η represents the emissivity of the sensing element; τ_T and τ_E represents the thermal and electrical constant, respectively. ω is the modulated frequency that can be ω_d or ω_r, stands for the two different LC transitions ('ON-OFF' and 'OFF-ON'). Moreover W_{ON} , W_{OFF} are the energy received by the SLEEPIR module when the LC shutter state is 'ON' and 'OFF'. And they can be rewritten as $\xi_{LC\_on} W$ and $\xi_{LC\_off} W$, where ξ_{LC_on} , ξ_{LC_off} represent the transmission ratio of LC shutter under state 'ON' and 'OFF'. Here, the transmission ratios are considered as

constants assume room object has a small range of temperature change; also, it can avoid the complicated integral on wavelength.

$$V_{peak,r} = \frac{R_f \eta p' A_p \omega_r (W_{OFF} - W_{ON})}{2G_T (1+\omega_r{}^2 \tau_T^2)^{1/2} (1+\omega_r{}^2 \tau_E^2)^{1/2}}$$
$$V_{peak,d} = \frac{R_f \eta p' A_p \omega_d (W_{ON} - W_{OFF})}{2G_T (1+\omega_d{}^2 \tau_T^2)^{1/2} (1+\omega_d{}^2 \tau_E^2)^{1/2}} \quad (3)$$

Then we can get the peak-peak voltage during a full-LC shutter actuation by calculating $V_{pp} = V_{peak,d} - V_{peak,r}$. The peak-peak voltage (marked in Figure 1.2) is a key indication extracted from the sensor output signal, representing the output of the SLEEPIR module for occupancy detection. The peak-peak voltage for 'occupied' and 'unoccupied' situations can be written as

$$V_{pp,u} = \sigma K_{PIR} K_{LC} K_{Pos,u} (T_{floor}^4 - T_{sensor}^4)$$
$$V_{pp,o} = \sigma K_{PIR} K_{LC} [K_{Pos} (T_{floor}^4 - T_{sensor}^4) + K_{human} (T_{human}^4 - T_{sensor}^4)] \quad (4)$$

It contains four coefficients: K_{PIR} K_{LC} K_{Pos} K_{human} and the detailed expressions are shown in equation (5). Here, K_{PIR} only depends on the properties of PIR sensor. Since we use an off-the-shell PIR sensor from Panasonic Inc., it can be considered as a constant. Then K_{LC} only depends on the fabrication of LC shutter. K_{Pos} only depends on the setup position of the SLEEPIR sensor. K_{human} depends on the body shape, posture, and position of the human in the FOV.

$$\begin{cases} K_{PIR} = \dfrac{R_f \eta p' A_p}{2G_T} \varepsilon_{sensor} \\ K_{LC} = \left\{ \left[\dfrac{\omega_r}{(1+\omega_r^2\tau_T^2)^{1/2}(1+\omega_r^2\tau_E^2)^{1/2}} + \dfrac{\omega_d}{(1+\omega_d^2\tau_T^2)^{1/2}(1+\omega_d{}^2\tau_E^2)^{1/2}} \right] \right\} (\xi_{LC\_on} - \xi_{LC\_off}) \\ K_{Pos} = A_{floor}\varepsilon_{floor} \\ K_{human} = A_{human}\varepsilon_{human} \end{cases} \quad (5)$$

The energy received by the human body is hard to measure during experiments since T_{human} and A_{human} can be highly sensitive to body shape, cloth, and posture. If we consider the unoccupied situation, the equation can be rewritten as an equation $V_{pp,u} = K(T_{floor}^4 - T_{sensor}^4)$, where $K = \sigma K_{PIR} K_{LC} K_{Pos,u}$ is a constant for specific LC shutter and fixed sensor node setup. Moreover, the peak-peak voltage of SLEEPIR module output signal is linearly proportional to the difference between the fourth power of the floor temperature and sensor temperature.

We determine $\Delta V_{pp} = |V_{pp,o} - V_{pp,u}|$ as a metrics for stationary occupancy detection, and it can be represented as the following equation

$$\Delta V_{pp} = \sigma K_{PIR} K_{LC} [(K_{Pos,o} - K_{Pos,u})(T_{floor}^4 - T_{sensor}^4) + K_{human}(T_{human}^4 - T_{sensor}^4)] \quad (6)$$

We can find that a high K_{LC} value is desired for SLEEPIR sensor to work as a presence detector as it will generate larger $\Delta V_{pp.}$

2.2. Experimental Characterization of Sensing Model

A blackbody radiation source (Model: Process Sensors BBS500) is placed 1cm close to the SLEEPIR module to cover its whole field of view (FOV). (Figure 2.2 (a)).

During this setup, we can consider the T_{floor} to be the temperature of the blackbody T_{bb}. Then $\Delta T = T_{BB}^4 - T_{sensor}^4$ can be calculated. Figure 2.2 (b) shows the plotting of sensor temperature T_{sensor} and the V_{pp} recording of the SLEEPIR module. The blackbody is set to different temperatures, and for each LC shutter and each temperature, 15 minutes of data is collected. For every LC shutter, the data collected is used to calculate the blackbody coefficient K_{BB} by fitting the data to a linear equation. Here, $K_{BB} = K_{PIR} K_{LC} K_{Pos,u} \sigma$. Based on equation (4)

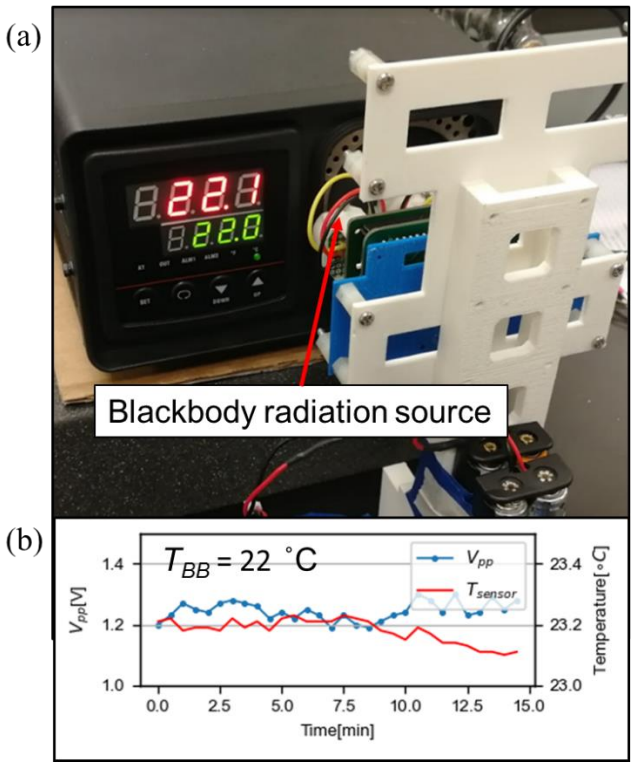


Figure 2.2 (a) Blackbody experiment (b) Sample output of SLEEPIR sensor V_{pp} T_{sensor}

Figure 2.3 shows the plotting of temperature difference ΔT and output V_{pp} of eight LC shutters. Here the LC shutters are named by their major fabrication parameters(M: mass ratio, D: cell gap, C: cooling rate). We find that the R^2 values are generally higher than 0.95, which validates the linear relationship with our proposed sensing model $V_{pp,u} = K_{BB}(T_{BB}^4 - T_{sensor}^4)$. Furthermore, different LC shutters can have very different K_{BB} values. For example, LC shutter M80D22C100_1 has a high K_{BB} value 0.450, and the K_{BB} value of LC shutter M80D22C100_6 is only 0.268. This K_{BB} value calculated from the blackbody experiment can help determine the quality of LC shutters. This is illustrated in section 'LC shutter classification'.

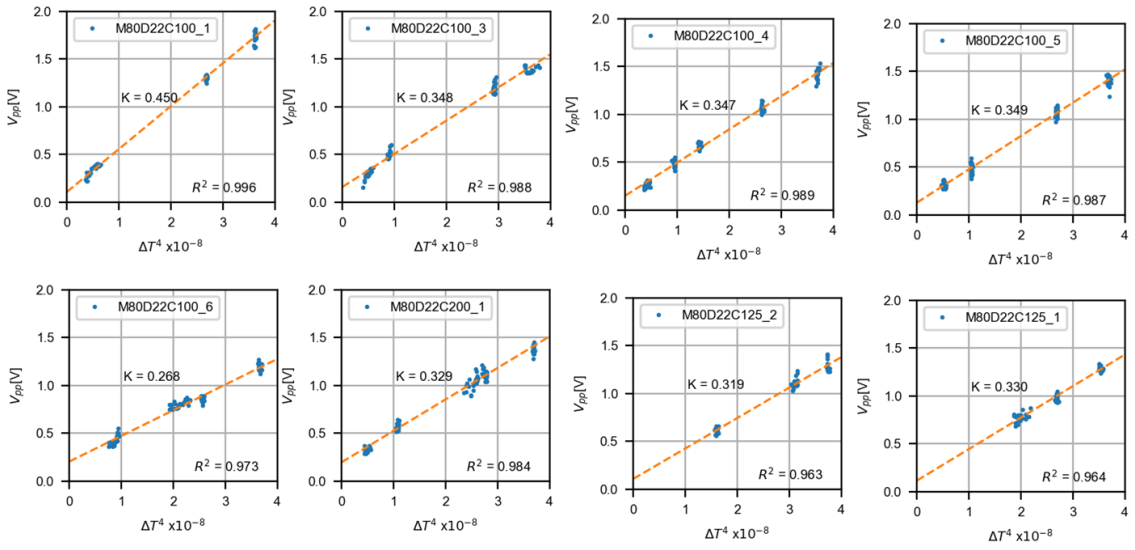


Figure 2.3 Experimental characteristic of the mathematical model

2.3. Classify LC Shutters

2.3.1. Sensor Noise

The noise signal of the sensor is a crucial factor that affects the sensor performance. The output of SLEEPIR will not be precisely the same even under the same sensor and background temperature. For example, Figure 2.2(b) shows that the V_{pp} value still oscillates between about 1.1V to 1.2V even when the sensor temperature remains around 23.2°C. This noise can be caused by hardware components such as the LC driving circuit and analog-digital conversion (ADC) circuits, so a better understanding of the noise level is desired. The data collected from the blackbody experiments are used to evaluate the sensor noise level. Two metrics are designed, (1) $V_{pp,u.std}$ (2) $\Delta V_{pp,i,u.}$

During the blackbody experiments, data is recorded for 15 minutes at every temperature per LC shutter. $V_{pp,u.std}$ is defined as the standard deviation of the V_{pp} within the 15 minutes time. Figure 2.4 shows the plotting of $V_{pp,u.std}$ vs K_{BB}, and we can find no apparent difference between the sensor noise $V_{pp,u.std}$ of different LC shutters. Moreover, the $V_{pp,u.std}$ value is generally smaller than 0.05V.

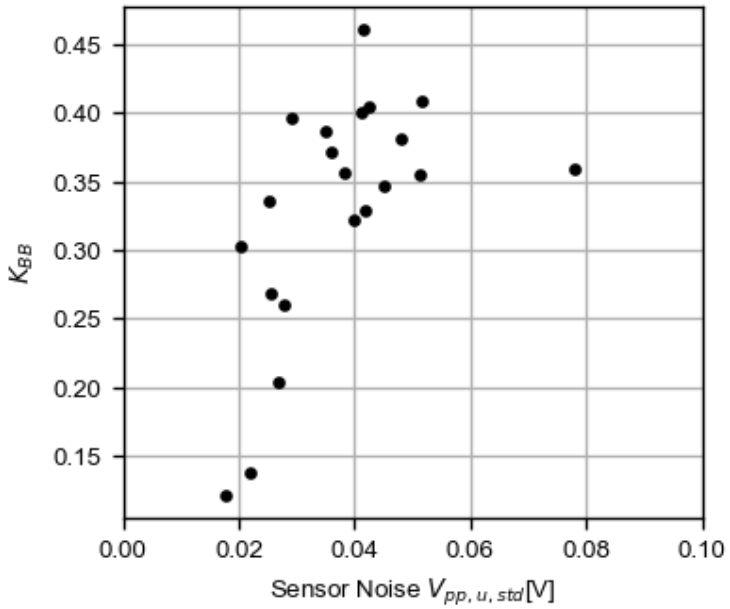


Figure 2.4 SLEEPIR Sensor noise: different LC shutter under same T_{sensor}, T_{BB}

$\Delta V_{pp,i,u.}$ is defined as $V_{pp}(i+1)-V_{pp}(i)$ the difference between two continuously measured peak-peak values. Therefore, for every LC shutter, a series of $\Delta V_{pp,i,u.}$ can be calculated per temperature. Then for every temperature, these values can be combined to represent the distribution of sensor noise. To be more specific, Figure 2.5 shows the cumulative distribution function (CDF) plotting of sensor signal noise level under different blackbody temperatures (23℃, 24℃, 25℃, 26℃). It indicates that more than 90% of the signal noise data is smaller than 0.05V and more than 99% of the signal noise data is smaller than 0.1V. So to avoid false detection caused by the sensor noise, the ΔV_{pp} value of the occupant experiment needs to be larger than 0.1V.

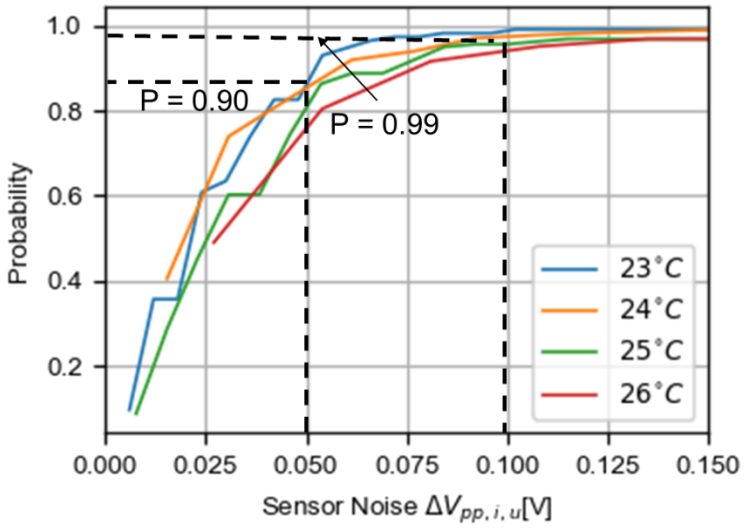


Figure 2.5 SLEEPIR Sensor noise: different T_{BB}

2.3.2. Determine Detection Range of SLEEPIR Sensor ΔV_{pp}

The lab occupant experiment is used to validate and evaluate the performance of the LC shutter as not every LC shutter manufactured is sufficiently good for a presence detector. It is performed in a lab with a room temperature of around 22˚C. During the experiments, the occupant sits stationary on a chair facing the sensor node Figure 2.6(a)). Therefore, V_{pp} during occupied and unoccupied situations $V_{pp,u}, V_{pp,o}$ can be achieved through the occupant experiment. Moreover, for any position, we can calculate ΔV_{pp}, the absolute difference between $V_{pp,u}, V_{pp,o}$ as the metrics to determine its stationary occupancy detection performance.

On the other hand, correcting ΔV_{pp} values at all distances is time-consuming, so a single value is desired to represent the stationary occupancy detection performance. Here we select the ΔV_{pp} value at 1.5m. The single distance of 1.5m is selected because it is the

desired detection distance for the sensor to be used in a 4m × 4m room. Furthermore, the desired ΔV_{pp} value should be larger than the sensor noise of SLEEPIR module. Then the LC shutters can be classified into three categories based on their ΔV_{pp}@1.5m: Good: >0.1V, Fair: [0.05V-0.1V]. Poor: <0.05V. Figure 2.6(b) shows the measured V_{pp} using Good, Fair, and Poor LC shutters for 1.2m,1.5m,1.8m. We can see that the ΔV_{pp} value decreases when the distance increases. Also, for good 'LC's ΔV_{pp} @1.5m is bigger than that of 'Poor' LCs and generally larger than 0.1V.

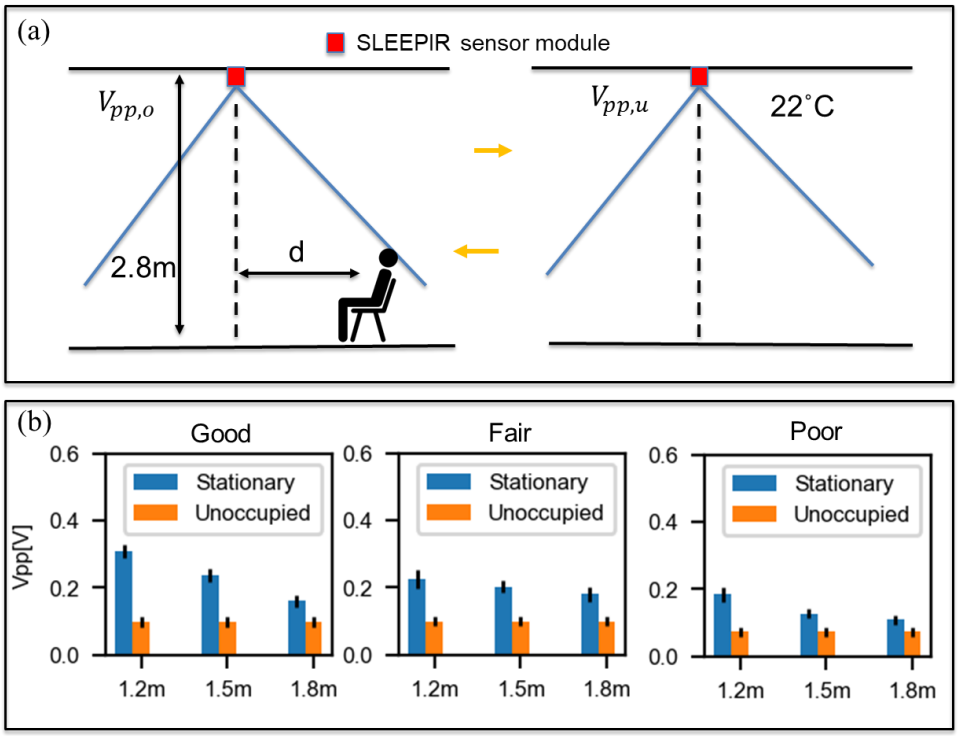


Figure 2.6 (a) Occupant experiment (b) Classify LC shutter using ΔV_{pp} @1.5m

2.3.3. LC Shutter Classification

Previously, the 'Modulation' calculated from FTIR spectrum and occupant experiments is used to identify good LC shutters[56]; however, the occupant experiments are time-consuming and require lots of labor. Also, we find that high modulation does not always lead to a high ΔV_{pp} value. In addition, 'Modulation' is not consistent even with the same fabrication parameter. For example, Table 2.1 shows 23 LC shutters included in this analysis. Most of them have relatively high modulation values (5-6), and 19 are fabricated using the same method(M80D22C100). However, the modulation is not relatively consistent (Figure 2.7).

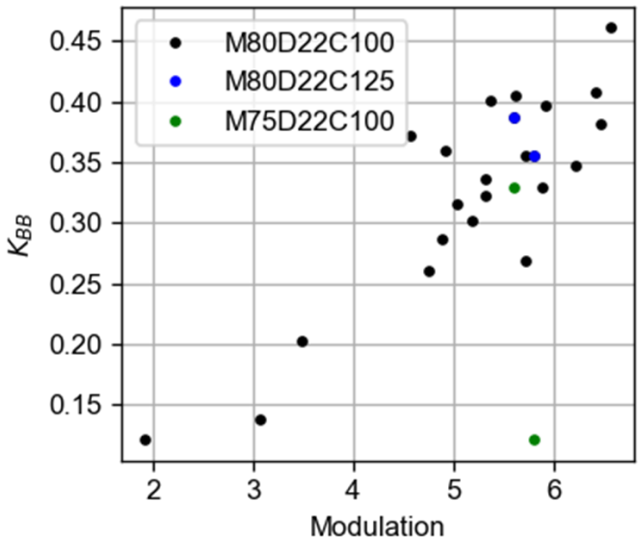


Figure 2.7 Modulation and K_{BB}

In summary, modulation may not be the best indicator to compare different LCs. Therefore, a more reliable and efficient characterization approach is desired. As the

position of SLEEPIR sensor is fixed during the blackbody experiment, the blackbody K coefficient K_{BB} only depends on the fabrication of the LC shutter based on our proposed sensing model (equation (4)). Here we propose to use K_{BB} as a better standard for classifying LC shutters.

Table 2.1 Specification of LC Shutters

LC ID	LC name	ΔV_{pp} @1.5m[V]	Modulation	R^2	K_{BB}
#1	M80D22C100_1	0.126	6.58	0.996	0.461
#2	M80D22C100_2	0.082	6.48	0.921	0.381
#3	M80D22C100_3	0.101	6.42	0.988	0.408
#4	M80D22C100_4	0.066	5.73	0.989	0.356
#5	M80D22C100_5	0.062	6.22	0.987	0.347
#6	M80D22C100_6	0.044	5.73	0.973	0.268
#7	M80D22C100_7	0.158	5.62	0.957	0.405
#8	M80D22C100_8	0.078	5.33	0.983	0.322
#9	M80D22C100_9	0.146	4.57	0.985	0.372
#10	M80D22C100_10	0.092	4.92	0.945	0.359
#11	M80D22C100_11	0.054	4.76	0.985	0.261
#12	M80D22C100_12	0.056	4.88	0.993	0.286
#13	M80D22C100_13	0.031	3.07	0.938	0.138
#14	M80D22C100_14	0.121	5.37	0.989	0.401
#15	M80D22C100_15	0.068	5.04	0.975	0.345
#16	M80D22C100_16	0.074	5.32	0.988	0.336
#17	M80D22C100_17	0.071	5.18	0.956	0.302
#18	M80D22C100_18	0.066	3.49	0.951	0.203
#19	M80D22C100_19	0.112	5.93	0.988	0.396
#20	M75D22C100_1	0.062	5.89	0.984	0.329
#21	M80D22C125_1	0.116	5.61	0.964	0.387
#22	M80D22C125_2	0.072	5.80	0.963	0.355
#23	M75D22C100_2	0.014	1.92	0.932	0.121

The correlation analysis can be used to compare the performance of K_{BB} and modulation. Figure 2.8(a) shows the plotting between the ΔV_{pp} @1.5m and K_{BB}, Figure 2.8(b) shows the plotting between the ΔV_{pp}@1.5m and modulation. The Pearson correlation coefficient is calculated, which shows that the K_{BB} coefficient is a better metric to determine the quality of an LC shutter as it has a better linear correlation ΔV_{pp}@1.5m. This black body method is a more reliable and less time-consuming method to characterize LC shutters. Moreover, we can set a standard here that $K_{BB} > 0.35$ for a 'Good' LC shutter to reduce the time of occupant experiment significantly

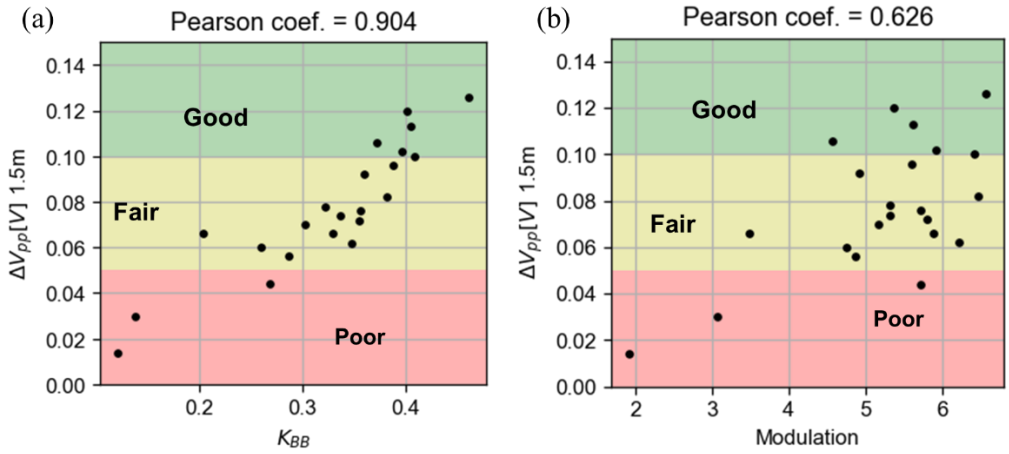


Figure 2.8 (a) ΔV_{pp}@1.5m vs K_{BB} (b) ΔV_{pp}@1.5m vs Modulation

2.3.4. SLEEPIR Sensor Field of View

As the occupant experiment only measures the sensor performance in one direction, we can measure the FOV of a SLEEPIR module by measuring the distance with $\Delta V_{pp}@1.5m > 0.1$V at different directions. During the FOV experiment, the occupant will sit in eight different directions. Figure 2.9(a) shows the FOV testing result of a SLEEPIR

module, and we can find that the FOV of a single SLEEPIR module is an eclipse, so a 20°C rotational degree is applied to the two SLEEPIR modules to enlarge the FOV of the SLEEPIR node during occupancy presence detection experiments (Figure 2.9 (b)).

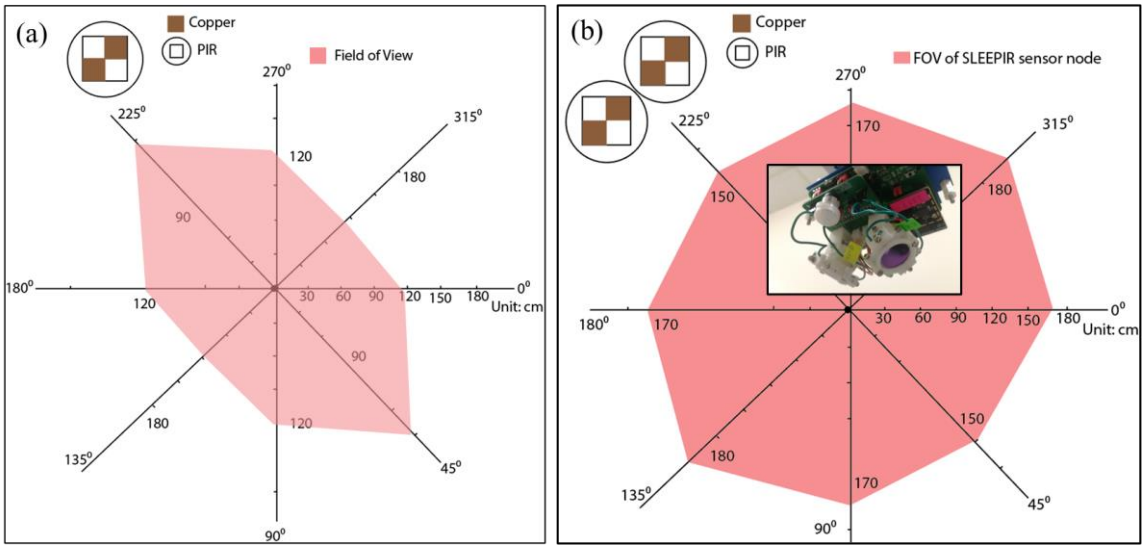


Figure 2.9(a) FOV of a SLEEPIR module with 'Good' LC shutter (b) FOV of a SLEEPIR node with two 20˚ rotated SLEEPIR modules

Based on this LC classification method and SLEEPIR sensor node design, three SLEEPIR sensor nodes are prototyped with 'Good' LC shutters, as shown in Table 2.2

Table 2.2 Selected SLEEPIR sensor node for occupancy presence detection

Sensor node	LC shutter
I	LC_0: #1 (Good); LC_1: #3 (Good)
II	LC_0: #7 (Good); LC_1: #9 (Good)
III	LC_0: #14 (Good); LC_1:#21 (Good)

2.4. Conclusion

This chapter proposes a mathematical model to predict SLEEPIR sensor output for stationary occupancy detection under changing ambient temperature. The unoccupied $V_{pp,u}$ is found to be proportional to the temperature difference of sensor and background $V_{pp,u} = K(T_{floor}^4 - T_{sensor}^4)$. The linearity is validated, and the coefficient K_{BB} is characterized using a blackbody radiation source. The LC shutter performance for stationary occupancy detection is evaluated by performing the occupant experiments to obtain ΔV_{pp} at different distances. The sensor noise is analyzed, and the LC shutters can be classified as 'Good','Fair', and 'Poor' based on metric ΔV_{pp}@1.5m. Moreover, K_{BB} is highly correlated with the ΔVpp@1.5m value, so it can help reduce the time of LC shutter classification vastly. Finally, SLEEPIR sensor nodes are prepared with 'Good' LC shutters to enable the maximum FOV of stationary occupancy detection.

3. OCCUPANCY PRESENCE DETECTION USING THE SLEEPIR SENSOR

3.1. Review of Data Processing Models for Occupancy Sensing using PIR sensor(s)

Occupancy presence detection reaches 97.3% within a 2.5h period of a single office by applying HMM methods on data of a digital PIR[14]. Furthermore, based on the digital PIR sensor data from the residential thermostats, 81-98% accuracy is reached for presence detection based on handcraft features + RF and RNN model[15]. Moreover, the number of occupants in a meeting room can be estimated with <1 RMSE using an analog PIR sensor and statistical regression model[18].

Multiple sensors are usually integrated into the same sensor node as a single PIR sensor contains limited information. The PIR sensor can work with CO_2 and temperature sensors for presence detection in offices, and a mean absolute error(MAE) of 2% is achieved using a rule-based probability estimation algorithm[52]. A sensor node with four analog PIRs is installed on the ceiling to detect eight walking directions. Then more than 98% accuracy is reached based on peak detection and statistical machine learning methods[96]. Furthermore, a CNN model helps to increase the classification accuracy to 99.5% [20]

Due to the single output data dimension of the PIR sensor, a sensor network is generally required for fine-grained occupancy objectives. Four digital PIR sensors are installed in a meeting room, and up to 98.3% is achieved for occupancy counting using the particle filter algorithm[97]. A digital PIR sensor and an electrical meter are installed on each desk in an office building, and then 78-87% occupancy counting accuracy is reached based on HMM and finite-state machine methods. Office building level

occupancy counting can reach 83% accuracy by applying the NN model to fused data of Wi-Fi-connected devices and >100 digital PIR sensors. Thirty-one digital PIR and four door sensors(magnetic sensors detecting the opening and closing of the doors) are installed in an apartment with only one resident for eight months. Then based on customized CNN models, four travel patterns and nine activities can be classified with 97.84% and 99.23% accuracy, respectively[98] [40].

3.2. Experiments Setup and Dataset Collection

The occupancy presence detection experiments are performed in two rooms: a lab room and a room in a residential apartment, as shown in Figure 3.1. Both experiment sites contain a table and a chair, and the main activity of the occupants is working when they stay in the room during the experiment period. The lab room has a surveillance camera to record ground truth. For the residential room, the ground truth is manually recorded by the occupant. Five different experiment setups are included, as shown in Table 3.1. Dataset LAB1, LAB2, and APT1 are uncontrolled daily occupancy detection experiments collected from the lab room (LAB1: subjects with T-shirts and LAB2: subjects with coats) and the apartment room (APT1). In addition, the 24 h data is used for algorithm evaluation for the lab room dataset, but only data from 10:00 to 0:00 is used for the residential room. So, there are 2,880 samples per day in the lab room and 1,680 samples per day in the residential apartment room. Moreover, two edge cases are also performed in the lab room. For EDGE1 data, the air around the sensor node is actively heated by a heater to simulate the fast-changing room temperature. For EDGE2, the occupant lies on a bed in the lab room.

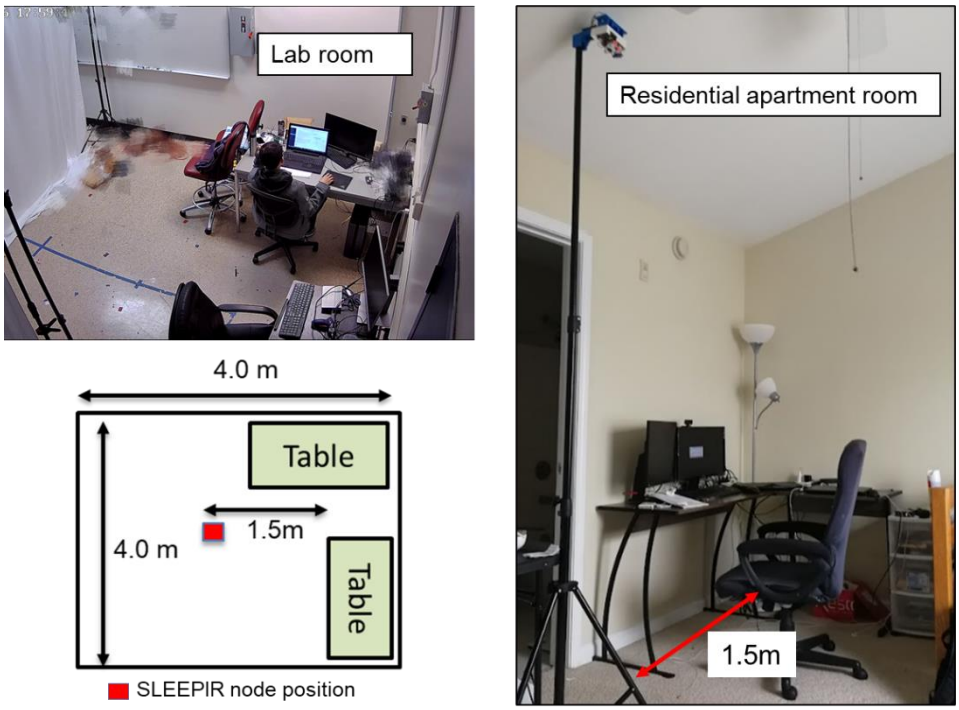


Figure 3.1 Optical photo of the experiment sites: lab room, residential room

Table 3.1 Summary of the Dataset

Name	Site/node#	No. subjects	Setup	Size
LAB1	Lab room/I	2	T-shirt	10 days
LAB2	Lab room/I	3	Coat	10 days
EDGE1	Lab room/I	1	Heating	2 days
EDGE2	Lab room/I	2	Ly down	3 days(90 min)
APT1	Residential room/II	1	T-shirt	5 days

The samples with the datasets can be separated into three categories: 'Unoccupied', 'Motion' and 'Stationary'. The 'Unoccupied' samples just mean the occupant is not in the room. The 'Motion' sample means the digital PIR detects the occupant on the SLEEPIR node, and the 'Stationary' sample means the occupant is in the room but cannot be detected by the PIR sensor. Figure 3.2 shows the distribution of

samples within a day for the dataset collected in the lab and residential homes. We can find out that there are generally more ‘Stationary’ samples than ‘Motion’ Samples.

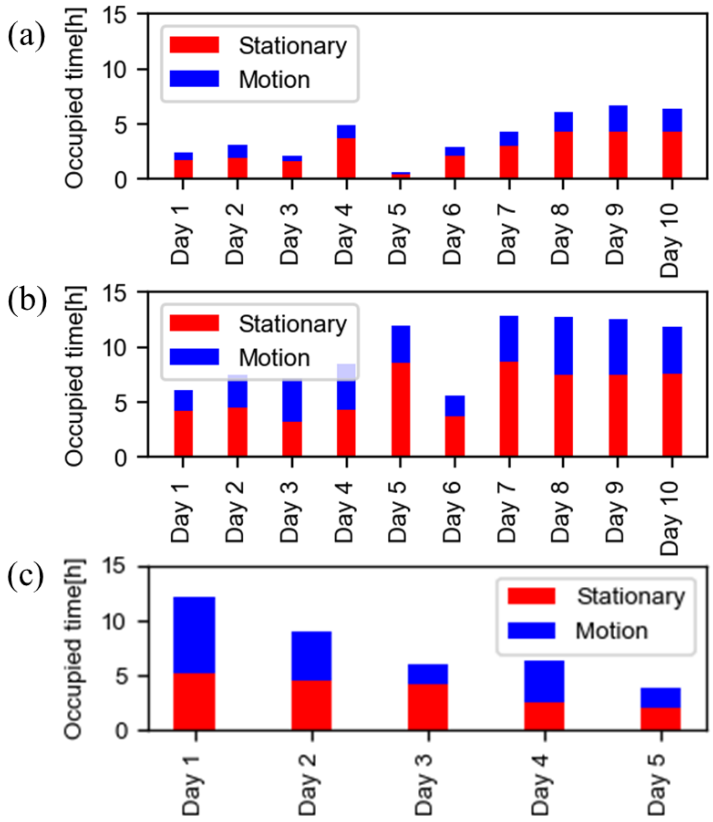


Figure 3.2 Distribution of different samples: (a) LAB1 (b) LAB2 (c) APT1

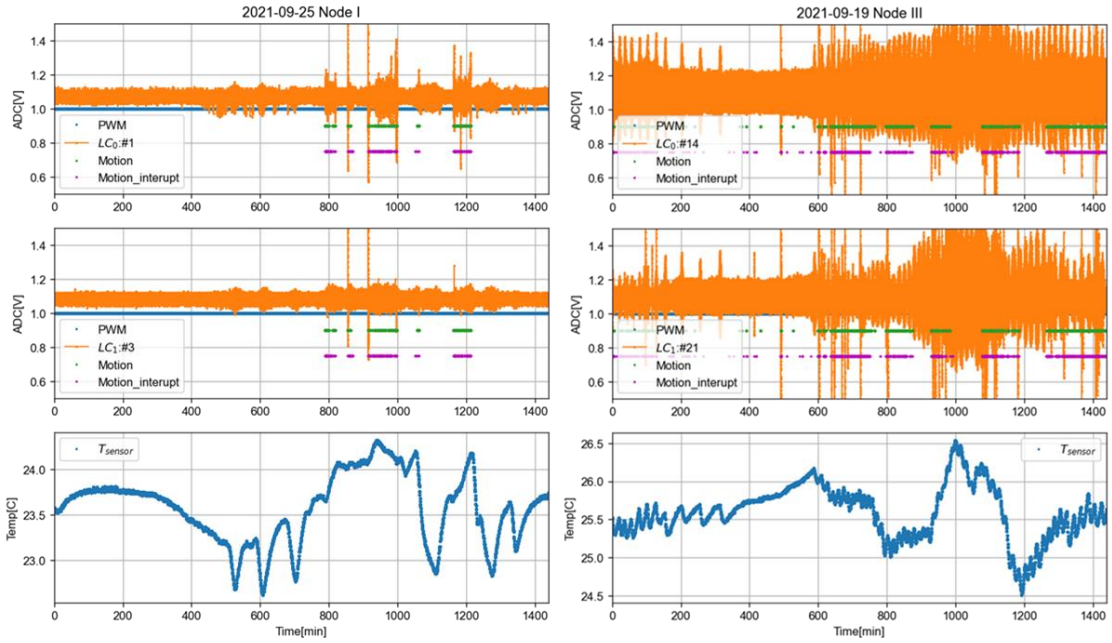


Figure 3.3 Plotting of daily SLEEPIR node output (a) lab (b) residential apartment

In addition, the daily maximum and minimum temperatures are listed in Figure 3.4. The temperature in the lab is much more stable in the day than that in the residential apartment. The max temperature is around 24˚C, and the minimum temperature is around 22.5 ˚C. On the other hand, the daily maximum temperature of the residential apartment room can reach 27 ˚C, and the daily minimum temperature is around 24˚C. Figure 3.3(a) and (b) show the typical daily raw sensor data plotting of the lab and residential apartment rooms, respectively.

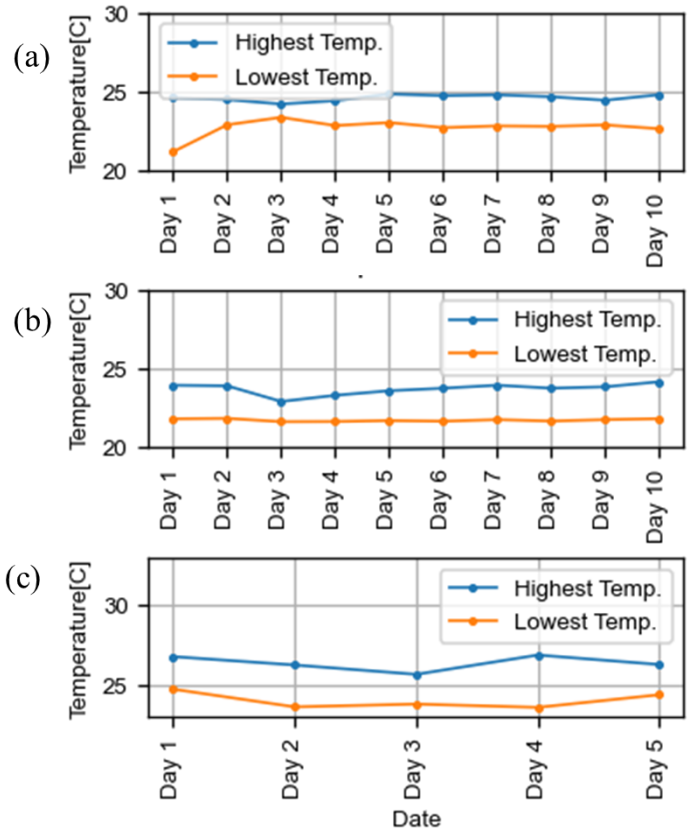
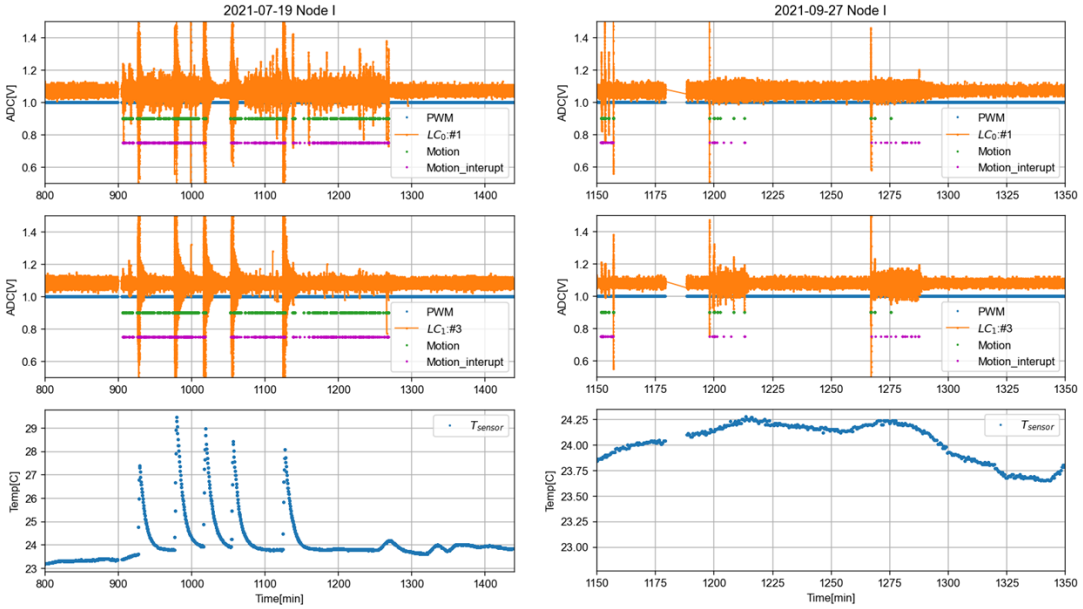


Figure 3.4 Daily maximum and minimum temperatures of the lab and residential room (a) LAB1 (b) LAB2 (c) APT1

Two edge cases are included in the dataset, lying on the bed, and changing ambient temperature. Figure 3.5 shows the setup of the two edge cases. Moreover, Figure 3.6(a) shows the raw sensor output during the ambient temperature changing experiment. The sensor temperature can be found to increase quickly multiple times within the day. Figure 3.6(b) shows the sensor output when the occupant is lying on the bed.

(a) (b)

Figure 3.5 Edge case experiment setup(a) ambient temperature changing (b) lying down



(a) (b)

Figure 3.6 (a) raw sensor data plotting (a) ambient temperature changing (b) lying on the bed.

3.3. Occupancy Presence Detection Algorithm

3.3.1. State Switch Detection Algorithm (StateS)

The core idea of the state switch detection (StateS) algorithm is to detect occupancy state change instead of directly detecting occupancy state. We can assume that motion can be detected by the onboard digital PIR sensor every time the occupancy state changes. Then we can compare samples before and after a motion period. If the occupant is still inside the room, the V_{pp} of the sample before and after a motion period should be very close. On the other hand, if the occupant left the room or entered the room, V_{pp} change should be observed. Figure 3.7 shows a sample plotting of this algorithm in which one occupant leaves the room and the occupancy state changes from 'occupied' to 'unoccupied'. Three periods are marked: stationary, motion, and occupied. The orange line represents the raw sensor data of SLEEPIR, and the red line represents the calculated V_{pp} values.

The motion period (the grey area) can easily be detected by the digital motion sensor or the SLEEPIR itself. After that, the no-motion period can be identified before and after the motion period (marked by two dashed brown boxes). Then the V_{pp} difference can be calculated, and we can find that the V_{pp} difference is about 0.24V, which can be used to indicate that the occupancy state has changed. Table 3.2 shows the pseudo-code to combine V_{pp} from two SLEEPIR modules and to determine whether the occupancy state is changed.

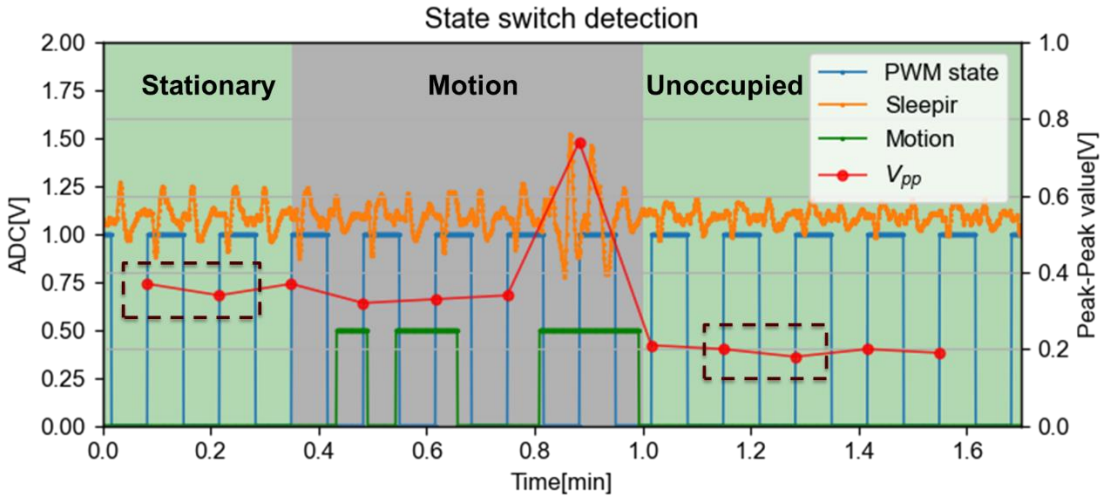


Figure 3.7 Example using the state switch detection algorithm

Table 3.2 Pseudo code of using the StateS detection algorithm

Algorithm: State switch detection
INPUT: A, B (2x3 matrix), $V_{threshold}$ **OUTPUT:** State_change(True, False) ΔV_{pp1} = avg(A[0,:]) – avg(B[0,:]) ΔV_{pp2}= avg(A[1,:]) – avg(B[1,:]) ΔV= abs(ΔVpp1 + ΔVpp2) **If** $\Delta V_{pp} > V_{threhold}$: State_change = True **Else:** State_change = False

3.3.2. Statistical Features + Machine Learning (Stat.ML)

The StateS algorithm detects the change of occupancy state instead of directly detecting occupancy state. Then a single false detection may cause an accumulative error; furthermore, the StateS algorithm only uses the V_{pp} value from each sample as the input; more features may be extracted to improve the detection accuracy with the machine learning classification models. The data within each actuation circle of SLEEPIR is single-dimensional time-series data. Many well-established features can be extracted from temporal, statistical, and spectral domains[99, 100].

Table 3.3 shows the features extracted in this work. During each actuation circle, 20 features are extracted from the output of each SLEEPIR module. Then a traditional machine learning classifier (NN, SVM, RF) is used to do the occupancy state detection.

Table 3.3 Features for statistical machine learning

Feature name (dimension)	Feature name (dimension)	Feature name (dimension)
Mean (1)	RMS (1)	FFT coefficient (5)
Variance (1)	Kurtosis (1)	Entropy (1)
Standard deviation (1)	Skewness (1)	Peak positions (5)
Max value (1)	Min value (1)	Absolute energy (1)

3.3.3. LSTM based Deep Learning Models: LSTM and Stat. LSTM

Statistical machine learning only uses features extracted from a circle of data (8s) to determine the occupancy state. However, the time-series feature may help to improve

detection accuracy. LSTM network is a powerful method to learn time-series features. Figure 3.8 compares the model systematic between statistical machine learning and deep learning based on the LSTM network. Two types of LSTM networks with different inputs are included. The LSTM model uses V_{pp}, temperature, and motion/digital PIR sensor output data as the input, and the Stat. LSTM model uses features listed in Table 3.3 as the input.

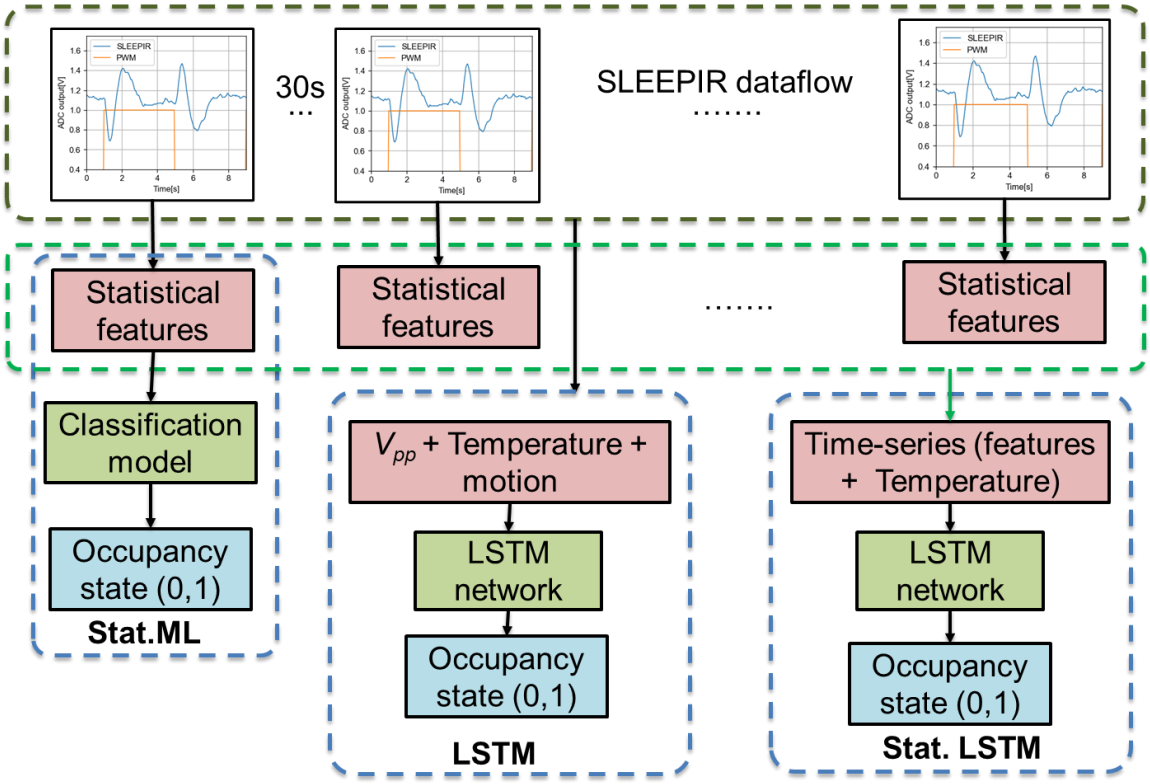


Figure 3.8 Comparison of Stat.ML, LSTM, and Stat.LSTM models

Figure 3.9 shows the structure of the two LSTM networks. A max-min normalization is performed to convert all features to the range [0,1] before being fed to LSTM layers. The LSTM network has two LSTM layers and one fully connected layer

with 128 neurons. The sequence length is 100, so we use 50 minutes of data to predict the occupancy state.

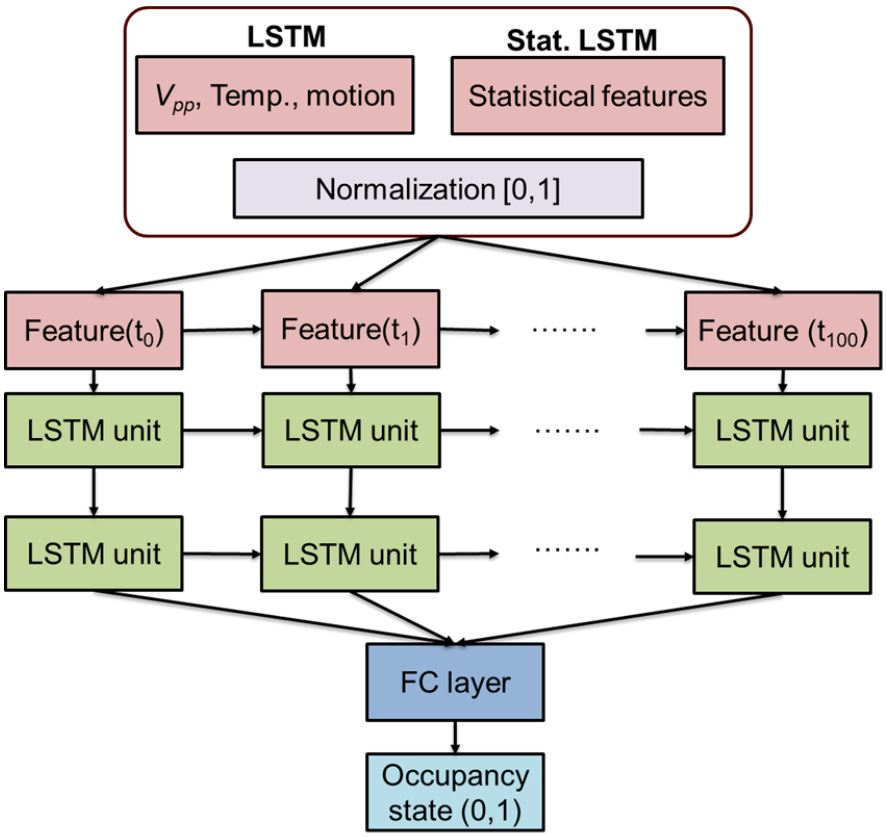


Figure 3.9 LSTM network structure: LSTM, Stat. LSTM

3.4. Result Analysis

3.4.1. Performance Comparison on Daily Occupancy Detection

The different occupancy presence detection algorithms are compared by reporting their daily occupancy detection accuracy. Each row in Table 3.4 shows the comparison of detection accuracy of a day using different algorithms. 'PIR' and 'PIR30' are two kinds of baseline algorithms. 'PIR' is to use the output of the digital PIR sensor onboard directly

as the occupancy state. 'PIR (30)' set an offset time of 30 minutes to the output of the digital PIR sensor. In other words, once the motion is detected, the 30 minutes after that will be considered as 'Occupied'.

Table 3.4 Comparison of daily occupancy detection: by dates

Dataset	PIR	PIR(30)	StateS	Stat.ML	LSTM	Stat. LSTM
LAB2/Coat	0.7644	0.9545	0.9128	0.8745	0.9732	0.9642
Model LAB	0.4647	0.9587	0.8719	0.7242	0.9568	0.9632
	0.4671	0.9413	0.8713	0.7845	0.9686	0.9654
	0.4743	0.9218	0.8687	0.7821	0.9652	0.9527
	0.5072	0.9340	0.8801	0.8576	0.9543	0.9532
LAB1/T-shirt	0.7461	0.9751	0.9314	0.9214	0.9776	0.9612
Model LAB	0.7312	0.9684	0.9517	0.9067	0.9612	0.9632
	0.7816	0.9520	0.9412	0.8547	0.9653	0.9746
	0.8541	0.9155	0.9317	0.8921	0.9337	0.9709
	0.8621	0.8933	0.9123	0.9185	0.9626	0.9713
Avg.	**0.6652**	**0.9414**	**0.9073**	**0.8516**	**0.9618**	**0.9639**
APT1	0.7378	0.9547	0.8714	0.8924	0.9746	0.9812
(Model APT)	0.8366	0.9888	0.8217	0.9043	0.9682	0.9732
Avg.	**0.7872**	**0.9718**	**0.8465**	**0.8983**	**0.9714**	**0.9772**

The deep learning model is first trained using five days from dataset LAB1 and five days from dataset LAB2, and this is considered the 'Model LAB'. Then, the other ten

day's data in LAB1 and LAB2 datasets are used as the training dataset. Finally, for the residential dataset APT1, another model, 'Model APT', is trained with three days of data.

The result shows that both LSTM and Stat. LSTM reaches reliable high accuracy >96.0% for both the lab room and residential apartment. Moreover, the Stat. LSTM model does not require the data from a digital PIR sensor. Furthermore, the same model trained with lab data is evaluated on the edge case dataset. As a result, high accuracy is also observed, which indicates that the proposed model is a reliable occupancy presence detection method. Table 3.5 shows the daily occupancy accuracy of different testing subjects. It shows that the proposed model Stat. LSTM can still reach high accuracy.

Table 3.5 Comparison of daily occupancy detection: by subjects

Subject	**info**	PIR	StateS	Stat.ML	LSTM	Stat.LSTM
Male 1	T-shirt	0.7645	0.9314	0.9214	0.9776	0.9612
	Coat	0.7816	0.9412	0.8547	0.9653	0.9746
Male 2	Coat	0.7644	0.9128	0.8745	0.9732	0.9642
Male 3	T-shirt	0.8721	0.9214	0.8963	0.9532	0.9715
Female 1	Coat	0.7966	0.8221	0.8825	0.9532	0.9587

3.4.2. Performance Comparison on Edge Cases

The edge cases data set is only used as testing data, and the model trained with the lab dataset,' Model LAB' , is used to output the result. Table 3.6 shows the detection result on edge cases. The proposed Stat. LSTM deep learning model still shows high accuracy for the edge cases.

Table 3.6 Comparison of performance on edge cases

Dataset	PIR	PIR(30)	StateS	Stat.ML	LSTM	Stat. LSTM
EDGE1	0.7845	0.9418	0.9127	0.8623	0.9763	0.9531
Heating	0.8064	0.9148	0.8924	0.8525	0.9814	0.9725
EDGE2	0.8943	0.9543	0.9127	0.8716	0.9683	0.9721
Lying down	0.9329	0.9329	0.8924	0.8415	0.9614	0.9635
	0.9253	0.8857	0.9215	0.8916	0.9514	0.9721

3.5. Conclusion

Experiments for occupancy presence detection are performed in a lab room and a room in a residential apartment. In addition, two edge cases experiments are also performed, including changing the room temperature and lying on a bed. Finally, different occupancy presence detection algorithms are proposed and compared.

The StateS model is an algorithm that only uses the V_{pp} of SLEEPIR sensor output. It detects the occupancy state change by comparing the V_{pp} level before and after a motion period. Stat. Machine learning uses features extracted from every sample of SLEEPIR output to train a classification model. The LSTM is a deep learning approach that uses time-series features extracted from the sequence of SLEEPIR sensor samples. The feature includes V_{pp} from SLEEPIR modules, temperature reading, and the output of the digital PIR sensor. Stat. LSTM is another deep learning model using statistical features instead of only V_{pp} values, and it does not need the data from a digital PIR sensor.

The daily occupancy detection accuracy using different occupancy presence detection algorithms is compared. The results show that both LSTM and Stat. LSTM

reached a high accuracy >96.0% for both lab rooms and the residential apartment. Moreover, the Stat. LSTM model does not need data from a digital PIR sensor. That is, we can eliminate the digital PIR sensor. Furthermore, the same model trained with lab data (Model LAB) is evaluated on the edge case dataset. As a result, high accuracy is also observed, which indicates that the proposed model is a reliable occupancy presence detection method.

After all, the SLEEPIR sensor is still a sensor node with a single dimension of data output. Though it can accurately detect stationary occupancy, its functionality is limited by presence detection. Therefore, sensing devices with high dimensions of data output are investigated in the following chapters. More specifically, in chapters 4 and 5, data processing models are developed using thermopile array sensors for fine-grained occupancy sensing, including identification, fall, posture, and facing direction detection.

4. FALLING AND IN-BED POSTURE DETECTION USING THERMOPILE ARRAY SENSOR AND MACHINE LEARNING[*†]

4.1. Review of Machine Learning Methods for Occupancy Sensing using Thermopile Array

Room level occupancy counting can be accomplished with 82.56% accuracy by a thermopile array sensor with 4×16 pixels[63]. Here, handcrafted features are applied with statistical classification models such as KNN, SVM. Moreover, counting can also be obtained by installing the sensor node on top and side of the door frame and monitoring the walking direction[91, 101]. The Grid-EYE sensor with 8×8 pixels can be mounted on the ceiling, and handcrafted features are extracted for different occupancy sensing objectives. For example, the SVM classifier can classify occupancy activities such as falling, walking, and sitting with 94.8% accuracy[65]. The RF classifier can classify falling, walking, sitting, and lying with an F1 score of 0.92 [102]. In addition, the DT classifier can identify 12 kitchen activities such as the operation of the oven and dishwasher with 79.9% precision and 75.1% recall using the same sensor setup[72]. Gaussian and Kalman filters are applied to Grid-EYE sensor data for multiple person(1-3)localization[68]. Six Grid-EYE sensors can be installed on the ceiling at different positions for occupancy counting to cover a large meeting room, and the occupancy number is classified into three levels. The softmax regression method is applied to reach

---

[*] Part of this chapter is reprinted from “Infrared–ultrasonic sensor fusion for support vector machine–based fall detection” by Chen, Z. and Wang, Y. 2018. Journal of Intelligent Material Systems and Structures 29(9): 2027-2039. with the permission of SAGE Publishing.

[†] Part of this chapter is reprinted, with permission, from "Remote Recognition of In-Bed Postures Using a Thermopile Array Sensor with Machine Learning." By Chen, Z. and Y. Wang (2021). IEEE SENSORS JOURNAL 21(9): 10428-10436. Copyright © 2021 IEEE

81.98% accuracy[28]. Handcrafted features can also be extracted from multiple sensors. For example, a thermopile array with 4×16 pixels is fused with PIR and ultrasonic distance sensor data for sitting, standing and walking detection with 99.4% accuracy[71].

4.2. Case Study: Handcrafted Feature Extraction for Fall Detection

4.2.1. Experiment Setup

The sensor platform is set at 0.8m high in front of the user/occupant, as shown in Figure 4.1. Since our sensor system can rotate and follow the occupant, the sensor is set in front of the subject during our fall detection experiments for convenience. And the distance from the sensor to the occupant does affect the detection algorithm. Also, the occupant should not be too close to the sensor; otherwise, the motion of the occupant cannot be fully recorded. Here, three distances (1.2 m, 1.5 m, 1.8 m) are selected in the experiment for evaluating the algorithm performance.

The threshold R_{th} for motion detection (feature N_M) is set based on the background data. A total of 10 seconds of background data is first recorded before testing (before the occupant shows up). Then the maximum RMS (RMS values are calculated based on equation between continuous frames is calculated. R_{th} must be larger than this value to avoid false negative detection in activity segmentation. In other words, no activity should be segmented when there is no occupant in the FOV. Based on our experiment, we set R_{th} to be 0.6ºC. The thermal images and distance data mapping during the entire testing period are continuously recorded, transmitted, and saved. A total of 180 actions have been captured for each distance during this experiment. Two kinds of falling activities are considered in the experiment, forward falling and sideway falling. In addition to the falling

activity – sitting, standing, and stooping are selected as non-falling activities because they are commonly used in daily lives and have similar gestures with the falling action. To clarify, stooping is defined as stooping down to pick up an item from the ground and returning to the standing position. Figure 4.1 shows the specification of the experiment.

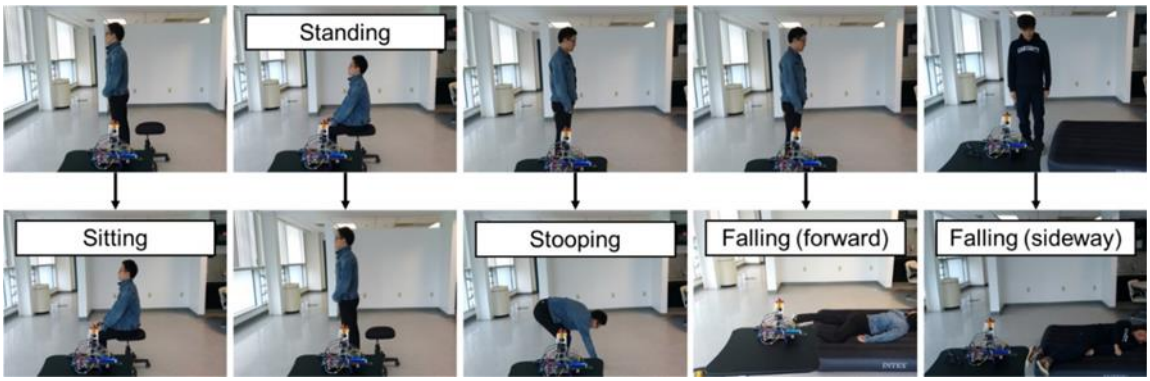


Figure 4.1 Snapshots of fall detection including five activity categories: sitting, standing, stooping, forward falling, and sideway falling. Reprinted from [64] ,with the permission of SAGE Publishing

Experiments are performed by three young volunteers in their early twenties; for each distance, each of them completed 15 times of forward falling, 15 times of sideway falling, 10 times of standing and sitting, respectively, and 10 times of stooping. Standing, sitting and stooping are all considered as the same category (non-falling) within the SVM algorithm. The 180 activities are treated as the training sets to calculate the SVM classifier for the continuous test. All the training sets are recorded individually, so a reliable classifier can be calculated. These 180 training sets are treated as discrete test databases, including 120 as the discrete training set and 60 as the test/validation set. Based on these discrete testing data, the reliability of the classifier is verified, and the best combination of features is found.

Table 4.1 Experimental Specifications of Fall Detection. Reprinted from [64] ,with the permission of SAGE Publishing

<table>
<tr><td rowspan="2">Training data</td><td colspan="3">Classifiers</td></tr>
<tr><td>Fall</td><td colspan="2">Non-fall</td></tr>
<tr><td rowspan="4">No. of captured actions for each occupant</td><td>Falling (forward)</td><td>Sitting</td><td>Standing</td></tr>
<tr><td>15</td><td>10</td><td>10</td></tr>
<tr><td>Falling (sideway)</td><td colspan="2">Stooping</td></tr>
<tr><td>15</td><td colspan="2">10</td></tr>
<tr><td colspan="3">No. of occupants</td><td>3</td></tr>
<tr><td colspan="3">Total No. of the dataset</td><td>180</td></tr>
<tr><td colspan="3">Environment temperature</td><td>24.5ºC</td></tr>
<tr><td colspan="3">Distance from the sensor to the occupant</td><td>1.2-1.8m</td></tr>
<tr><td colspan="3">Sensor setup height</td><td>0.8 m</td></tr>
</table>

However, in actual application scenario, user's activities are recorded continuously by the sensor fusion platform, and activities can happen one by one; therefore, another series of experiments are designed for continuous activity recording, following the order of sitting, standing, stooping, sitting, standing, forward falling, return to standing status, sitting, standing and sideway falling, as shown in Figure 4.2

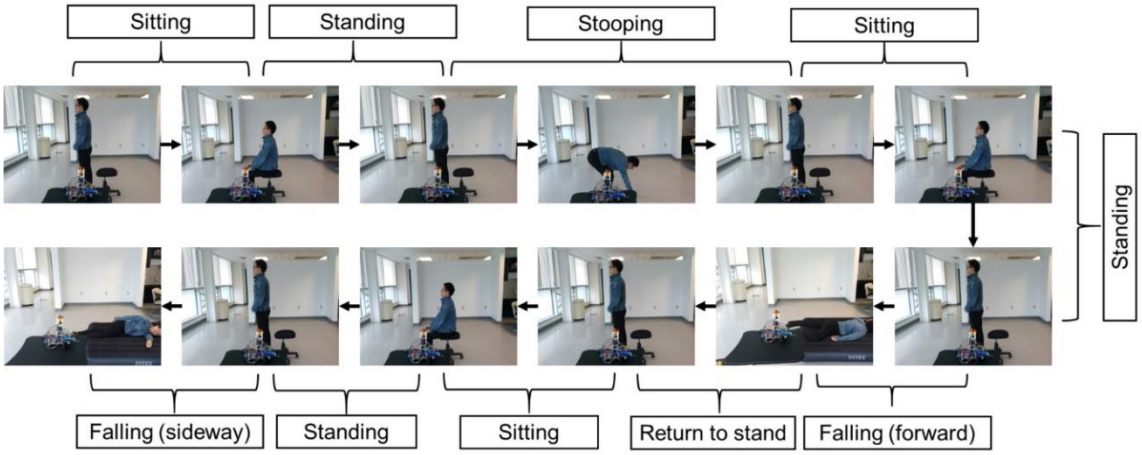


Figure 4.2 Snapshots of continuous activity data capturing including five activity categories: sitting, standing, stooping, forward falling, and sideway falling. Reprinted from [64] ,with the permission of SAGE Publishing

More specifically, these activities are performed continuously by three aforementioned users and repeated twice. That is, there are 6 continuous test segments recorded for each distance and are later on used as the test sets. As mentioned previously, each test segment contains 10 pre-defined activities, including 8 non-fall activities and 2 fall activities. At each distance, the aforementioned discrete data sets (180 total) are treated as the training sets for the continuous test, which do not include the activity from lying status to standing status. Figure 4.3 demonstrates an example of 1 test segment with 10 continuous activities recorded at 1.5 m.

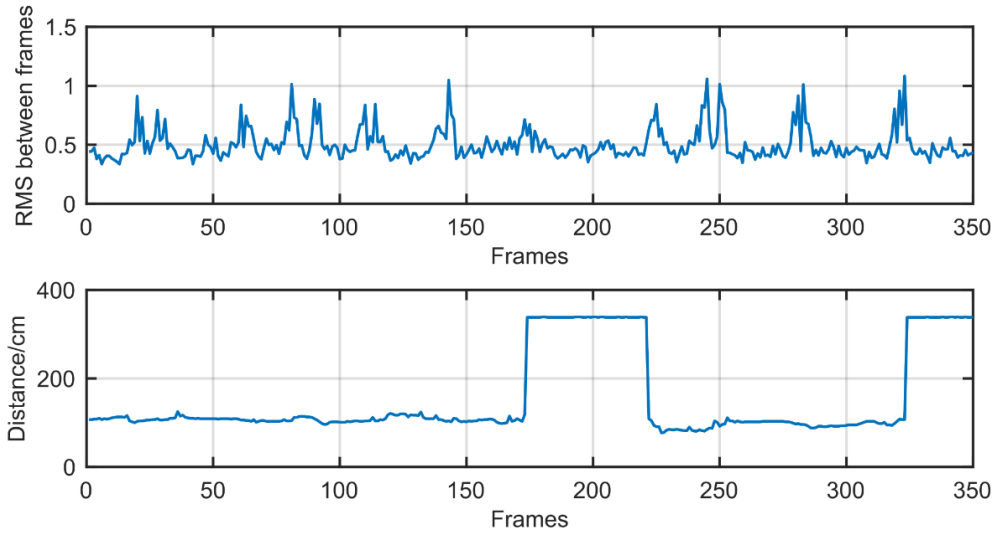


Figure 4.3 An example of 1 test segment containing 10 continuous activities: RMS values were calculated based on equation (7). Reprinted from [64] ,with the permission of SAGE Publishing

4.2.2. Handcrafted Feature Extraction

A feature set is extracted from each activity, and a fall or non-fall category will be further identified through the SVM classifier. Here, the feature set includes four features extracted from the Grid-EYE and one feature from the distance sensor:

(1) N_M: The number of consecutive frames where motion is detected, shown in Figure 4.4

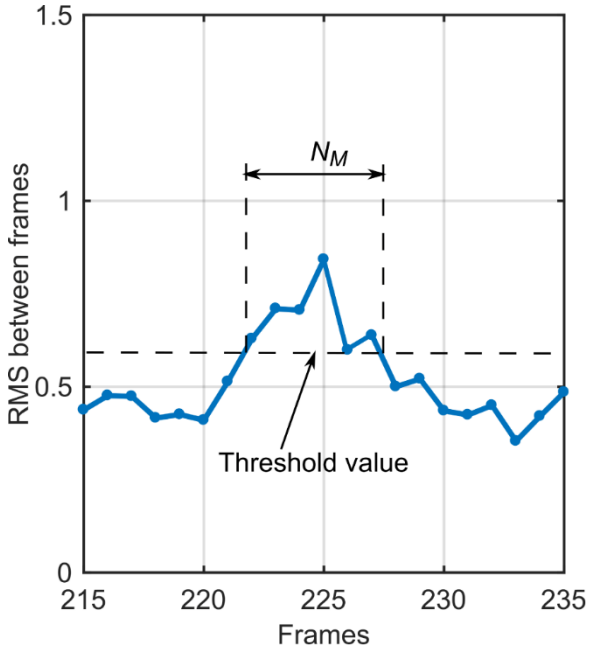


Figure 4.4 The RMS values between consecutive frames indicating feature N_M Reprinted from [64] ,with the permission of SAGE Publishing

The root mean square (RMS) between two consecutive frames is calculated based on the following equation:

$$RMS_k = \sqrt{\frac{\sum_{i=1}^{8}\sum_{j=1}^{8}(T_{i,j,k+1}-T_{i,j,k})^2}{64}} \qquad (7)$$

Where $T_{i,j,k}$ represents the Grid-EYE pixel data at coordinate *(i, j)* on the k_{th} frame.

If the RMS is higher than a threshold value R_{th}, then a motion is detected; all consecutive frames together can be taken as an activity occurring. N_{M} is the total number of frames for an activity, as shown in Figure 4.4. Since the data is received at a constant frequency, this feature reflects the duration of the activity.

(2) T_{M}: Peak value of each pixel's temperature change before and after an activity (from the first to the last frame of N_{M}).

$$T_M = \max\left[T_{i,j,k_0+N_M} - T_{i,j,k_0}\right], 1 \le i \le 8, 1 \le j \le 8. \quad (8)$$

Here k_0 is the index of the first frame of an activity. This feature tends to be large when a fall occurs since a fall is usually associated with a large acceleration.

(3) T_C: Peak temperature change of each pixel between two consecutive frames.

$$T_c = max\left[\max\left[T_{i,j,k+1} - T_{i,j,k}\right] 1 \le i \le 8, 1 \le j \le 8\right] k_0 \le k \le k_0 + N_M \quad (9)$$

This feature reflects the user's speed in action and tends to be large when a fall occurs.

(4) D_M: Distance between the maximum temperature pixel before and after an activity.

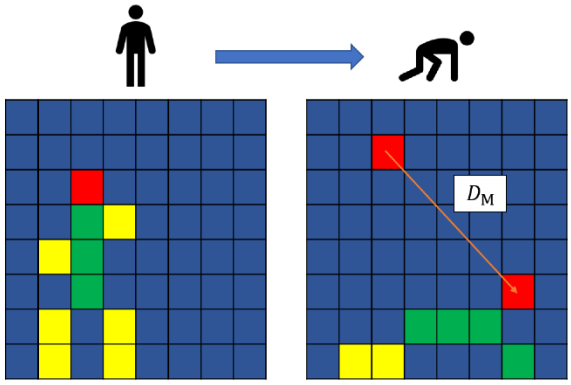


Figure 4.5 Schematic of 8 x 8-pixel thermal image indicating feature D_M. Reprinted from [64] ,with the permission of SAGE Publishing

The feature D_M is the Euclidean distance between the positions of the pixel where the maximum temperature is observed in 2 frames before the start of N_M and 2 frames after the end of N_M. (Figure 4.5)

(5) D_{diff} : the range sensor data difference between the start and end of an activity.

$$D_{\text{diff}} = C \times (U_{k_0+N_M} - U_{k_0}) \quad (10)$$

where U_k is the distance sensor data of the k_{th} frame. Since the original D_{diff} values are much larger than other feature values, a coefficient is added to help calculate the classifier.

Here the coefficient value C is set as 1/100 to ensure that D_{diff} has a similar data range as other features.

Since the sensor data is collected continuously, one activity must be identified before features can be extracted from the next activity and applied to classification. In other words, the time frame corresponding to the beginning and the end of each activity needs to be identified. In this paper, the data sequence is firstly separated based on the feature N_{M}, and then the SVM classifier is used to determine if the activity is 'fall' or not. Figure 4.6 shows the diagram of activity identification at a continuous period.

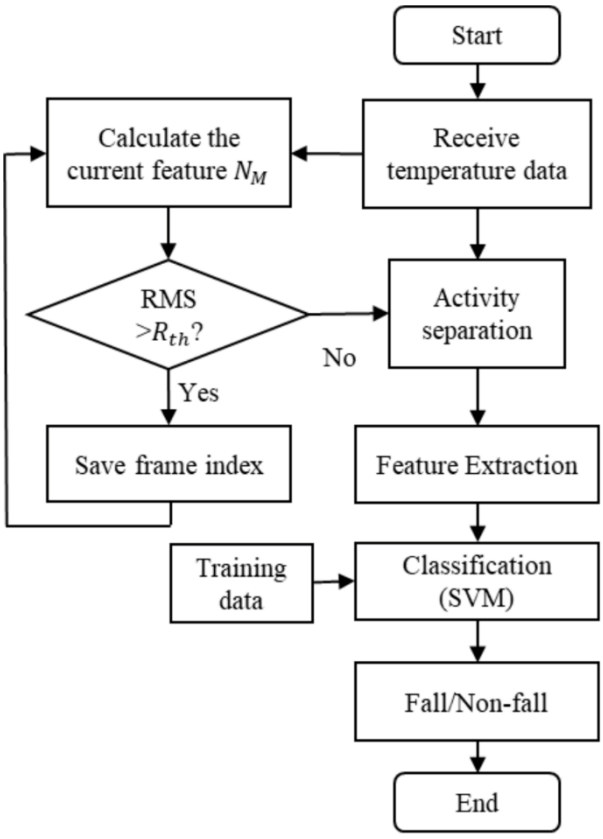


Figure 4.6 Schematic diagram of the fall detection algorithm. Reprinted from [64] ,with the permission of SAGE Publishing

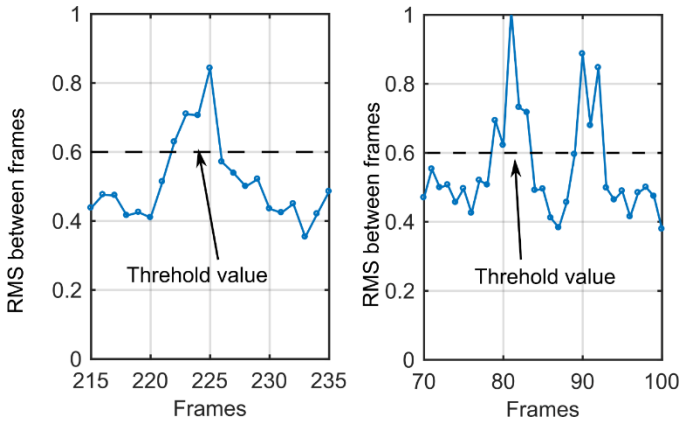


Figure 4.7 (a) Activity period with frames containing a single peak (b) Activity period with frames containing multiple peaks: RMS values were calculated based on equation (1) Reprinted from [64] ,with the permission of SAGE Publishing

4.2.3. Experimental Results

To verify the functionality of the fall detection algorithm, two testing methods (discrete and continuous) are executed. For discrete testing, 60 out of 180 data sets are treated the test data and the remaining 120 data sets are the training sets. Results are listed in Figure 4.8, where the accuracy is averaged by ten different selections of test sets. One can find out that the feature sets containing features extracted from both the Grid-EYE and the range sensor always achieve higher accuracy than these containing features from the single sensor. For instance, at a distance of 1.2m, the feature set (T_C D_M T_M) reaches the highest accuracy of 93.3% when the stand-alone Grid-EYE sensor is applied. However, with the assistance of the range sensor, the accuracy is increased to 99.7%. At a distance of 1.5m, the feature set (N_M T_C T_M) reaches the highest accuracy of 94.7% when the stand-alone Grid-EYE sensor is applied. However, with the assistance of the HC-SR04, only two other features T_C and T_M are required to reach the highest accuracy of 98.7%. At the

distance of 1.8m, the detection accuracy becomes lower. It can be caused by the decrease of temperature difference between the occupancy body temperature and the ambient environmental temperature as well as reliability of the ultrasonic sensor. While we can still find out that the highest accuracy 88.7% is reached when the stand-alone Grid-EYE sensor is applied. And with the assistance of the HC-SR04, the accuracy is increased to 91.3%.

This discrete testing also helps us to find the best feature set. By summarizing the accuracy from three distances, we select feature set (T_C D_M T_M) and (T_C T_M D_{diff}) as the best combination of features when only data from Grid-EYE sensor is used and when both sensors are applied. These two combinations of features will be used for classification in continuous experiments. Figure 4.9 illustrates the feature space of these two feature sets. Feature sets (N_M T_C T_M D_{diff}) and (T_C D_M T_M D_{diff}) also have higher accuracy, however, since they both require four features, we consider (T_C T_M D_{diff}) as the best feature set even though its accuracy is about 2% lower than that of the above two feature sets. The detection result for the continuous testing-based feature set (N_M T_C T_M D_{diff}) and (T_C D_M T_M D_{diff}) are also included.

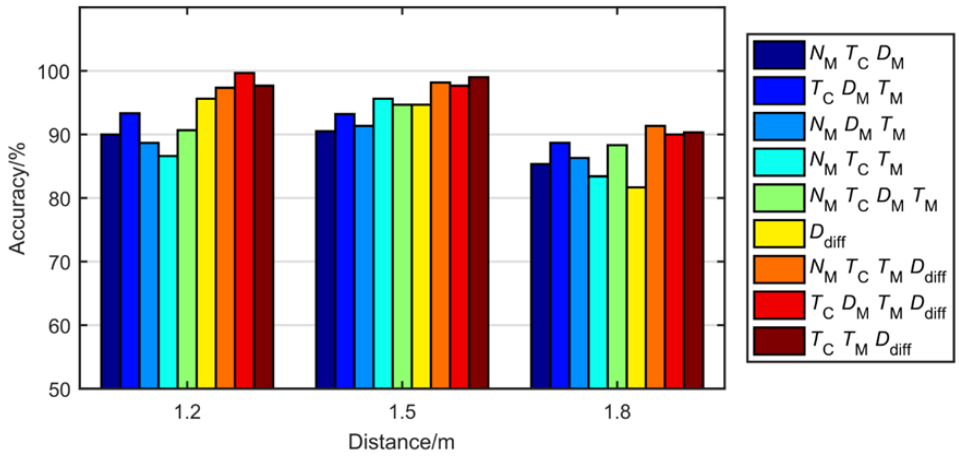


Figure 4.8 Feature analysis for SVM classifier. Reprinted from [64] ,with the permission of SAGE Publishing

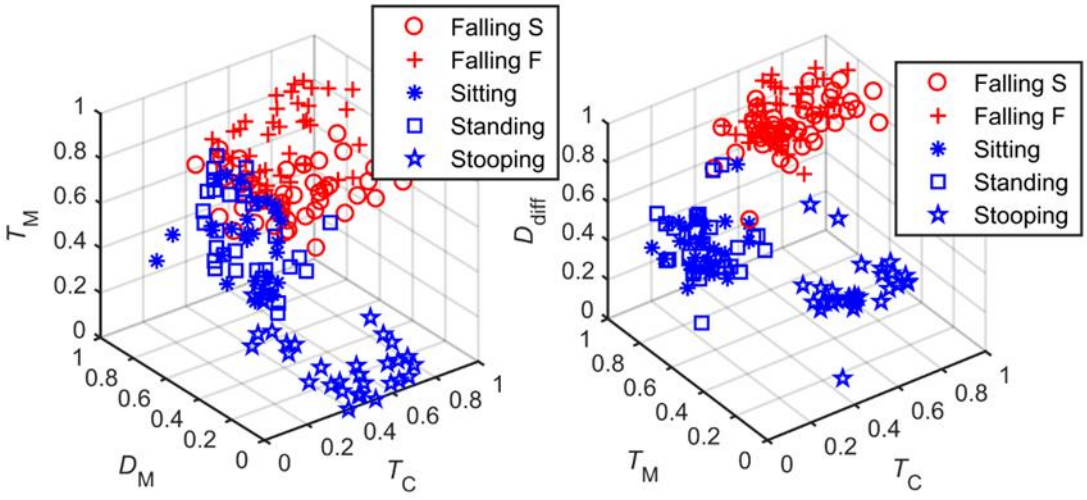


Figure 4.9 Feature space plotting of feature set (T_C , D_M , T_M) and (T_C , T_M , D_{diff}). Reprinted from [64] ,with the permission of SAGE Publishing

The second method uses the continuously recorded 18 test segments (six test segments at three distances, a total of 180 activities) as the test sets and each activity contains a certain number of frames. For each distance, a total of 180 activities obtained from the aforementioned discrete test are used as the training sets. Additional 6 continuous

testing segments, each containing 10 activities are used as the test sets. Figure 4.10 shows testing results of one of the testing segments with 10 testing activity periods (400 frames total), where all the 10 activities are identified correctly, and each activity period is marked by the red lines.

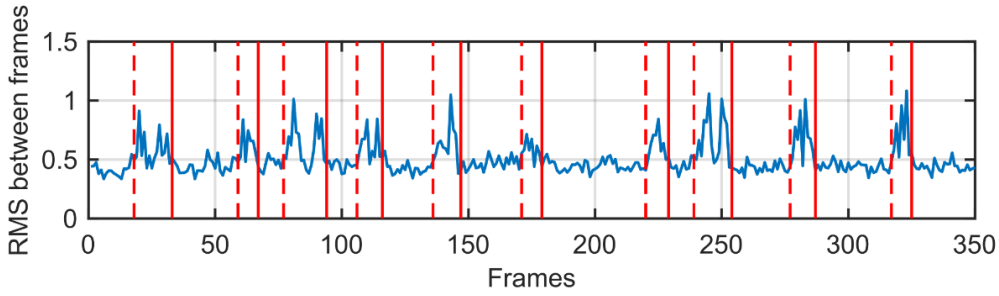


Figure 4.10 Results of activity identification: RMS values were calculated based on equation (7). Reprinted from [64] ,with the permission of SAGE Publishing

Figure 4.10 shows experimental results of activity segmentation, where Falling F represents forward falling and Falling S represents sideway falling. All the 'fall' activities are successfully segmented.

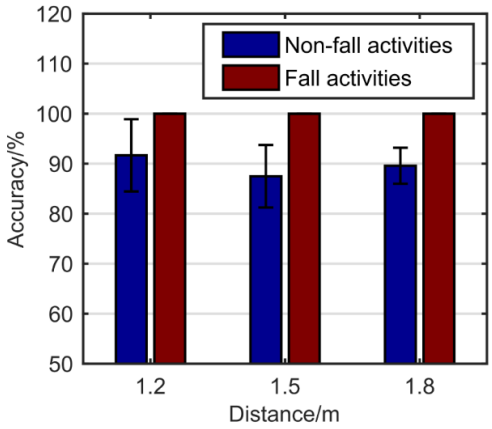


Figure 4.11 Accuracy on activity segmentation based on distance. Reprinted from [64] ,with the permission of SAGE Publishing

For non-fall activities, the overall segmentation accuracy is about 89.6%, and the inaccuracy is mainly due to false positive detection caused by stooping as shown in Figure 4.12, which can be further addressed by analyzing the change of the moving speed (the acceleration analysis between each frame) in the next step.

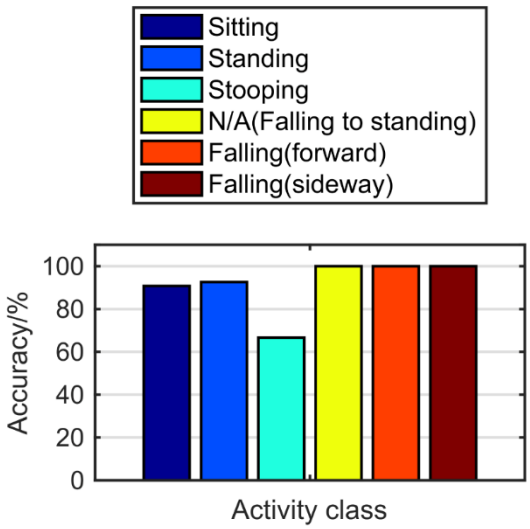


Figure 4.12 Accuracy on segmentation based on activity category. Reprinted from [64] ,with the permission of SAGE Publishing

The accuracy analysis of fall detection result based on continuous activity data recording is shown in Figure 4.13. By using the stand-alone Grid-EYE sensor, the accuracy for fall detection is not so satisfied, as shown in Figure 4.13(a), while a relative higher accuracy is reached by integrating the HC-SR04 ultrasonic sensor. Figure 4.13(b) shows the fall detection result based on feature set (T_C T_M D_{diff}). The detection accuracy on non-fall activities, fall activities and overall activities are plotted separately. At the distance 1.2 m and 1.5 m, all the 'fall' activities are successfully identified. While false negative detection does happen, the accuracy of non-fall activity detection accuracy is

about 89.3%, in other words, about 10.7% of non-fall activities are incorrectly detected as fall activities. When distance is increased to 1.8m, the false positive detection rate is smaller, while at the same time the false positive detection happens and the accuracy to detect a 'fall' is decreased. The overall accuracy keeps around 90.0%.

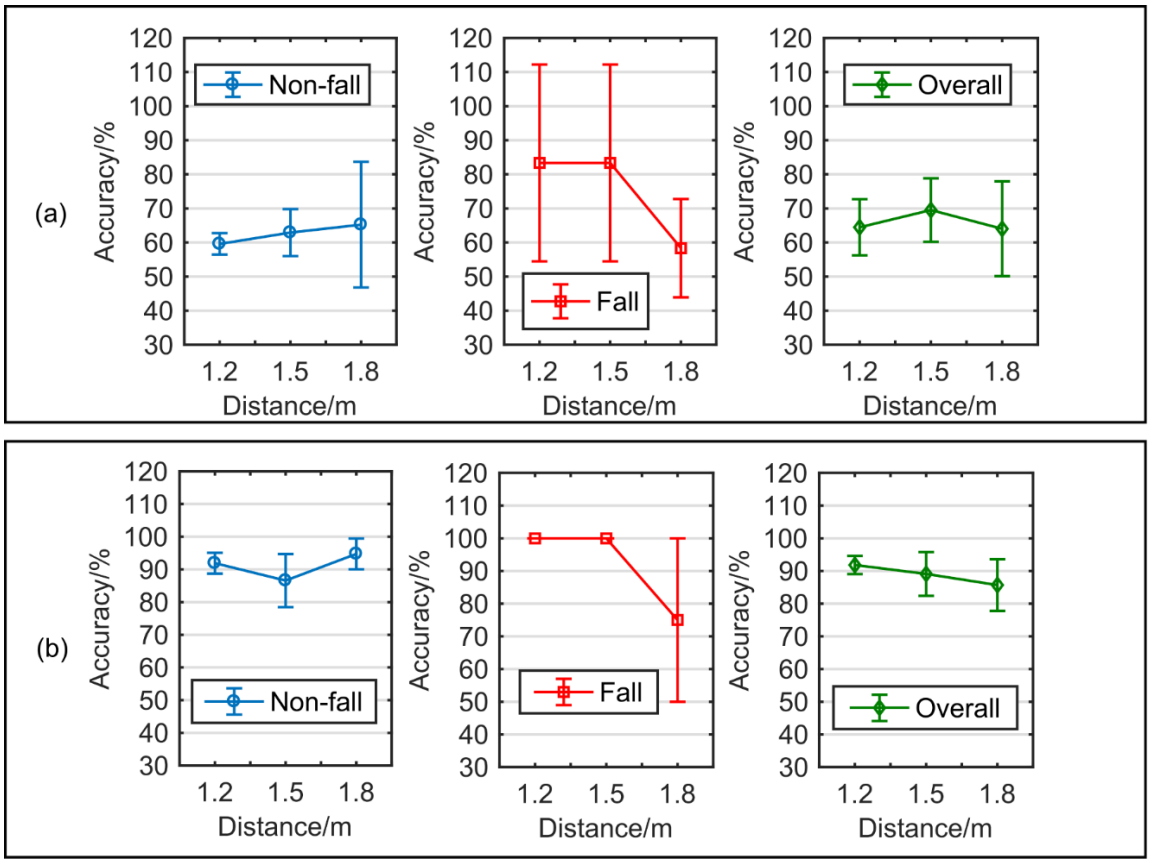


Figure 4.13 Continuous testing accuracy based on feature set (a) T_C T_M D_{diff} (b) T_C D_M T_M . Reprinted from [64] ,with the permission of SAGE Publishing

Fall detection accuracy based on feature set (N_M T_C T_M D_{diff}) and (T_C D_M T_M D_{diff}) are also included in Figure 4.14(a) and (b). Their true positive rate (detection accuracy on fall activities) are similar with that of feature set (T_C T_M D_{diff}). In addition, feature set (N_M T_C T_M D_{diff}) results in a higher false negative rate at closer distance, that means it tends to

detect a ‘non-fall’ activity as a ‘fall’ activity. While feature set (T_C D_M T_M D_{diff}) has a higher false negative rate when the distance increases. And all the results show a rising in the false positive rate when the distance increases. In summary, these analyses confirm that (T_C T_M D_{diff}) is the best combination of features.

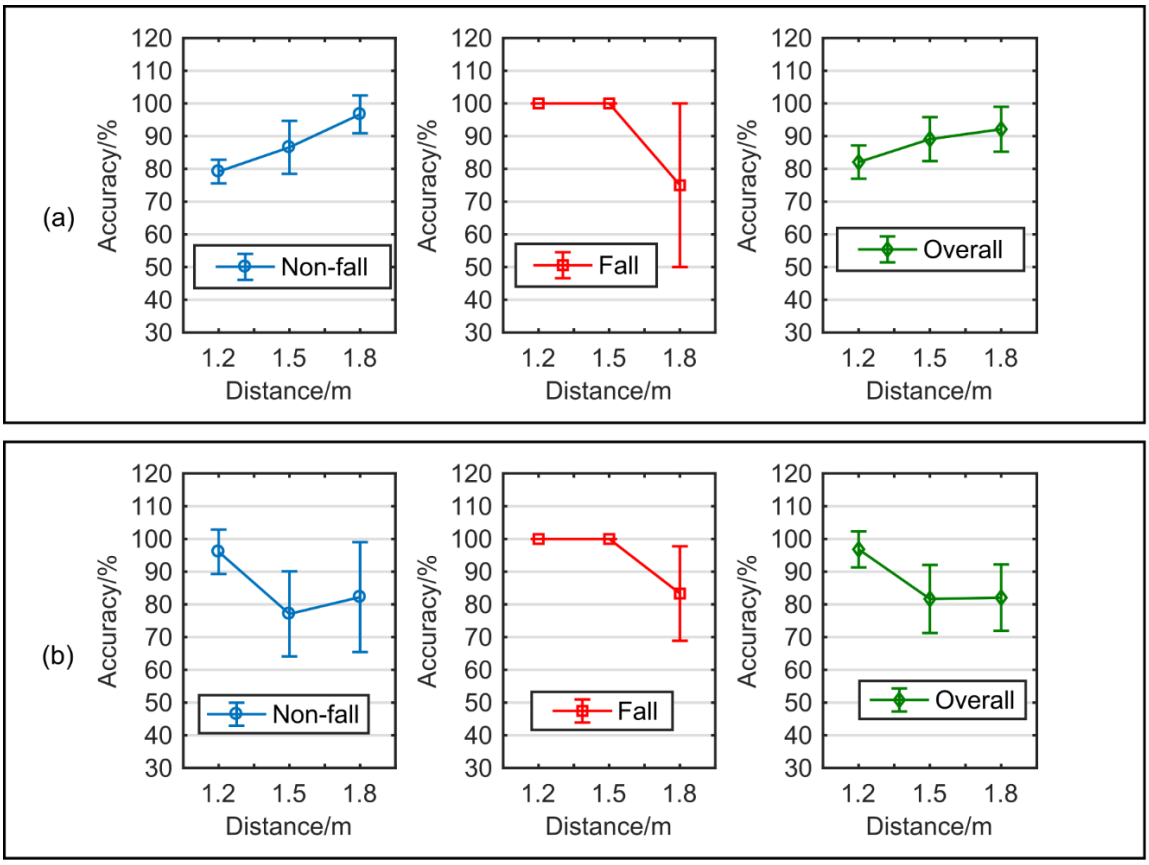


Figure 4.14 Continuous testing accuracy based on feature set (a) N_M T_C T_M D_{diff} (b) T_C D_M T_M D_{diff}. Reprinted from [64] ,with the permission of SAGE Publishing

4.3. Case Study: HOG Feature Extraction for in-Bed Posture Detection

4.3.1. Experiment Setup

The Type III thermopile array sensor node is used for in-bed posture detection. Two same sensor nodes are installed at two different locations for comparison. As shown

in Figure 4.15, one is placed on the top of the bed to simulate the ceiling-mounted condition, where the vertical distance between the sensor and the bed is 2.4m. Another sensor node attached to a tripod is placed next to the bed to simulate the wall-mounted condition. The vertical distance between the wall-mounted sensor node and the bed is 1.2m, and the sensor has an elevation angle of about -65°. This setting makes the FOV of the sensors slightly larger than the bed surface area to ensure the highest resolution of the posture images.

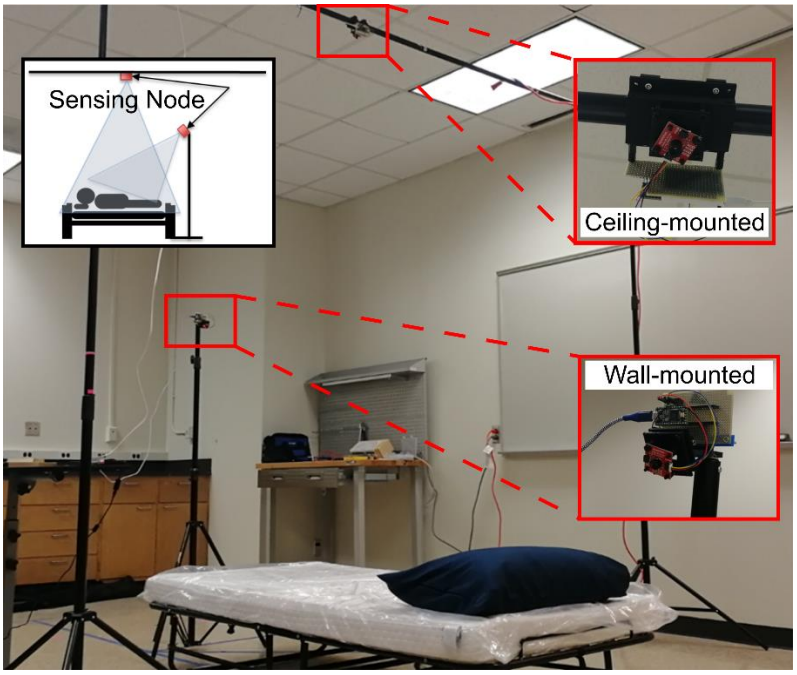


Figure 4.15 Experiment setup for in-bed posture detection. © 2021 IEEE. Reprinted with permission from [27].

A total of 12 young subjects (10 males and 2 females, 162 cm – 185 cm high and weighted between 48kg and 83kg) are tested in the experiment, following Texas A&M University institutional review board (IRB) 2018-1681D. The experiments were performed between 1-6 pm from 12/17 -12/22, 2019. One section lasts for 30 minutes for

one subject to perform different postures following the description in Figure 4.16(a). The total of 9 postures chosen from the most popular American sleep postures [103] include (1) soldier (SD), (2) left foetus (LF), (3) right foetus (RF), (4) left log (LL), (5) right log (RL), (6) freefaller (FF), (7) starfish (SF), (8) left yearner (LY), and (9) right yearner (RY).

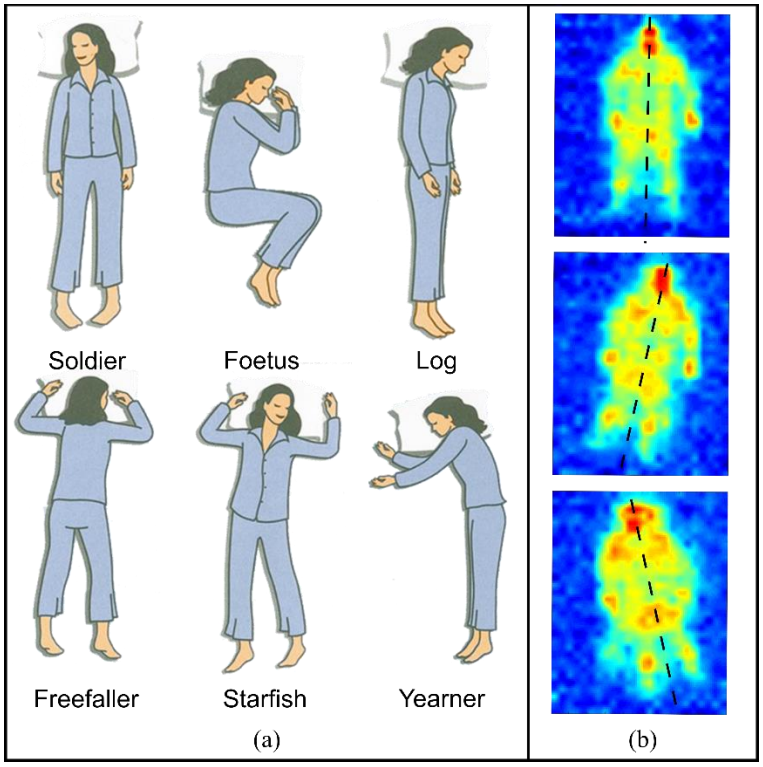


Figure 4.16 (a) In bed postures included in the experiment (b) Example plotting of different variants in posture 'Soldier'. © 2021 IEEE. Reprinted with permission from [27].

During the experiment, each subject is asked to naturally perform their postures but move their body around to generate variances within the same posture category shown in Figure 4.16(a). For instance, Figure 4.16 (b) shows the different images captured for the same posture 'Soldier'. This way, postures containing different variants are recorded

to generate a more comprehensive dataset. In other words, the same posture dataset contains images with different orientations and locations of the torso and the limbs. The experiments were performed on five different days. Each subject maintains each in-bed posture for about 60 seconds to get about 480 frames of data. Then for each posture, the data is downsampled to 60 frames to generate the dataset. In other words, the dataset totals 60 (frames) × 9 (postures) × 12 (test subjects) = 6480 frames of data. In addition, we can consider the highest pixel value within each thermal image to be the user body temperature and the lowest pixel value to be the ambient temperature. Then the user body temperature range of this dataset is 29.1°C to 33.7°C, and the ambient temperature range is 19.8°C to 21.25°C.

4.3.2. HOG Feature Extraction

A preprocessing of the raw sensor data is needed first for extracting robust HOG features. Since both the HOG and hand-crafted features are highly sensitive to the orientation and location of the user body, the primary goal of preprocessing is to align the user body part to the center of the thermal image, which facilitates robust feature selection. The preprocessing contains five steps: interpolation, temperature scaling, binarization, noise filtering, as well as rotation and translation.

Figure 4.17 shows the result of each step of preprocessing based on a raw thermal infrared image of soldier posture. First, the 32×24 raw image (Figure 4.17(a)) is interpolated to a 128×96 image (Figure 4.17 (b)). Next, all the temperature values in the image are scaled to the range [23 32] ºC using the min-max normalization method (Figure

4.17 (c)). This step addresses the impact of the person-to-person body temperature difference, and then binarization can be performed based on a fixed threshold value to highlight the user body part. 26.5ºC is used as the threshold value, which is selected as it returns the best detection accuracy compared with that using other threshold values. A filtering step is applied to keep the binary image with the largest connected area as is, while that with smaller connected areas is treated as background noise. Figure 4.17 (d) shows the result after binarization and filtering. Next, the translation and rotation are processed since different users can lie on different positions of the bed with different angles even for the same posture. Figure 4.17 (e) shows that the highlighted area can be rotated and translated to align the centerline of the bed by considering it as an eclipse and calculating its determinant direction and center based on the polar moment of inertia and weight center. Finally, for feature extraction, all the pixels with the value '1' of the highlighted area (dark red) in the binary image are replaced with their original values in the interpolated image, and the rest areas are filled with the threshold value as shown in Figure 4.17 (f).

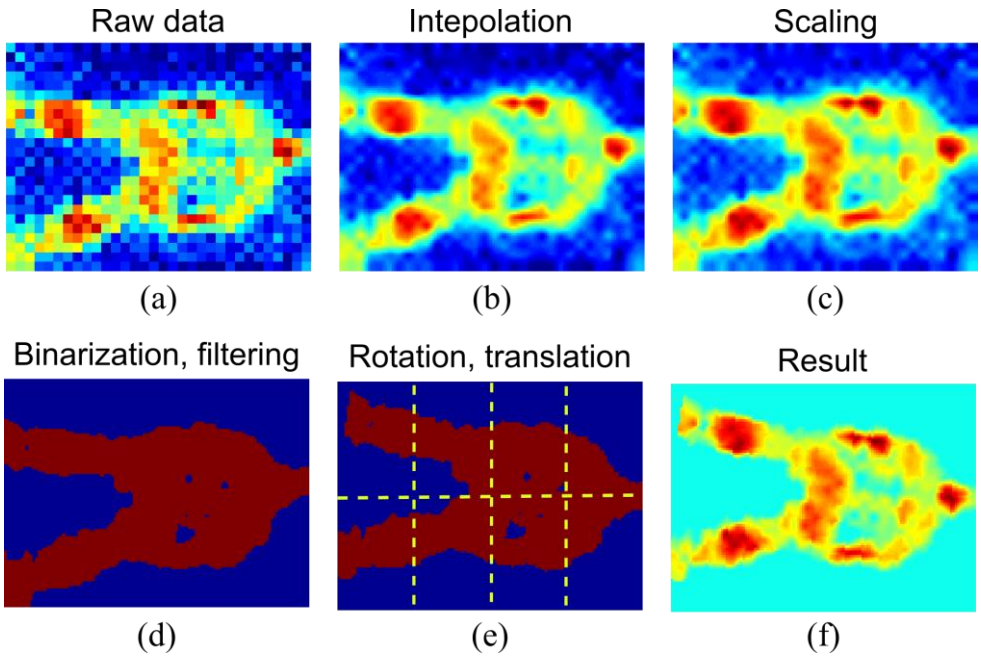


Figure 4.17 Data preprocessing for in-bed posture detection. © 2021 IEEE. Reprinted with permission from [27].

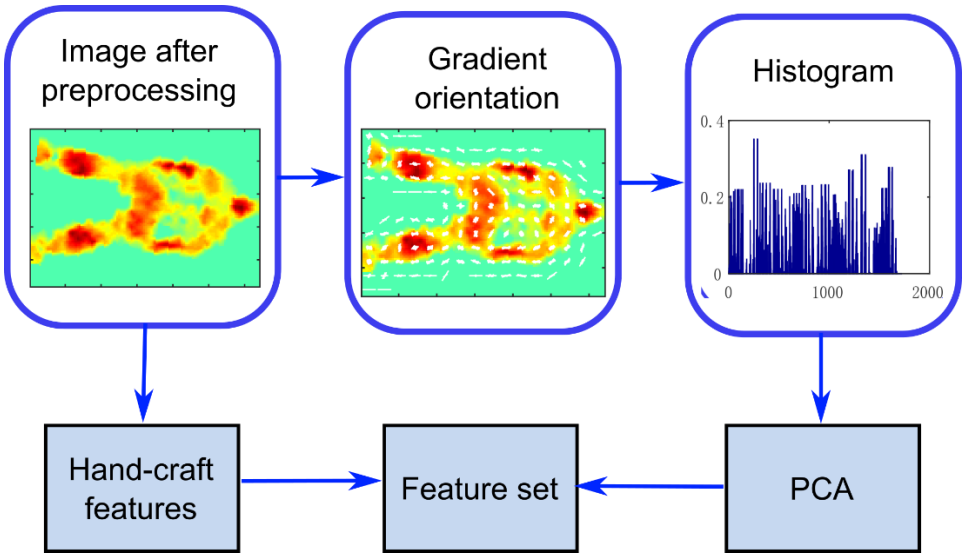


Figure 4.18 HOG feature extraction for in-bed posture detection. © 2021 IEEE. Reprinted with permission from [27].

HOG is a well-known method used in computer vision for object detection [83] by calculating the subject's edge direction (orientations of gradients). The edge directions can outline the shape of the user body postures, then their histogram values calculated based on the gradient orientations can be used as effective features for posture identification.

There are several key parameters in the HOG process that can be tuned to generate different feature vectors of the same raw image.

1) Cell: at first, the image is divided into different rectangular pixel regions (8×8 in this work), and each region is called a cell. For each cell, we can calculate its oriented gradients. Then the number of histogram bins is set as 9 to calculate the histogram bin values for each cell.

2) Block: the image is then divided into larger rectangular pixel regions, and each region is called a block. Each block contains a group of cells (4×3 cells in this work). The histogram values of each block are normalized to increase the feature robustness. The block overlap is another important parameter, but we set it as 0 since all thermal images have been preprocessed to align with the centerline. Finally, histogram values generated from all cells are connected to generate the feature vector. As every thermal image contains 16×12 cells after preprocessing and each cell has 9 histogram bin values, a feature vector with the size of 1728×1 can be generated for each thermal image frame.

4.3.2.1. Principal Component Analysis

The HOG feature vectors contain redundant data, and lots of null values and their dimension (1728) are much higher than the raw input data (768). To address this, we use PCA to reduce dimensions of HOG feature vectors and thus reduce overfitting and

redundancy in the feature set. PCA uses an orthogonal transformation to convert the set of possibly correlated features into a set of linearly uncorrelated values known as principal components. The first principal component has the largest possible variance, which represents the most variability in the dataset. PCA can work as a feature selection method that converts the original large feature vector to a short feature vector, including only important features. As a feature set of 1728×1 can be extracted from every thermal image data frame, the PCA process helps reduce the computing cost.

4.3.3. Handcrafted Feature Extraction (Baseline)

To evaluate the proposed HOG feature extraction approach, we use the hand-crafted feature extraction method as the baseline for comparison. The hand-crafted features, developed based on temperature distribution and spatial relationship [104], are listed in Table 2. A histogram is calculated based on the temperature value distribution of the thermal image after preprocessing (Figure 4.17 (f)). The bar width of this histogram is set as 1°C, and the temperature range is set as 27°C - 32°C. Thus, the histogram only contains 5 bars, and the values of each bar height are used as features 1 to 5.

Features 6 to 9 contain 4 properties (mean, variance, skewness, kurtosis) of the histogram, respectively. Features 10 to 18 contain the spatial features of the thermal image. They can be calculated from the binary image shown in Figure 4.17 (e). Feature 10 represents the number of pixels with the value of '1' in the binary image (dark red color). Feature 11 to 18 represent the number of pixels with a value of '1' of each of the equally sized subdivisions (totally eight, separated by the yellow dashed lines in Fig. 3(e)) divided by the value of feature 10. The total number of pixels of each image after interpolation is

96×128 = 12,288, and thus each subdivision of image has the number of 48×32 =1,536 pixels.

Table 4.2 Hand-crafted Features. © 2021 IEEE. Reprinted with permission from [27].

Feature ID	Description
1-5	Thermal histogram value (27°C - 32°C)
6	Mean of histogram
7	Variance of histogram
8	Skewness of histogram
9	Kurtosis of histogram
10	Number of pixel '1' in the binary image
11 - 18	Number of pixel '1' within each subdivision of binary image (8 total), divided by value of feature 10

We use Information gain to identify the best feature subset of hand-crafted features The Information gain measures how much 'information' a feature provides about the classification. The information gain can be used to rank the importance of different features [105]. The information gain is given by calculating the reduction between the entropy and conditional entropy [106].

$$IG(X,F) = E(X) - \sum_{v \in F} \frac{X_v}{X} E(X_v)$$

Here X_v is the subset of dataset X in which feature F has the value of v, and the entropy is given by

$$E(X) = -\sum_i p(x_i) \log_2 p(x_i)$$

, where $p(x_i)$ is the probability of x_i occurring in the dataset X . The feature with higher IG values is more important than those with lower values. Then the best feature subset with *n* features can be determined by choosing *n* of the most important features. In this paper, IG is used to evaluate all the hand-crafted features.

4.3.4. Experimental Results

4.3.4.1. Classification Models

It is challenging to select a suitable classifier without first evaluating the datasets. So, it is worth testing several popular algorithms on the dataset. A list of classification algorithms compared here includes the LR, the Support Vector Machine (SVM), the Decision Tree (DT), the Random Forest (RF), the Gradient Boost Decision Tree (GBDT), and the multi-layer perceptron (MLP), a traditional feed-forward neural network. Scikit-learn, a widely used python library of machine learning mainly used for the implementation of the algorithms. For the SVM method, both the linear kernel and radial basis function (RBF) kernel are tested. Moreover, for the MLP, three types of neuron types are tested: logistic, tanh, and rectified linear unit (Relu).

4.3.4.2. Leave-one-out cross-validation

The leave-one-out cross-validation method is a standard method to evaluate the performance of the classification model. Here the 5-fold leave-one-out cross-validation method is used to analyze data collected from the ceiling-mounted sensor node only to

compare the performance of different classifiers using the proposed HOG+PCA-based feature extraction method.

The whole data set is separated into 5 subsets, and the accuracy can be calculated by using one of the subsets as the testing dataset and the other four subsets as the training dataset. This process is repeated five times to get an average accuracy. Here the principal component number is selected as 300, and detailed analysis is included in sub-section B to get the best number.

Table 4.3 summarizes the classification accuracies and the training time based on 5-fold leave-one-out cross-validation. For hand-crafted feature extraction, the RF classifier reaches the highest accuracy of 96.3%, and all the MLP classifiers also have high accuracies (> 96.0%). In addition, we demonstrate that the hand-crafted feature selection (90.0%) does not work well with the LR, the SVM, or the GBDT classifier. However, by using the HOG+PCA-based feature extraction approach, most classifiers reach high accuracies (> 99.0%). In particular, the MLP with the Relu unit has the highest accuracy of 99.8%, while only DT and GBDT have a relatively poorer accuracy, as shown in Table 4.3.

One can conclude that the overall performance of HOG+PCA-based feature extraction is better than that of hand-crafted features, and both feature extraction methods can reach high accuracies above 96.0%. This means both methods are suitable for personalized detection of in-bed posture.

Table 4.3 Cross-Validation Accuracy. © 2021 IEEE. Reprinted with permission from [27].

Classifier	Hand-crafted [%]	Training time[s]	HOG+ PCA [%]	Training time[s]
LR	69.9	0.7	99.6	3.7
DT	86.3	0.4	92.1	1.1
RF	**96.3**	5.5	99.4	7.2
SVM (linear)	74.8	24.1	99.7	7.7
SVM(RBF)	68.7	51.4	99.5	17.8
GBDT	85.2	35.5	97.3	61.3
MLP(Logistic)	**96.9**	6.7	99.4	0.9
MLP (tanh)	**96.0**	8.7	99.6	0.6
MLP(Relu)	**96.2**	6.8	99.8	0.9

4.3.4.3. Leave-one-subject-out cross-validation

For 5-fold leave-one-out cross-validation, both the training set and the testing set contain data from all the users. This helps achieve a higher accuracy as each frame of data in the testing dataset may find a similar data frame in the training dataset. However, this may not be very practical in real practice as it is challenging to get a training dataset from every new user. The leave-one-subject-out cross-validation, also known as the cross-user-validation, becomes a helpful method to evaluate the robustness of the classification model. [107] In other words, data from one of the 12 users serve as the testing set, and the other data are used as the training dataset. This step is repeated 12 times by setting each user's data as the testing dataset. The average value is calculated as the accuracy of leave-one-subject-out cross-validation. Instead of just evaluating the performance of classifying 9 postures, the dataset is also relabeled as 4 posture types: Front, Left, Right, and Back since they are the most commonly used posture category for in-bed posture detection. More specifically, 'Front' contains soldier and starfish, 'Left' contains left log, left fetus, and

left yearner, ‘Right’ contains right log, right fetus, and yearner, and ‘Back’ only contains the freefall posture. This 4-posture dataset is down-sampled again to keep balanced data size within each posture. Furthermore, to reduce the computational cost, the number of PCA components for the cross-user-validation is set as 30.

Table 4.4 9-posture Cross-user-validation accuracy. © 2021 IEEE. Reprinted with permission from [27].

Classifier	Ceiling-mounted [%]	Wall-mounted [%]
LR	83.6 ± 9.5	73.3 ± 12.8
RF	82.3 ± 9.7	71.4 ± 13.3
SVM (linear)	84.9 ±10.3	70.7 ± 11.2
SVM(RBF)	**86.0 ± 8.8**	**76.3 ± 12.4**
MLP (logistic)	85.3 ± 8.3	71.1 ± 12.2
MLP (tanh)	85.0 ± 9.8	70.2 ± 11.2
MLP(Relu)	**86.2 ± 8.7**	71.3 ± 12.4

In addition, the detection accuracy based on hand-crafted feature selection using cross-user-validation is not listed in detail. Because the best result (less than 70.0%) is vastly lower than that using HOG+PCA-based feature extraction. So only the cross-user-validation result based on HOG+PCA-based feature extraction is analyzed for both mounting types.

Table 4.4 summarizes the classification result of using 9-posture cross-user-validation. Overall, the ceiling-mounted approach results in about 10.0% higher accuracy than that of the wall-mounted approach using the same data processing method. For the ceiling-mounted sensor node, SVM and MLP classifiers have relatively better performance in which the MLP with Relu neural unit the highest accuracy of 86.1% with 8.8% deviation.

Table 4.5 4-Posture Cross-User-Validation Accuracy. © 2021 IEEE. Reprinted with permission from [27].

Classifier	Ceiling-mounted [%]	Wall-mounted [%]
LR	88.0 ± 9.2	73.7 ± 8.8
RF	84.4 ±11.7	72.5 ±10.1
SVM (linear)	88.5 ± 9.8	71.3 ± 8.4
SVM(RBF)	89.0 ± 9.8	80.4 ± 8.2
MLP (logistic)	87.4 ± 9.4	75.7 ± 9.0
MLP (tanh)	88.2 ± 9.3	73.8 ± 8.1
MLP(Relu)	88.7 ± 9.4	77.8 ± 8.5

Table 4.5 summarizes the result of 4-posture cross-user-validation accuracy. Both the SVM and MLP classifiers still have better performance than the other classifiers, while the SVM classifier with RBF kernel achieves the highest accuracy of 89.0% with ±9.8% deviation. Compared to the wall-mounted sensor node, the ceiling-mounted sensor node increases the accuracy by at least 8.6%. From Table 6, we can find that both the SVM and MLP classifiers return the best results. This also matches with the result of 5-fold leave-one-out cross-validation. The MLP classifier is a NN that contains hundreds of neurons, so it tends to have stronger fitting abilities when working as a classifier. Also, the outperforming SVM classifier agrees with the research observation by the computer vision community that the SVM can achieve better performance than other classifiers when using HOG features [83].

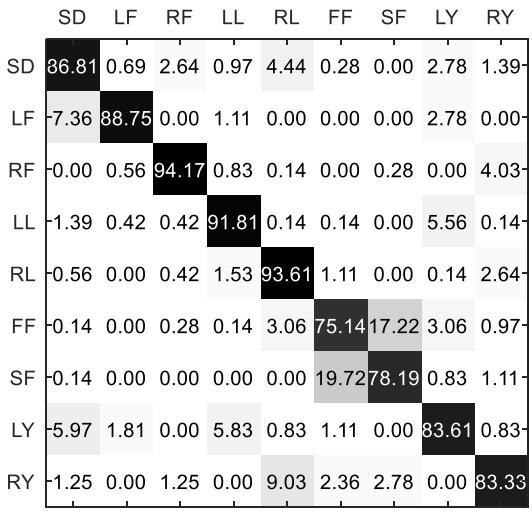


Figure 4.19 Confusion matrix for 9 class cross-user validation using MLP(Relu) classifier and top sensor data. © 2021 IEEE. Reprinted with permission from [27].

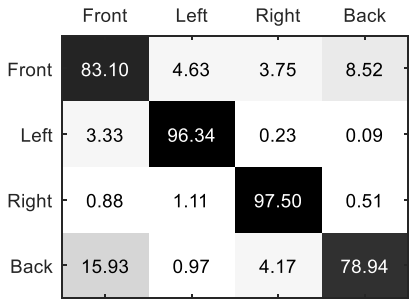


Figure 4.20 Confusion matrix for 4 class cross-user validation using SVM（RBF) classifier and top sensor data. © 2021 IEEE. Reprinted with permission from [27].

Figure 4.19 shows the confusion matrix based on the cross-user- validation for the 9-posture classification using the MLP(Relu) classifier. The values in the confusion matrix are calculated based on the average value of accuracy in the cross-user-validation. The 'Freefall' and 'starfish' postures have relatively poorer performance compared to other postures. 19.7% of the 'starfish' postures are wrongly detected as 'Freefall'. 17.2% of the

‘freefall’ postures are wrongly detected as the ‘starfish’. A possible reason is that ‘starfish’ is a horizontal flip of ‘freefall’, and they have similar shape profiles. One significant difference between these two postures is that ‘starfish’ should have a much higher temperature around the head area than the ‘freefall’ since ‘freefall’ is a face-down posture. This can be used as a uniquely effective feature to differentiate the ‘freefall’ from the ‘starfish’ posture, while reliably identifying the head location during a low-pixel thermal image poses another challenge.

Figure 4.19 shows the confusion matrix for the 4-posture classification using the MLP classifier with Relu unit and the SVM classifier with linear kernel, respectively. The confusion matrix shows that the left and right postures have relatively higher accuracies (> 96.0%) than those of the other postures. Moreover, the front and back postures are more likely to be wrongly identified.

4.3.4.4. Evaluation of different preprocessing methods

The preprocessing could be a crucial step for further improving the final classification accuracy. There are two significant steps in the preprocessing step: translation and rotation (RT) and temperature scaling (TS). Table 4.6 compares the performance of different combinations of preprocessing and feature extraction methods. For each combination, the highest accuracy achieved from cross-user-validation among all classifiers is used for further comparison.

We can find that the RT and the TS can largely improve the robustness of the classification model when using the HOG+PCA-based feature extraction. By applying both the RT and TS, the cross-user-validation accuracy of 4 postures is increased from

60.7% to 88.97%, and the cross-user-validation accuracy of 9 postures is increased from 58.3% to 86.2%

Table 4.6 Accuracy using Different Preprocessing Methods. © 2021 IEEE. Reprinted with permission from [27].

RT	TS	HOG	4-posture [%]	9-posture [%]
●		●	81.3	75.5
	●	●	60.7	58.3
		●	67.7	61.8
●	●	●	89.0	86.2

Table 4.7 compares the result in this work with other approaches. Our proposed model achieved a significantly high accuracy (99.5%) for leave one out cross-validation. Therefore, here we only compare the work with cross-user validation. We can see that our proposed model is still about 8.9% lower compared to 97.9% accuracy from the state-of-art method using pressure mat. However, a bed with a pressure sensing mat is very costly (>$5,000) and requires frequent maintenance, making it unsuitable for widely home and long-term use. The proposed thermopile array sensor-based approach provides a low-cost alternative (less than $100). Furthermore, the thermopile array sensors have many other advantages, such as non-contact passive sensing, non-privacy invasion, plug and play, and low power consumption. This makes them more practical for long-term application.

Table 4.7 Comparison with the state of the art

Sensor type	**No. pixel**	**No. of Pos.**	**Evaluation (cross-user)**	**Acc.[%]**	**Ref.**
Pressure mat	64×27	4	Training: 7, Testing: 5	97.9	[108]
	128×64	6	Training: 13, Testing: 1	83.0, precision 83.2, recall	[106]
	128×64	6	Training: 13, Testing: 1	91.2	[109]
Thermopile array	32×24	4	Training: 11, Testing: 1	89.0	This work
		9		86.1	

4.4. Conclusion

This chapter proposes traditional machine learning methods such as handcrafted and HOG feature extraction for falling and in-bed posture detection.

A handcrafted feature extraction method is proposed for fall detection, which relies on an 8x8 pixel Grid-EYE sensor and a distance sensor. Experimental studies include three non-fall activities - standing, sitting, and stooping, and two fall actions – forward falling and sideway falling to simulate elderly daily activities. Different feature sets for the SVM-based machine learning algorithm are analyzed, and their impact on fall detection accuracy is evaluated and compared empirically. Overall accuracy above 90% is achieved.

An in-bed posture detection method has been developed. The preprocessing method and feature extraction approach based on the combination of the histogram of oriented gradient and the principal component analysis (HOG+PCA) is compared to the traditional hand-crafted feature classification. The user study shows that a total of 9 in-bed postures can be successfully classified with 5-fold cross-validation-accuracy over 99.8%. The cross-user-validation is also performed to evaluate the robustness of posture classification models. The 4-posture classification achieves an accuracy of 89.0% when using the SVM classifier, and the 9-posture classification achieves an accuracy of 86.2% when using the NN classifier based on 30 principal components features. On the other hand, the cross-user-validation using hand-crafted feature extraction is not satisfied (<70.0%), which indicates that they are only suitable for the personalized model.

The results show the traditional machine learning methods can reach high accuracy for falling and posture detection, but a unique processing method is required for different

objectives. Moreover, the performance is sensitive to the feature extraction step. For example, HOG needs to work with specific data preprocessing to extract reliable features.

5. FACING DIRECTION DETECTION AND IDENTIFICATION USING THERMOPILE ARRAY SENSOR AND DEEP LEARNING*

5.1. Review of Deep Learning Methods using Thermopile Array Sensor

Some deep learning models have recently also been applied to thermopile array sensor data. For example, three Grid-EYE sensors are installed on the two walls and the ceiling of a gym room to detect 26 different yoga postures[66]. Moreover, a 99.8% accuracy is reached using a customized CNN model with three convolutional layers. In addition, a deep convolutional encoder-decoder model reaches 98.43% accuracy when applying to the MLX90640 sensor data for occupancy counting with adaptive placement [30].

5.2. Case Study: Pre-Trained CNN for Facing Direction Detection

5.2.1. Experiment Setup

The experiment is performed in a specified laboratory area. Each testing subject/user sits on a chair that can be easily rotated. Type A thermopile sensor node is set in front of the user at the height of 0.75m. Five facing directions are tested in our experiment: Left 45º/90º, Right 45º/90º, and Front. The top view of the experimental setup schematic and its snapshot are shown in Figure 5.1(a) and (b), respectively, in which the user facing direction is 'Right' 45º. The snapshots of the other four tested directions are shown in Figure 5.1(c). Three detection distances are tested: 0.6m, 1.2m, and 1.8m. For

---

all these three detection distances, the facing direction keeps the same as the torso direction. Totally five users participated in this experiment.

The training datasets and background features are always captured beforehand. Data images of each facing direction from each subject are saved for at least five seconds, and only the first 50 images of each facing direction from each subject are used in the training dataset to make a fair comparison. That is, the training dataset includes a total of 250 images from each subject.

To record testing datasets, a facing direction instruction video is played to the subjects simultaneously to provide ground truth. The instruction in the video is designed as follows: each subject rotates continuously seven times while keeping at each facing direction for four seconds before rotating to the next direction. The order of designed directions is 'Front', 'Left 45º', 'Left 90º', 'Left 45º', 'Front', 'Right 45º', 'Right 90º' and 'Front'. For the first six rotations, the subject only needs to rotate 45º. During the last rotation, the user rotates directly from 'Right 90º' to the 'Front' position.

Based on this design, the testing datasets should include 320 images (32 seconds). However, the actual recorded testing datasets have about 350 images because the data recording always starts before playing the instruction video and ends after the end of the instruction video. That is, the first 'Front' and the last 'Front' contain data longer than 4 seconds. And thus, in the ideal case, the ground truth reference needs to be modified to be consistent with the testing time frame, which is difficult to achieve, as the rotation speed of each user is different. In this experiment, we directly extend the ground truth datasets from 320 to 350 by aligning the first frame of 'Left 45º' and adding 'Front' to the

beginning and the end, so it has consistent time frames with the testing datasets. This may still cause inaccuracy in the detection because there is always a delay before the user starts rotating to the specific direction after the user reacts to the instruction video. This delay varies between different users.

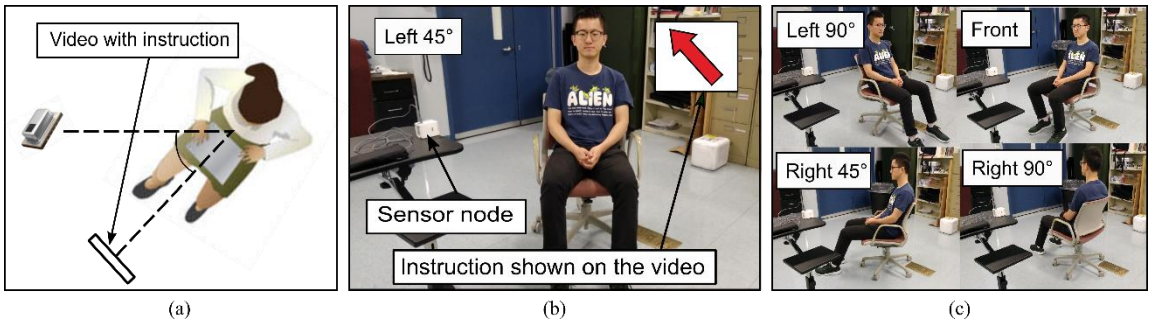


Figure 5.1 (a) Systematic of testing experiment top view (b) Experiment setup with facing direction 'Right 45º' (c) Snapshots of experiment setup. © 2018 IEEE. Reprinted with permission from [67].

5.2.2. Data Preprocessing

This section introduces the preprocessing algorithms to convert a raw 8×8 temperature image to a 32×32 binary matrix. Furthermore, the step-by-step data visualization is shown in fig. 5. More specifically, Figure 5.2(a) and (b) show the photographic representation and raw data received by the Grid-EYE sensor. The first step of preprocessing (Figure 5.2 (c)) is to subtract the background temperature noises of other non-human heat sources (i.e., wall light, computer, HVAC vents etc.) from the raw data. The purpose is to identify and focus the subject in the FOV. Then the 8×8 matrix is converted to the 32×32 matrix using bilinear interpolation, as shown in Figure 5.2 (d). In

the end, binarization with adaptive threshold values is implemented to highlight the main body of the subject in the image.

To obtain the threshold value, the maximum temperature difference of every column of the 32 x 32 matrix is first sorted and among which the minimum temperature difference is taken as the threshold for binarization. The final effect after binarization is shown in Figure 5.2 (e). Binarization can efficiently reduce redundancy information in the temperature image. The features will then be extracted based on the image with binary data. After preprocessing, two feature extraction methods are used to compare their detection accuracy.

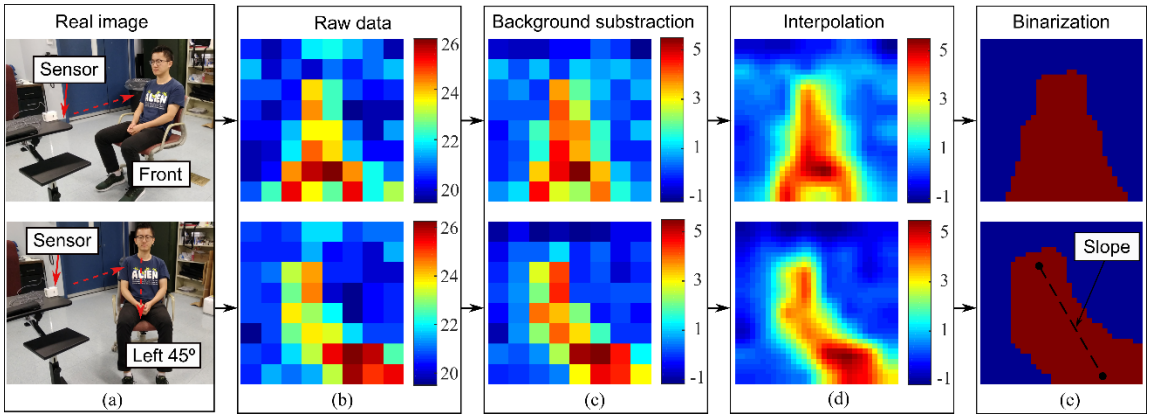


Figure 5.2 (a) Optical image (b)Raw data from the Grid-EYE sensor (c) Data after background subtraction; (d) Linearly interpolate image to a 32×32 matrix;(e) image after Binarization with adaptive threshold values. © 2018 IEEE. Reprinted with permission from [67].

5.2.3. Pre-trained CNN

This section introduces the CNN-based feature extraction method. In the process of a CNN layer, the input is a certain area size (width × height) of the data, and the output is the inner product of the input and multiple layers of the filter (with a fixed weight of the

neurons). As shown in Fig. 4, the red rectangle stands for a filter, and the 8x8 group of yellow circles represents an image. A convolutional operation is performed on the input data based on a filter with weights w_1, w_2, w_3, w_4, and bias b_0, determined by the pretrained model. The blue circle represents the output of the convolution, which is a character of the area in the red rectangle. After calculating the local data within a window, the data window continues to slide until all data has been calculated. This is considered as a single CNN layer. And the green circles represent the character matrix after the original data is processed by several layers of CNN.

In this experiment, the CNN model [110]. consisting of a seven-layer structure (fig. 5) is used for recognizing images of characters. The first layer is a convolutional layer with no padding, the output of a 28×28×20 matrix. And the input is a 32×32×1 image, that is why the raw 8×8 image of Grid-EYE sensor is first converted to a 32×32 binary image (fig.5). The second layer is a max-pool layer. The third layer is a convolutional layer. The fourth layer is another max-pool layer. The fifth layer is a convolutional layer, and the sixth layer is a Rectified Linear Unit (ReLU) layer. The first six layers can produce a matrix of 2×2×500 features, a characteristic of deep learning that it constructs. The last layer is a fully connected lay that gives the possibility of the 26 categories (letters). Noting that the original CNN model was not trained based on thermal image, it cannot work automatically as a classifier, so the first six layers are used as the feature extractor and it was integrated with the SVM classifier. The produced matrix of 2×2×500 features of each thermal image is rearranged to a 1×2000 and saved as a feature set with 2000 dimensions.

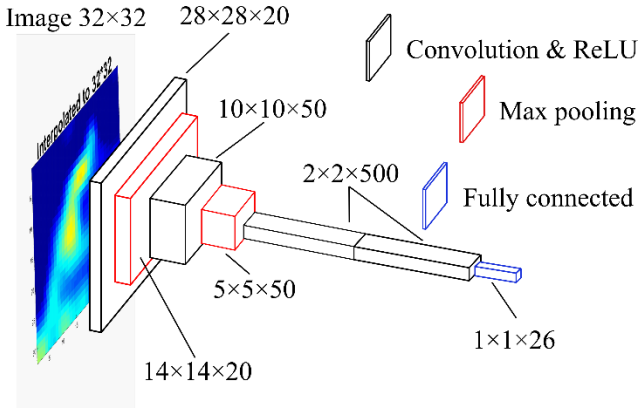


Figure 5.3 The structure of the CNN model used in this paper. © 2018 IEEE. Reprinted with permission from [67].

5.2.4. Handcrafted Features (Baseline)

Three Handcrafted features are extracted from the binarized image (fig.3(e)) to be used for SVM classification. Each binary image consists of 32 by 32 pixels. Pixels of 1 are considered as high temperature pixels. Pixels of 0 are considered as low temperature pixels. Each binary image is divided into four parts evenly, named the top left, top right, bottom left, and bottom right.

Feature 1: Slope of the high temperature pixels. The middle point of high temperature pixels of each row is defined first. The slope is then calculated by connecting the middle point with the maximum and minimum horizontal coordinate, as shown in fig. 5(e).

Feature 2: Area difference between the bottom two parts of the thermal image. That is the number of pixel differences between the bottom left and the bottom right.

Feature 3: Area difference between the top two parts of the thermal image. That is the number of pixel differences between the bottom left and the bottom right. Algorithm 1 shows how the three features are calculated in detail.

5.2.5. Experimental Results

Since three features are included in the handcrafted feature extraction method, the performance of different feature combinations is compared. The results are shown in Figure 5.4. The detection accuracy by using most feature combinations is above 80.0% except the feature set (2,3) at a distance of 0.6m and feature set (1,3) at 1.8m. The average accuracy at three detection distances for feature set (1,2), (1,3), (2,3) and (1,2,3) are 81.7%, 79.3% 80.2% and 82.5%, respectively. Feature set (1,2,3) has the highest accuracy at 0.6m and 1.8m, and feature set (1,2) has the highest accuracy at 1.8m. We consider (1,2,3) to be the best feature set as it has the highest average accuracy even though it does not maintain the highest accuracy at all three detection distances.

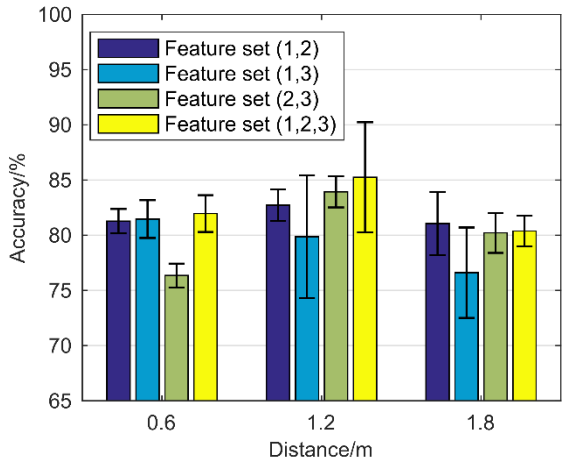


Figure 5.4 Feature set analysis on handcrafted features. © 2018 IEEE. Reprinted with permission from [67].

Figure 5.5 compares the detection results using two feature extraction methods at three distances: (a) 0.6m (b) 1.2m (c) 1.8m. Results on the left panel are based on handcrafted feature set (1,2,3). Results on the right panel are based on feature extracted

by CNN. Experimental results show that by adding a median filter with a window size of three, the detection accuracy can be further improved. The results after adding the median filter are shown in Figure 5.6.

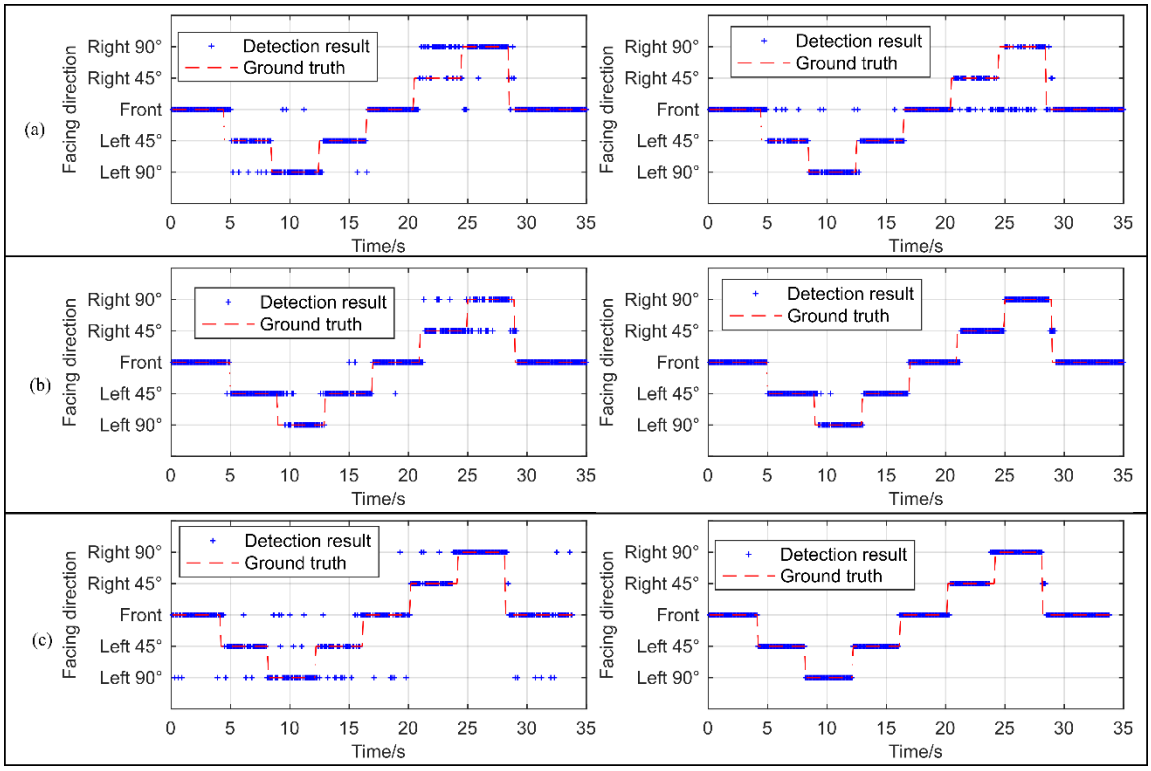


Figure 5.5 Facing direction detection results at distance of (a) 0.6m (b) 1.2m (c) 1.8m. The left panel shows the detection results based on manually-define features and the SVM classifier. The right panel shows the detection results based on the CNN feature extraction and the SVM classifier. © 2018 IEEE. Reprinted with permission from [67].

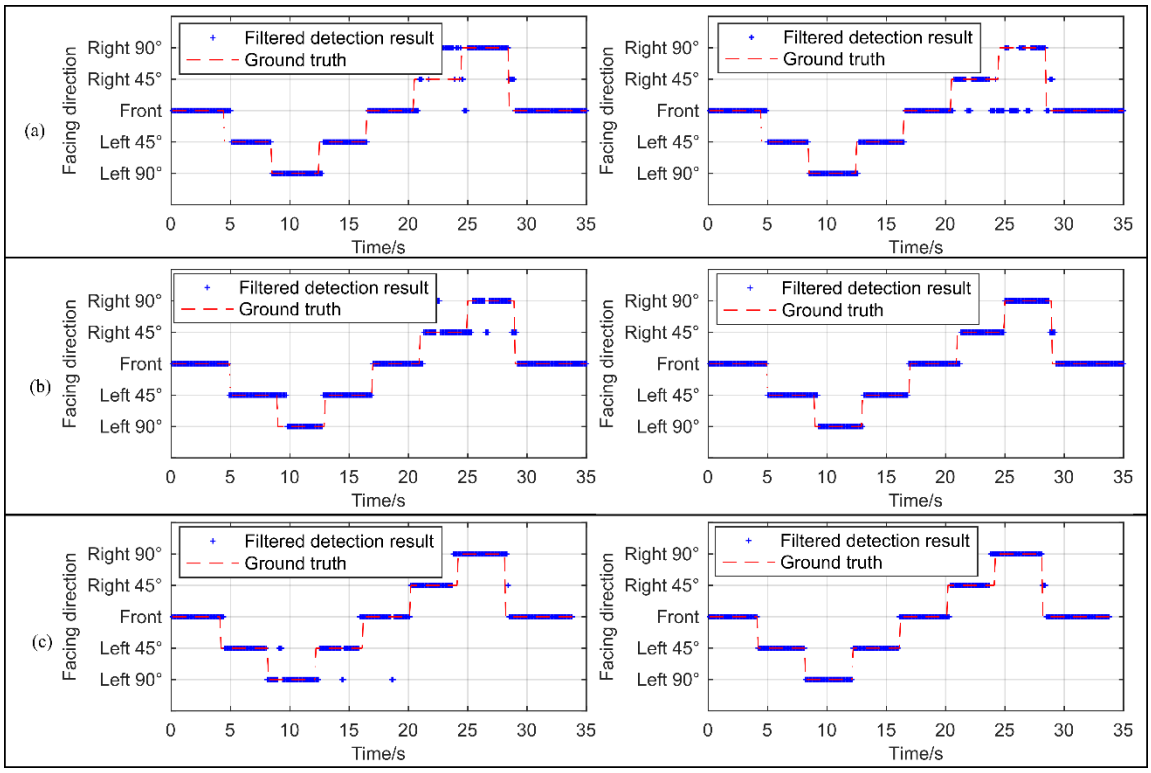


Figure 5.6 Facing direction detection results after adding a median filter at distance of (a) 0.6m (b) 1.2m (c) 1.8m. The left panel shows the detection results based on handcrafted features and the SVM classifier. The right panel shows the detection results based on the CNN feature extraction and the SVM classifier. © 2018 IEEE. Reprinted with permission from [67].

Figure 5.7 summarizes the accuracy of facing direction detection without post-processing. Compared to handcrafted feature extraction, the average detection accuracy of using CNN-based feature extraction is improved by 3.4%, 8.2%, and 12.9%, respectively, at a distance of 0.6m, 1.2m, and 1.8m. The post-processing filtering does not lead to an obvious accuracy improvement of using CNN-based feature extraction, and the average accuracy improvement is only 2.4%. A possible reason is that the accuracy without filtering is already high (86.2%, 93.6%, 93.4% at a distance of 0.6m, 1.2m, 1.8m, respectively) and relatively close to 100%. However, the extra filtering results in a much

more obvious increase for handcrafted feature extraction: an increase of 4.2% (81.2% to 85.4%), 5.4% (85.2% to 90.6%) and 5.2% (80.4% to 85.6%) at distance of 0.6m, 1.2m and 1.8m, respectively. Furthermore, when the distance increases from 1.2m to 1.8m, the accuracy using ma handcrafted feature extraction has an obvious decrease (about -5.3%), while detection accuracy using CNN-based feature extraction remains a high value > 95.0%. In summary, the CNN-based feature extraction detection method remains a high and reliable accuracy method compared to the handcrafted feature extraction method, regardless of its detection distances. A median filter can increase the detection accuracy, particularly for detection based on the handcrafted feature extraction method at a further distance.

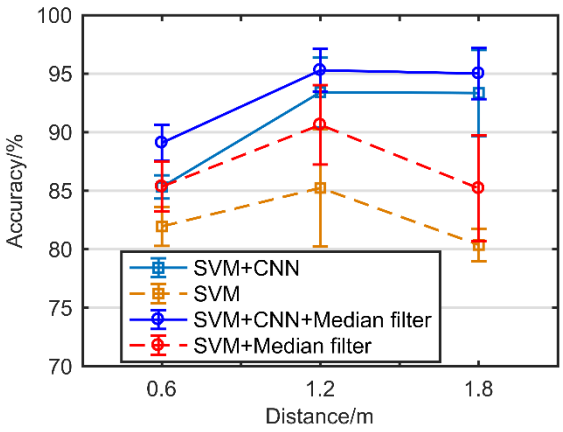


Figure 5.7 Facing direction detection accuracy comparison. CNN based feature extraction + SVM classifier vs handcrafted feature extraction + SVM classifier vs CNN based feature extraction + SVM classifier + median filter vs handcrafted features + SVM classifier + median filter. © 2018 IEEE. Reprinted with permission from [67].

5.3. Case Study: Customized CNN for Occupancy Identification

5.3.1. Experiment Setup

The Type B thermopile array node is installed on the ceiling near a lab entrance for occupancy identification, as shown in **Figure 5.8**. There are two fixed marking points on the ground (A and B). Each testing subject follows the same routine to walk in and out of the room during the experiment (yellow line in the subplotting figure). By doing this, the effect of the walking trajectory on the identification performance can be avoided. For each testing subject, they are asked to walk in and out 40 times each. The data saving is manually controlled. Every time, data saving starts first, and then the subject begins to walk. The data saving ends when she/he arrives at the target point (B for 'walk in' and A for 'walk out'). Eight testing subjects (two female and six male) participated in the experiments. Their height and weight information are listed in **Table 5.1**.

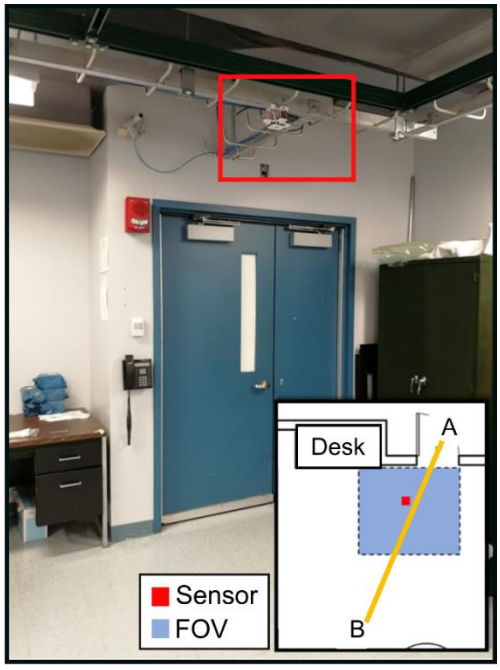


Figure 5.8 Experiment setup of occupancy identification

Table 5.1 Personal information of the testing subjects

User	Gender	Height[cm]	Weight[kg]
1	M	177	68
2	M	170	70
3	M	172	71
4	M	168	65
5	F	165	47
6	F	163	48
7	M	180	75
8	M	182	80

Thus, we have a total of 40(times)×2(direction) ×8(subjects) = 640 instances of data. Here, an 'instance' of data contains all sensor data from when the subject enters and ends when she/he leaves through the entrance/ doorway.

5.3.2. Customized CNN Model

Figure 5.9 shows the systematic of the identification algorithm. The preprocessing is performed first to fuse the data from two types of sensors and generate a modified feature image which is used as the input to the CNN model

(1) Distance sensor data filtering and normalization

Data from the distance sensor requires a denoising and normalization process. The original data from distance sensors only have valid values between 10 to 2000 with unit

mm. First, filtering is performed to remain only the valid data, and then the normalization is performed to convert the data range to (20,35) using the following equation.

$$D_n = \frac{(2000 - D)}{3} + 20$$

Here D is the original measurement data of the range sensor, and D_n is the value after normalization. This step converts the data from the distance sensors to a similar range as that collected from the Grid-EYE sensor. This helps the model training achieve high accuracy.

(2) Time segmentation

The CNN model requires an input with a fixed dimension. However, the dimensions of the different data instances are different. Thus, time segmentation needs to be determined before data can be fed into the CNN model. Here, it is achieved by finding the center time index and selecting a preset time length L. The center time index is determined by checking the range of valid distance sensor data, and the middle time index is considered the center index. For example, if the first valid range sensor data has an index T_1, and the last valid range sensor data has an index T_2, the center time index will be $(T_1 + T_2)/2$, as shown in Figure 5.9 (a). A constant value of 60(frames) is selected as the time length L, equivalent to data of 2.5s since its sampling frequency is 25Hz. By manually checking the plotting of Grid-EYE sensor data (Figure 5.9(b)), this time length is sufficiently long to include the data from the whole period where the occupant is within the sensor FOV.

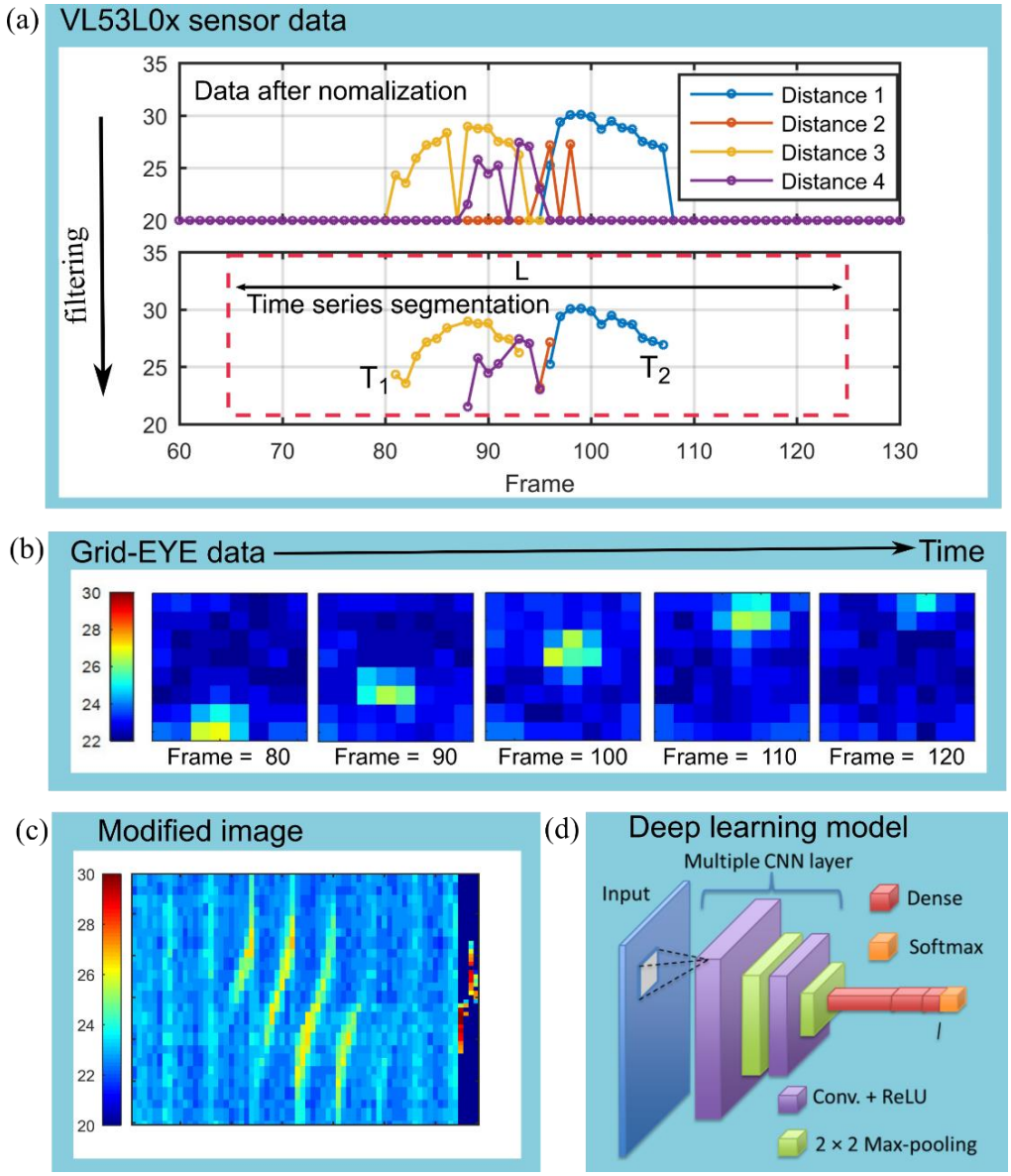


Figure 5.9 Systematic of data processing pipeline: (a) data preprocessing of VL53L0X (b) Grid-EYE data matching (c) modified feature image (d) CNN based classification model

(3) Reshape the sensor data.

Since the sampling frequency of the two sensors is different, we need to match the thermal data from Grid-EYE (10Hz) with the data from VL53L0 distance sensors(25Hz) based on the time index. In other words, each timestamp corresponds to 1×4 data from the distance sensors and 8×8 data from the infrared array sensor. Figure 5.9(b) shows the plotting of Grid-EYE data at frame number 80 – 120, and we can see that the high-temperature area moves from the left bottom of the figure to the right top, which indicates the movement of the testing subject under the sensor node.

After that, data from the two types of sensors are reshaped to a 1×68 matrix. Next, data from the segmented period are combined to generate a modified data matrix. Each line of this data matrix is the sensor data from a timestamp. Finally, a 60×68 feature image is generated to be the CNN model's input, as shown in Figure 5.9(c).

(4) CNN model

A CNN model can automatically extract features from the feature image and complete the classification, as shown in Figure 5.9(d). However, the CNN model contains multiple convolutional layers and fully connected (FC) layers. Therefore, the structure of the CNN model needs to be designed and optimized to get both high accuracy and low computing cost. The CNN model's performance with different hyperparameters is compared, including the number of convolutional layers, FC layers, and filter size.

5.3.3. Experimental Results

Two steps are applied to achieve and evaluate the performance of the CNN model. The first setup aims to determine the hyperparameters of the CNN model, such as the

number of layers and filters. To do so, we separate the whole dataset into two parts, the training data, and validation data. Moreover, they are fixed for every set of hyperparameters. To be more specific, the first 60 instances of each user are used as the training data, and the rest 20 instances are used as the validation dataset. Therefore, the training dataset has 480 instances of data, and the validation dataset has 160 instances.

During the second step, the four-fold cross-validation is used to evaluate the performance of the identification model with a specific structure identified in the first step. The total dataset is randomly separated into four folds; each folder contains 160 instances of data. Three-fold (480 instances of data) is used as training, and one-fold (160 instances of data) is used as testing. This process can be repeated four times by using different folds as the testing dataset. Finally, the average accuracy is reported as the model accuracy.

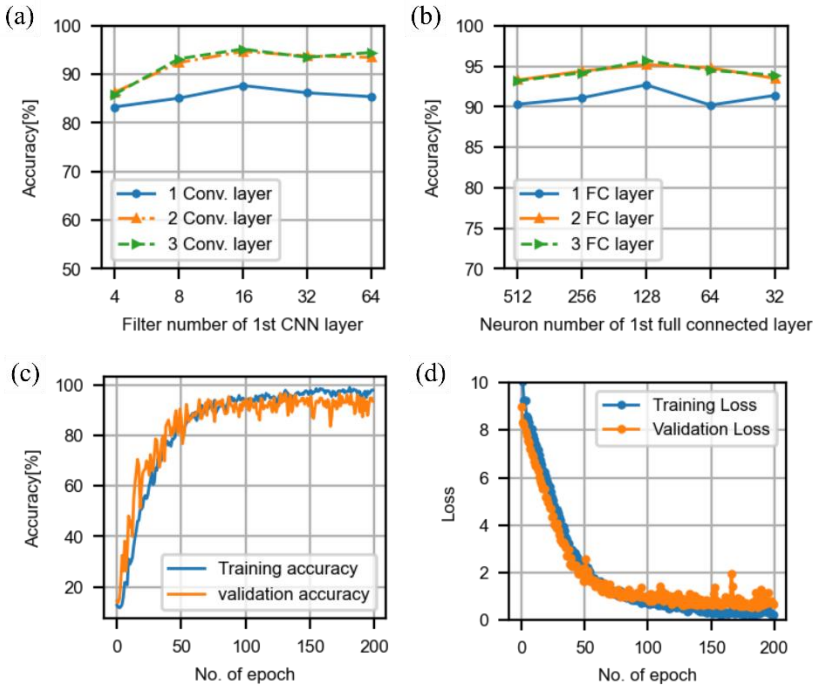


Figure 5.10 Hyperparameter tuning of the deep learning model for occupancy identification

Figure 5.10(a) shows the accuracy comparison of the model with different numbers of convolutional layers and filters. The CNN model with one, two, and three convolutional layers are included, and for each convolutional layer, we tune the filter's number. The X-axis listed the filter number of the first convolutional layer. For the other layers, the number of the filter doubles that of its previous layer. For example, if the first layer has 16 filters for a two-layer CNN model, the second layer will have 32 filters. In addition, the fully connected layers are fixed as two layers with 256 and 128 neurons. We can find that the accuracy of the CNN model with two layers or three layers is significantly higher than that of the model with one layer.

In comparison, the accuracy of the three-layer model is similar to that of the two-layer model. In addition, for the two-layer model, we can find that the accuracy increases when the filter size of the first layer increases from 4 to 16, and then it slightly decreases. Thus, we choose a CNN model with two convolutional layers in this work. After determining the hyperparameters of convolutional layers, we compare the model with different FC layers. Figure 5.10 (b) shows that 2 FC layers with 128 and 64 neurons reach the highest accuracy.

Table 5.2 shows the details of selected hyperparameters, which achieve the highest accuracy. It contains two convolutional layers with 3×3 filters and three fully connected layers. In addition, there is a max-pooling layer after each convolutional layer. Figure 5.10 (c) and (d) shows the accuracy and loss vs. epoch number during training using the aforementioned CNN structure, respectively. We found that the best configuration of these

hyperparameters can reach a training accuracy of 98.1% and a validation accuracy of 95.0%.

Table 5.2 Selected hyperparameters

Layer	Hyperparameter	Value
Input	Input image size	68 × 60
1st convolutional	Filter size	3 × 3
	Filter number	8
Max-pooling	Filter size	2 × 2
2nd convolutional	Filter number	16
	Filter size	3 × 3
Max-pooling	Filter size	2 × 2
1st Full connected	No. of neurons	128
2nd Fully connected	No. of neurons	64
3rd Fully connected	No. of neurons	8

After the hyperparameter of the model is determined, 4-fold cross-validation is performed to analyze the model performance further. Figure 5.11 shows the confusion matrix of the cross-validation result with 8 users. The accuracy is higher than 90% for all users.

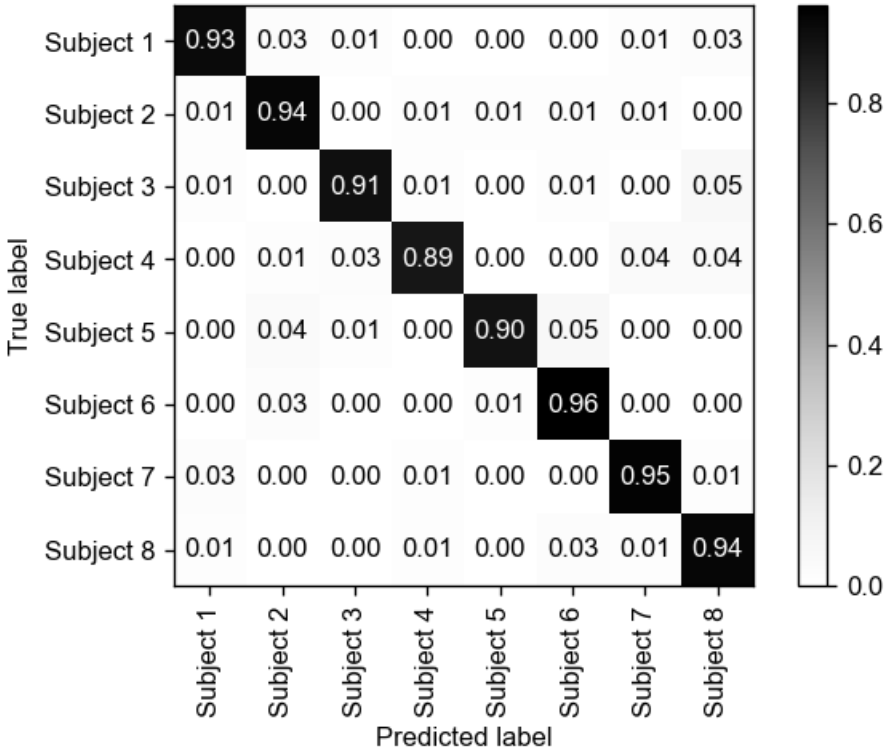


Figure 5.11 Confusion matrix of occupancy identification

Table 5.3 shows the comparison between state-of-the-art approaches and the result in this work. The proposed work has lower power consumption and higher accuracy than most other methods. The accuracy is only slightly lower than the approach using three ultrasonic sensors on the door frame. However, their power consumption is much higher, and the ultrasonic signal is not animal friendly.

Table 5.3 Comparison with the state of the art

Category	Sensing approach	No. of subjects	Acc. [%]	FOV	Power [mW]	Ref.
Body shape ultrasonic distance (doorframe)	1 on top	4	89.7	Doorway	75	[111]
	2 on top	5	90.1	Doorway	150	[112]
	1 on top, 2 on side	20	95.3	Doorway	225	[113]
RF, Gait signature	UWB (3.1 4.8GHz)	8	88.2	Doorway	116	[114]
		15	80.9	3m to user	116	[48]
	Wi-Fi	6	88.3	3m (2 access points)	200	[115]
		4	89.9		200	[116]
On-object sensing	5 Accelerometers on objects, kitchen	5	96.1	N/A	<5	[117]
Thermal signature + body shape	1 Grid-EYE(8×8)+ 4 distance sensors	8	94.4	Doorway 3m x 3m	100	**This work**

5.4. Conclusions

This chapter proposes deep learning methods such as pre-trained CNN and customized CNN models for facing direction detection and occupancy identification.

A pre-trained CNN model is applied as a feature extractor for facing direction detection. The experiments are performed at three distances, 0.6m, 1.2m, and 1.8m. The

CNN-based feature extraction demonstrates more reliable performance for facing direction detection compared to the handcrafted features regardless of the detection ranges. For example, the average accuracy reaches 89.1%, 95.3%, and 95.1% at the distance of 0.6m, 1.2m, and 1.8m, respectively.

A customized CNN model is proposed to fuse the data from two types of sensors for occupancy identification. In addition, a non-intrusive sensor node is prototyped based on a thermopile array and four distance sensors. The sensor node is installed near the lab entrance as the realistic setup to get the dataset, and 94.4% accuracy is achieved to identify eight different people. The proposed approach also shows a higher accuracy and lower power consumption compared to the state-of-the-art.

Deep learning methods can automatically extract features from the raw sensor output without applying specific feature extraction methods. Moreover, they are more robust than the handcrafted features.

6. CONCLUSIONS

6.1. Contribution

Two types of infrared sensing technologies, SLEEPIR, and thermopile array have been applied for fine-grained occupancy sensing objectives such as presence detection, posture detection, facing direction detection, and occupancy identification. In addition, data processing models based on handcrafted features, traditional machine learning, and deep learning are proposed.

In Chapter 2, a mathematical model is proposed to predict SLEEPIR output for stationary occupancy detection under changing ambient temperature. The unoccupied $V_{pp,u}$ is found to be proportional to the temperature difference of sensor and background $V_{pp,u} = K(T_{floor}^4 - T_{sensor}^4)$. The linearity is validated, and the coefficient K_{BB} is characterized using a blackbody radiation source. The LC shutters' performance for stationary occupancy detection is evaluated by performing the occupant experiments to obtain ΔV_{pp} at different distances. The sensor noise is analyzed, and the LC shutters can be classified as 'Good','Fair', and 'Poor' based on metric ΔV_{pp}@1.5m. Moreover, K_{BB} is highly correlated with the ΔVpp@1.5m value, so it can help reduce the time of LC shutter classification vastly. Finally, SLEEPIR sensor nodes are prepared with 'Good' LC shutters to enable the maximum FOV of stationary occupancy detection.

In Chapter 3, datasets for occupancy presence detection using SLEEPIR sensor node are collected in a lab room and a room in a residential apartment. In addition, two edge cases experiments are also performed, including changing the room temperature and lying on a bed. Finally, different occupancy presence detection algorithms are proposed

and compared. The LSTM is a deep learning approach that uses time-series features extracted from the sequence of SLEEPIR sensor samples. The feature includes V_{pp} from SLEEPIR modules, temperature reading, and the output of the digital PIR sensor. Stat. LSTM is another deep learning model using statistical features instead of only V_{pp} values, and it does not need the data from a digital PIR sensor.

The different occupancy presence detection algorithms are compared by reporting their daily occupancy detection accuracy. The result shows that both LSTM and Stat. LSTM reaches reliable high accuracy >96.0% for both the lab room and residential apartment. Moreover, the Stat. LSTM model does not require the data from a digital PIR sensor. Furthermore, the result of the edge case experiment indicates that the proposed model is a reliable occupancy presence detection method.

In Chapter 4, handcrafted and HOG feature extraction methods are proposed for fall detection and in-bed posture detection, respectively. The fall detection relies on an 8x8 pixel Grid-EYE sensor and a distance sensor. Experimental studies include three non-fall activities - standing, sitting, and stooping, and two fall actions – forward falling and sideway falling to simulate elderly daily activities. Different feature sets for the SVM-based machine learning algorithm are analyzed, and their impact on fall detection accuracy is evaluated and compared empirically. Overall accuracy above 90% is achieved. For the in-bed posture detection, the preprocessing method and feature extraction approach based on the HOG features and the principal component analysis (HOG+PCA) is compared to the traditional hand-crafted feature classification. The user study shows that a total of 9 in-bed postures can be successfully classified with 5-fold cross-validation-accuracy over

99.8%. The cross-user-validation is also performed to evaluate the robustness of posture classification models. The HOG features show significantly higher accuracy than handcrafted features. The 4-posture classification achieves an accuracy of 89.0% when using the SVM classifier, and the 9-posture classification achieves an accuracy of 86.2% when using the NN classifier based on 30 principal components features. On the other hand, the cross-user-validation using hand-crafted feature extraction is not satisfied (<70.0%), which indicates that they are only suitable for the personalized model.

In Chapter 5, a pre-trained CNN model is proposed as a feature extractor for facing direction detection. A customized CNN is proposed to fuse the data from two types of sensors for occupancy identification. The facing direction detection experiments are performed at three distances, 0.6m, 1.2m, and 1.8m. The CNN-based feature extraction demonstrates more reliable performance for facing direction detection compared to the handcrafted features regardless of the detection ranges. For example, the average accuracy reaches 89.1%, 95.3%, and 95.1% at the distance of 0.6m, 1.2m, and 1.8m, respectively. A non-intrusive sensor node is prototyped based on a thermopile array and four distance sensors for occupancy identification. The sensor node is installed near the lab entrance as the realistic setup to get the dataset, and 94.4% accuracy is achieved to identify eight different people. The proposed approach also shows a higher accuracy and lower power consumption compared to the state-of-the-art.

6.2. Future Work

The SLEEPIR sensor nodes have been tested within the lab and residential apartment for weeks. However, the activity and number of the human testing subject

involved in the experiment are still limited. In addition, based on the proposed mathematical model, the SLEEPIR sensor output is sensitive to the posture and clothing of the testing subjects. So future work should include more variances in the dataset. Currently, only a single SLEEPIR sensor node is deployed for occupancy detection in a single room, and its application is limited to presence detection. A SLEEPIR sensor network can be applied to extend the SLEEPIR system for fine-grained occupancy objectives such as animal detection and human activity detection[118, 119].

The proposed deep learning model has shown high accuracy for occupancy identification. However, the variance in the dataset is still limited. For example, only one walking path is included in the dataset, and all the users are not carrying items such as a bag during the experiments. Furthermore, the same occupancy may have different clothing and shoe heights. The sensitivity of the occupancy identification model needs to be analyzed under different scenarios.

The cross-user validation accuracy is still limited as it is usually hard to have a general model for different users. A possible reason is that the feature extracted may highly depend on the identity of the testing subject. Therefore, the desired feature extractor should extract features only related to the postures instead of identity. Deep learning models such as CNN may also cause overfitting due to their strong feature extraction capabilities. Domain-adversarial training method can be a solution to improve the cross-user validation accuracy[120]. For example, an occupancy identification classifier may serve as the domain classifier to help extract robust features during deep learning model training[121]. In addition, multitask learning could be another method to improve the

generalization of models by using the training information related to other tasks[122] [123]. For example, a multitask model is trained for motion classification and occupancy identification[48]. In-bed posture and identification can be detected together with a deep multitask model [124] .

REFERENCES

[1] X. Chen, Q. Wang, and J. Srebric, "Occupant feedback based model predictive control for thermal comfort and energy optimization: A chamber experimental evaluation," *Applied Energy*, vol. 164, pp. 341-351, 2016.

[2] S. Naylor, M. Gillott, and T. Lau, "A review of occupant-centric building control strategies to reduce building energy use," *Renewable and Sustainable Energy Reviews*, vol. 96, pp. 1-10, 2018.

[3] M. F. Haniff, H. Selamat, R. Yusof, S. Buyamin, and F. S. Ismail, "Review of HVAC scheduling techniques for buildings towards energy-efficient and cost-effective operations," *Renewable and Sustainable Energy Reviews*, vol. 27, pp. 94-103, 2013.

[4] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," presented at the Proceedings of the 2nd ACM Workshop on Embedded Sensing Systems for Energy-Efficiency in Building, Zurich, Switzerland, 2010.

[5] T. Guettari, D. Istrate, J. Boudy, B.-E. Benkelfat, B. Fumel, and J.-C. Daviet, "Design and First Evaluation of a Sleep Characterization Monitoring System Using a Remote Contactless Sensor," *IEEE Journal of Biomedical and Health Informatics*, vol. 21, no. 6, pp. 1511-1523, 2017.

[6] F. Erden, S. Velipasalar, A. Z. Alkar, and A. E. Cetin, "Sensors in Assisted Living: A survey of signal and image processing methods," *IEEE Signal Processing Magazine*, vol. 33, no. 2, pp. 36-44, 2016.

[7] V. Verhaert, B. Haex, T. De Wilde, D. Berckmans, M. Vandekerckhove, J. Verbraecken, and J. Vander Sloten, "Unobtrusive assessment of motor patterns during sleep based on mattress indentation measurements," *IEEE Transactions on Information Technology in Biomedicine*, vol. 15, no. 5, pp. 787-794, 2011.

[8] Y. S. Delahoz and M. A. Labrador, "Survey on fall detection and fall prevention using wearable and external sensors," *Sensors*, vol. 14, no. 10, pp. 19806-19842, 2014.

[9] M. Veselý and W. Zeiler, "Personalized conditioning and its impact on thermal comfort and energy performance–A review," *Renewable and Sustainable Energy Reviews*, vol. 34, pp. 401-408, 2014.

[10] G. Mokhtari, N. Bashi, Q. Zhang, and G. Nourbakhsh, "Non-wearable human identification sensors for smart home environment: a review," *Sensor Review*, vol. 38, no. 3, pp. 391-404, 2018.

[11] J. Kim, S. Schiavon, and G. Brager, "Personal comfort models–A new paradigm in thermal comfort for occupant-centric environmental control," *Building and Environment,* vol. 132, pp. 114-124, 2018.

[12] J. Kaartinen, I. Kuhlman, and P. Peura, "Long-term monitoring of movements in bed and their relation to subjective sleep quality," *Sleep and Hypnosis,* vol. 5, pp. 145-153, 2003.

[13] J. Razjouyan, H. Lee, S. Parthasarathy, J. Mohler, A. Sharafkhaneh, and B. Najafi, "Improving sleep quality assessment using wearable sensors by including information from postural/sleep position changes and body acceleration: a comparison of chest-worn sensors, wrist actigraphy, and polysomnography," *Journal of Clinical Sleep Medicine,* vol. 13, no. 11, pp. 1301-1310, 2017.

[14] P. Liu, S. K. Nguang, and A. Partridge, "Occupancy Inference Using Pyroelectric Infrared Sensors Through Hidden Markov Models," *IEEE Sensors Journal,* vol. 16, no. 4, pp. 1062-1068, 2016.

[15] B. Huchuk, S. Sanner, and W. O'Brien, "Comparison of machine learning models for occupancy prediction in residential buildings using connected thermostat data," *Building and Environment,* vol. 160, p. 106177, Aug 2019.

[16] E. Soltanaghaei and K. Whitehouse, "Practical occupancy detection for programmable and smart thermostats," *Applied Energy,* vol. 220, pp. 842-855, 2018.

[17] Y. Peng, A. Rysanek, Z. Nagy, and A. Schlüter, "Using machine learning techniques for occupancy-prediction-based cooling control in office buildings," *Applied Energy,* vol. 211, pp. 1343-1358, 2018.

[18] Y. P. Raykov, E. Ozer, G. Dasika, A. Boukouvalas, and M. A. Little, "Predicting room occupancy with a single passive infrared (PIR) sensor through behavior extraction," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* 2016, pp. 1016-1027.

[19] P. De, A. Chatterjee, and A. Rakshit, "PIR Sensor-Based AAL Tool for Human Movement Detection: Modified MCP-Based Dictionary Learning Approach," *IEEE Transactions on Instrumentation and Measurement,* vol. 69, no. 10, pp. 7377-7385, 2020.

[20] J. Yun and J. Woo, "A Comparative Analysis of Deep Learning and Machine Learning on Detecting Movement Directions Using PIR Sensors," *IEEE Internet of Things Journal,* vol. 7, no. 4, pp. 2855-2868, 2019.

[21] J. Zou, Q. Zhao, W. Yang, and F. Wang, "Occupancy detection in the office by analyzing surveillance videos and its application to building energy conservation," *Energy and Buildings,* vol. 152, pp. 385-398, 2017.

[22] N. Cao, J. Ting, S. Sen, and A. Raychowdhury, "Smart sensing for HVAC control: Collaborative intelligence in optical and IR cameras," *IEEE Transactions on Industrial Electronics,* vol. 65, no. 12, pp. 9785-9794, 2018.

[23] H. Na, J.-H. Choi, H. Kim, and T. Kim, "Development of a human metabolic rate prediction model based on the use of Kinect-camera generated visual data-driven approaches," *Building and Environment,* vol. 160, p. 106216, Aug 2019.

[24] Z. Chen, R. Zhao, Q. Zhu, M. K. Masood, Y. C. Soh, and K. Mao, "Building occupancy estimation with environmental sensors via CDBLSTM," *IEEE Transactions on Industrial Electronics,* vol. 64, no. 12, pp. 9549-9559, 2017.

[25] L. Zimmermann, R. Weigel, and G. Fischer, "Fusion of Nonintrusive Environmental Sensors for Occupancy Detection in Smart Homes," *IEEE Internet of Things Journal,* vol. 5, no. 4, pp. 2343-2352, 2018.

[26] L. M. Candanedo and V. Feldheim, "Accurate occupancy detection of an office room from light, temperature, humidity and CO2 measurements using statistical learning models," *Energy and Buildings,* vol. 112, pp. 28-39, 2016.

[27] Z. Chen and Y. Wang, "Remote Recognition of In-Bed Postures Using a Thermopile Array Sensor With Machine Learning," *IEEE Sensors Journal,* vol. 21, no. 9, pp. 10428-10436, 2021.

[28] Y. Yuan, X. Li, Z. Liu, and X. Guan, "Occupancy estimation in buildings based on infrared array sensors detection," *IEEE Sensors Journal,* vol. 20, no. 2, pp. 1043-1053, 2019.

[29] V. Chidurala and X. Li, "Occupancy Estimation Using Thermal Imaging Sensors and Machine Learning Algorithms," *IEEE Sensors Journal,* vol. 21, no. 6, pp. 8627-8638, 2021.

[30] A. Naser, A. Lotfi, and J. Zhong, "Adaptive thermal sensor array placement for human segmentation and occupancy estimation," *IEEE Sensors Journal,* vol. 21, no. 2, pp. 1993-2002, 2020.

[31] H. Zou, H. Jiang, J. Yang, L. Xie, and C. Spanos, "Non-intrusive occupancy sensing in commercial buildings," *Energy and Buildings,* vol. 154, pp. 633-643, Nov 1 2017.

[32] J. Yang, H. Zou, H. Jiang, and L. Xie, "Device-free occupant activity sensing using WiFi-enabled IoT devices for smart homes," *IEEE Internet of Things Journal,* vol. 5, no. 5, pp. 3991-4002, 2018.

[33] J. W. Choi, D. H. Yim, and S. H. Cho, "People counting based on an IR-UWB radar sensor," *IEEE Sensors Journal,* vol. 17, no. 17, pp. 5717-5727, 2017.

[34] H. Yan, Y. Zhang, Y. Wang, and K. Xu, "WiAct: A passive WiFi-based human activity recognition system," *IEEE Sensors Journal,* vol. 20, no. 1, pp. 296-305, 2019.

[35] J. Lu, T. Zhang, F. Hu, and Q. Hao, "Preprocessing Design in Pyroelectric Infrared Sensor-Based Human-Tracking System: On Sensor Selection and Calibration," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 47, no. 2, pp. 263-275, 2017.

[36] J. Lu, J. Gong, Q. Hao, and F. Hu, "Multi-agent based wireless pyroelectric infrared sensor networks for multi-human tracking and self-calibration," in *Proceedings of the 2013 IEEE Sensors Conference* 2013, pp. 1-4.

[37] B. Yang and M. Zhang, "Credit-Based Multiple Human Location for Passive Binary Pyroelectric Infrared Sensor Tracking System: Free From Region Partition and Classifier," *IEEE Sensors Journal,* vol. 17, no. 1, pp. 37-45, 2017.

[38] T. Yang, P. Guo, W. Liu, X. Liu, and T. Hao, "Enhancing PIR-Based Multi-Person Localization Through Combining Deep Learning With Domain Knowledge," *IEEE Sensors Journal,* vol. 21, no. 4, pp. 4874-4886, 2020.

[39] H. Wang, H. Zheng, J. C. Augusto, S. Martin, M. Mulvenna, W. Carswell, J. Wallace, P. Jeffers, B. Taylor, and K. McSorley, "Monitoring and analysis of sleep pattern for people with early dementia," in *Proceedings of the 2010 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW)* 2010, pp. 405-410.

[40] M. Gochoo, T.-H. Tan, S.-H. Liu, F.-R. Jean, F. S. Alnajjar, and S.-C. Huang, "Unobtrusive activity recognition of elderly people living alone using anonymous binary sensors and DCNN," *IEEE Journal of Biomedical and Health Informatics,* vol. 23, no. 2, pp. 693-702, 2018.

[41] R. Ma, F. Hu, and Q. Hao, "Active Compressive Sensing via Pyroelectric Infrared Sensor for Human Situation Recognition," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 47, no. 12, pp. 3340-3350, 2017.

[42] Q. Guan, C. Li, L. Qin, and G. Wang, "Daily activity recognition using pyroelectric infrared sensors and reference structures," *IEEE Sensors Journal*, vol. 19, no. 5, pp. 1645-1652, 2018.

[43] J. Lee, M. Hong, and S. Ryu, "Sleep monitoring system using kinect sensor," *International Journal of Distributed Sensor Networks*, vol. 11, no. 10, p. 875371, 2015.

[44] A. Khalili, A.-H. Soliman, M. Asaduzzaman, and A. Griffiths, "Wi-Fi sensing: Applications and challenges," *The Journal of Engineering*, vol. 2020, no. 3, pp. 87-97, 2020.

[45] W. Wang, A. X. Liu, and M. Shahzad, "Gait recognition using wifi signals," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing* 2016, pp. 363-373.

[46] J. Liu, Y. Chen, Y. Wang, X. Chen, J. Cheng, and J. Yang, "Monitoring Vital Signs and Postures During Sleep Using WiFi Signals," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2071-2084, 2018.

[47] J. W. Choi, X. Quan, and S. H. Cho, "Bi-directional passing people counting system based on IR-UWB radar sensors," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 512-522, 2018.

[48] Y. Lang, Q. Wang, Y. Yang, C. Hou, H. Liu, and Y. He, "Joint motion classification and person identification via multitask learning for smart homes," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 9596-9605, 2019.

[49] W. Wang, T. Hong, N. Xu, X. Xu, J. Chen, and X. Shan, "Cross-source sensing data fusion for building occupancy prediction with adaptive lasso feature filtering," *Building and Environment*, vol. 162, p. 106280, 2019.

[50] A. K. Mikkilineni, J. Dong, T. Kuruganti, and D. Fugate, "A Novel Occupancy Detection Solution using Low-Power IR-FPA Based Wireless Occupancy Sensor," *Energy and Buildings*, vol. 192, pp. 63-74, 2019.

[51] Y. Wang, S. Cang, and H. Yu, "A data fusion-based hybrid sensory system for older people's daily activity and daily routine recognition," *IEEE Sensors Journal*, vol. 18, no. 16, pp. 6874-6888, 2018.

[52] T. H. Pedersen, K. U. Nielsen, and S. Petersen, "Method for room occupancy detection based on trajectory of indoor climate sensor data," *Building and Environment*, vol. 115, pp. 147-156, 2017.

[53] B. W. Hobson, D. Lowcay, H. B. Gunay, A. Ashouri, and G. R. Newsham, "Opportunistic occupancy-count estimation using sensor fusion: A case study," *Building and environment,* vol. 159, p. 106154, 2019.

[54] L. Wu and Y. Wang, "A Low-Power Electric-Mechanical Driving Approach for True Occupancy Detection Using a Shuttered Passive Infrared Sensor," *IEEE Sensors Journal,* vol. 19, no. 1, pp. 47-57, 2018.

[55] H. Liu, Y. Wang, K. Wang, and H. Lin, "Turning a pyroelectric infrared motion sensor into a high-accuracy presence detector by using a narrow semi-transparent chopper," *Applied Physics Letters,* vol. 111, no. 24, p. 243901, 2017.

[56] L. Wu, Y. Wang, and H. Liu, "Occupancy detection and localization by monitoring nonlinear energy flow of a shuttered passive infrared sensor," *IEEE Sensors Journal,* vol. 18, no. 21, pp. 8656-8666, 2018.

[57] L. Wu, F. Gou, S.-T. Wu, and Y. Wang, "SLEEPIR: Synchronized Low-Energy Electronically Chopped PIR Sensor for True Presence Detection," *IEEE Sensors Letters,* vol. 4, no. 3, pp. 1-4, 2020.

[58] L. Wu and Y. Wang, "Stationary and Moving Occupancy Detection Using the SLEEPIR Sensor Module and Machine Learning," *IEEE Sensors Journal,* vol. 21, no. 13, pp. 14701-14708, 2021.

[59] L. Wu and Y. Wang, "Performance Optimization of the SLEEPIR Sensor Towards Indoor Stationary Occupancy Detection," *IEEE Sensors Journal,* vol. 21, no. 21, pp. 23776-23786, 2021.

[60] J. Andrews, M. Kowsika, A. Vakil, and J. Li, "A motion induced passive infrared (PIR) sensor for stationary human occupancy detection," in *Proceedings of the 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)* 2020, pp. 1295-1304.

[61] Y. Man-Soon, H. Sung-Moo, and U. Soon-Chul, "A newly designed chopper for pyroelectric infrared sensor by using a dome-shaped piezoelectric linear motor (DSPLM)," *Journal of Electroceramics,* vol. 23, no. 2-4, p. 242, 2009.

[62] A. Beltran, V. L. Erickson, and A. E. Cerpa, "Thermosense: Occupancy thermal based sensing for hvac control," in *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings* 2013, pp. 1-8.

[63] A. Tyndall, R. Cardell-Oliver, and A. Keating, "Occupancy estimation using a low-pixel count thermal imager," *IEEE Sensors Journal,* vol. 16, no. 10, pp. 3784-3791, 2016.

[64] Z. Chen and Y. Wang, "Infrared–ultrasonic sensor fusion for support vector machine–based fall detection," *J. Intell. Mater. Syst. Struct.*, vol. 29, no. 9, pp. 2027-2039, 2018.

[65] S. Mashiyama, J. Hong, and T. Ohtsuki, "Activity recognition using low resolution infrared array sensor," in *Proceedings of the 2015 IEEE International Conference on Communications (ICC)* 2015, pp. 495-500.

[66] M. Gochoo, T.-H. Tan, S.-C. Huang, T. Batjargal, J.-W. Hsieh, F. S. Alnajjar, and Y.-F. Chen, "Novel IoT-Based Privacy-Preserving Yoga Posture Recognition System Using Low-Resolution Infrared Sensors and Deep Learning," *IEEE Internet of Things Journal*, 2019.

[67] Z. Chen, Y. Wang, and H. Liu, "Unobtrusive Sensor based Occupancy Facing Direction Detection and Tracking using Advanced Machine Learning Algorithms," *IEEE Sensors Journal*, vol. 18, no. 15, pp. 6360-6368, 2018.

[68] D. Qu, B. Yang, and N. Gu, "Indoor Multiple Human Targets Localization and Tracking using Thermopile Sensor," *Infrared Physics & Technology*, 2019.

[69] D. Hauschildt and N. Kirchhof, "Advances in Thermal Infrared Localization: Challenges and Solutions," in *Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation*, R. Mautz, M. Kunz, and H. Ingensand, Eds., ed, 2010.

[70] M. Kuki, H. Nakajima, N. Tsuchiya, and Y. Hata, "Multi-human locating in real environment by thermal sensor," in *Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics* 2013, pp. 4623-4628.

[71] N. Belapurkar, J. Harbour, S. Shelke, and B. Aksanli, "Building Data-Aware and Energy-Efficient Smart Spaces," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 4526-4537, 2018.

[72] P. Hevesi, S. Wille, G. Pirkl, N. Wehn, and P. Lukowicz, "Monitoring household activities and user location with a cheap, unobtrusive thermal sensor array," presented at the Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing (UBICOMP), 2014.

[73] A. Ghahramani, G. Castro, B. Becerik-Gerber, and X. Yu, "Infrared thermography of human face for monitoring thermoregulation performance and estimating personal thermal comfort," *Building and Environment*, vol. 109, pp. 1-11, 2016.

[74] H. Saha, A. R. Florita, G. P. Henze, and S. Sarkar, "Occupancy sensing in buildings: A review of data analytics approaches," *Energy and Buildings*, vol. 188, pp. 278-285, 2019.

[75] H. Choi, C. Y. Um, K. Kang, H. Kim, and T. Kim, "Review of vision-based occupant information sensing systems for occupant-centric control," *Building and Environment*, vol. 203, p. 108064, 2021.

[76] Z. Yang, N. Li, B. Becerik-Gerber, and M. Orosz, "A systematic approach to occupancy modeling in ambient sensor-rich buildings," *Simulation*, vol. 90, no. 8, pp. 960-977, 2014.

[77] M. Zuraimi, A. Pantazaras, K. Chaturvedi, J. Yang, K. Tham, and S. Lee, "Predicting occupancy counts using physical and statistical Co2-based modeling methodologies," *Building and Environment*, vol. 123, pp. 517-528, 2017.

[78] J. Chaney, E. H. Owens, and A. D. Peacock, "An evidence based approach to determining residential occupancy and its role in demand response management," *Energy and Buildings*, vol. 125, pp. 254-266, 2016.

[79] L. M. Candanedo, V. Feldheim, and D. Deramaix, "A methodology based on Hidden Markov Models for occupancy detection and a case study in a low energy residential building," *Energy and Buildings*, vol. 148, pp. 327-341, 2017.

[80] H.-B. Zhang, Y.-X. Zhang, B. Zhong, Q. Lei, L. Yang, J.-X. Du, and D.-S. Chen, "A comprehensive survey of vision-based human action recognition methods," *Sensors*, vol. 19, no. 5, p. 1005, 2019.

[81] I. Rodríguez-Moreno, J. M. Martínez-Otzeta, B. Sierra, I. Rodriguez, and E. Jauregi, "Video activity recognition: State-of-the-art," *Sensors*, vol. 19, no. 14, p. 3160, 2019.

[82] J. Yang, A. Pantazaras, K. A. Chaturvedi, A. K. Chandran, M. Santamouris, S. E. Lee, and K. W. Tham, "Comparison of different occupancy counting methods for single system-single zone applications," *Energy and Buildings*, vol. 172, pp. 221-234, 2018.

[83] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of 2005 International Conference on Computer Vision & Pattern Recognition*, San Diego, United States, 2005-06-20 2005, pp. 886--893.

[84] D. Liu, X. Guan, Y. Du, and Q. Zhao, "Measuring indoor occupancy in intelligent buildings using the fusion of vision sensors," *Measurement Science and Technology*, vol. 24, no. 7, p. 074023, 2013.

[85] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition* 2016, pp. 779-788.

[86] H. Na, H. Choi, and T. Kim, "Metabolic rate estimation method using image deep learning," in *Building Simulation* 2020, pp. 1077-1093.

[87] E. J. Choi, Y. Yoo, B. R. Park, Y. J. Choi, and J. W. Moon, "Development of occupant pose classification model using deep neural network for personalized thermal conditioning," *Energies,* vol. 13, no. 1, p. 45, 2020.

[88] H. Jiang and E. Learned-Miller, "Face detection with the faster R-CNN," in *Proceedings of the 12th IEEE International Conference on Automatic Face & Gesture Recognition* 2017, pp. 650-657.

[89] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, "Deep learning for sensor-based activity recognition: A survey," *Pattern Recognition Letters,* 2018.

[90] H. Zou, Y. Zhou, J. Yang, and C. J. Spanos, "Towards occupant activity driven smart buildings via WiFi-enabled IoT devices and deep learning," *Energy and Buildings,* vol. 177, pp. 12-22, 2018.

[91] C. Perra, A. Kumar, M. Losito, P. Pirino, M. Moradpour, and G. Gatto, "Monitoring Indoor People Presence in Buildings Using Low-Cost Infrared Sensor Array in Doorways," *Sensors,* vol. 21, no. 12, p. 4062, 2021.

[92] Z. Liu, M. Yang, Y. Yuan, and K. Y. Chan, "Fall detection and personnel tracking system using infrared array sensors," *IEEE Sensors Journal,* vol. 20, no. 16, pp. 9558-9566, 2020.

[93] Y. Zhang and B. Yang, "Traffic Flow Detection Using Thermopile Array Sensor," *IEEE Sensors Journal,* vol. 20, no. 10, pp. 5155-5164, 2020.

[94] N. Gu, B. Yang, and T. Zhang, "Dynamic Fuzzy Background Removal for Indoor Human Target Perception Based on Thermopile Array Sensor," *IEEE Sensors Journal,* vol. 20, no. 1, pp. 67-76, 2019.

[95] J. Fraden, "Handbook of modern sensors," ed: Springer, 2013.

[96] J. Yun and M. Song, "Detecting Direction of Movement Using Pyroelectric Infrared Sensors," *IEEE Sensors Journal,* vol. 14(5), no. 5, pp. 1482-1489, 2014.

[97] T. Yokoishi, J. Mitsugi, O. Nakamura, and J. Murai, "Room occupancy determination with particle filtering of networked pyroelectric infrared (PIR) sensor data," in *Proceedings of the 2012 IEEE Sensors* 2012, pp. 1-4.

[98] M. Gochoo, T.-H. Tan, V. Velusamy, S.-H. Liu, D. Bayanduuren, and S.-C. Huang, "Device-free non-privacy invasive classification of elderly travel patterns in a smart house using PIR sensors and DCNN," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 390-400, 2017.

[99] M. Barandas, D. Folgado, L. Fernandes, S. Santos, M. Abreu, P. Bota, H. Liu, T. Schultz, and H. Gamboa, "TSFEL: Time series feature extraction library," *SoftwareX*, vol. 11, p. 100456, 2020.

[100] M. Christ, N. Braun, J. Neuffer, and A. W. Kempa-Liehr, "Time series feature extraction on basis of scalable hypothesis tests (tsfresh–a python package)," *Neurocomputing*, vol. 307, pp. 72-77, 2018.

[101] H. Mohammadmoradi, S. Munir, O. Gnawali, and C. Shelton, "Measuring People-Flow Through Doorways using Easy-to-Install IR Array Sensors," in *Proceedings of the 13th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, 2017, pp. 35-43.

[102] S. Mashiyama, J. Hong, and T. Ohtsuki, "A fall detection system using low resolution infrared array sensor," in *Proceedings of the 25th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communication (PIMRC)* 2014, pp. 2109-2113.

[103] *Starfish or Freefall? What Your Sleep Position Can Tell You*. Available: https://bettersleep.org/better-sleep/sleep-positions/

[104] Z. Chen and Y. Wang, "Sleep monitoring using an infrared thermal array sensor," in *Proceedings of the 2019 Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems* 2019, p. 109701D.

[105] J. Wu, L. Sun, and R. Jafari, "A Wearable System for Recognizing American Sign Language in Real-Time Using IMU and Surface EMG Sensors," *IEEE Journal of Biomedical and Health Informatics*, vol. 20, no. 5, pp. 1281-1290, 2016.

[106] J. J. Liu, W. Xu, M.-C. Huang, N. Alshurafa, M. Sarrafzadeh, N. Raut, and B. Yadegar, "Sleep posture analysis using a dense pressure sensitive bedsheet," *Pervasive and Mobile Computing*, vol. 10, pp. 34-50, 2014.

[107] H. Lu and Y. Li, "Gesture on: Enabling always-on touch gestures for fast mobile access from the device standby mode," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* 2015, pp. 3355-3364.

[108] G. Matar, J.-M. Lina, and G. Kaddoum, "Artificial neural network for in-bed posture classification using bed-sheet pressure sensors," *IEEE Journal of Biomedical and Health Informatics,* vol. 24, no. 1, pp. 101-110, 2019.

[109] X. Xu, F. Lin, A. Wang, Y. Hu, M.-C. Huang, and W. Xu, "Body-earth mover's distance: A matching-based approach for sleep posture recognition," *IEEE Transactions on Biomedical Circuits and Systems,* vol. 10, no. 5, pp. 1023-1035, 2016.

[110] A. Vedaldi and A. Zisserman. (2016). *VGG Convolutional Neural Networks Practical*. Available: http://www.robots.ox.ac.uk/~vgg/practicals/cnn/#vgg-convolutional-neural-networks-practical

[111] T. W. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse, "Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors," in *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems* 2012, pp. 309-322.

[112] G. Mokhtari, Q. Zhang, G. Nourbakhsh, S. Ball, and M. Karunanithi, "BLUESOUND: A new resident identification sensor using ultrasound array and BLE technology for smart home platform," *IEEE Sensors Journal,* vol. 17, no. 5, pp. 1503-1512, 2017.

[113] N. Khalil, D. Benhaddou, O. Gnawali, and J. Subhlok, "Nonintrusive ultrasonic-based occupant identification for energy efficient smart building applications," *Applied Energy,* vol. 220, pp. 814-828, June 2018.

[114] G. Mokhtari, Q. Zhang, C. Hargrave, and J. C. Ralston, "Non-wearable UWB sensor for human identification in smart home," *IEEE Sensors Journal,* vol. 17, no. 11, pp. 3332-3340, 2017.

[115] T. Xin, B. Guo, Z. Wang, M. Li, Z. Yu, and X. Zhou, "Freesense: Indoor human identification with Wi-Fi signals," in *Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM)* 2016, pp. 1-7.

[116] Y. Zeng, P. H. Pathak, and P. Mohapatra, "WiWho: wifi-based person identification in smart spaces," in *Proceedings of the 15th International Conference on Information Processing in Sensor Networks* 2016, p. 4.

[117] J. Han, S. Pan, M. K. Sinha, H. Y. Noh, P. Zhang, and P. Tague, "Sensetribute: smart home occupant identification via fusion across on-object sensing devices," presented at the Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments, 2017.

[118] A. Naser, A. Lotfi, J. Zhong, and J. He, "Human activity of daily living recognition in presence of an animal pet using thermal sensor array," in *Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments* 2020, pp. 1-6.

[119] R. C. Luo and O. Chen, "Wireless and Pyroelectric Sensory Fusion System for Indoor Human/Robot Localization and Monitoring," *IEEE/ASME Transactions on Mechatronics,* vol. 18, no. 3, p. 845, 2013.

[120] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research,* vol. 17, no. 1, pp. 2096-2030, 2016.

[121] L. Zhang, N. C. Hurley, B. Ibrahim, E. Spatz, H. M. Krumholz, R. Jafari, and M. J. Bobak, "Developing Personalized Models of Blood Pressure Estimation from Wearable Sensors Data Using Minimally-trained Domain Adversarial Neural Networks," in *Proceedings of the 2020 Machine Learning for Healthcare Conference* 2020, pp. 97-120.

[122] Y. Zhang and Q. Yang, "A survey on multi-task learning," *IEEE Transactions on Knowledge and Data Engineering, Early Access*, 2021.

[123] H. Yang, S. Gong, Y. Liu, Z. Lin, and Y. Qu, "A multi-task learning model for daily activity forecast in smart home," *Sensors,* vol. 20, no. 7, p. 1933, 2020.

[124] V. Davoodnia and A. Etemad, "Identity and Posture Recognition in Smart Beds with Deep Multitask Learning," in *Proceedings of the 2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)* 2019, pp. 3054-3059.