Smart Streetlight Energy Saving System Based on mmWave Radar

Yu-Kai Lin, Jhih-Ci Li, Kai-Li Wang, Yu-Ping Liao[*]

Department of Electrical Engineering, Chung Yuan Christian University, Taoyuan, Taiwan, ROC

Abstract

Streetlights serve as fundamental infrastructure to meet the lighting needs of people on every road. However, their extensive deployment often results in unnecessary energy waste, with many streetlights maintaining high brightness despite minimal usage during the night. This study aims to develop a smart energy-efficient streetlight system that automatically adjusts lighting levels based on the absence of vehicles and pedestrians, detected after a 3-minute countdown. Specifically, the study utilizes mmWave radar to collect point cloud data, which undergoes denoising through Doppler, DBSCAN, and XYZ techniques. Additionally, the mmWave radar assists in training an LSTM model to identify pedestrian pathways. The implementation of the proposed system significantly reduces energy consumption and annual costs by automatically dimming or turning off streetlights in areas with minimal pedestrian activity during nighttime.

Keywords: deep learning, LSTM, mmWave radar, point cloud data, DBSCAN

1. Introduction

Given the need to reduce energy consumption and operating costs, the installation of smart systems has been globally implemented. Specifically, advanced smart operation technology is required to effectively control, manage, and communicate street lighting to minimize energy consumption. Hence, to expound on the technology, the following introduction will be divided into three parts, i.e., background, motivation, and finally, goal.
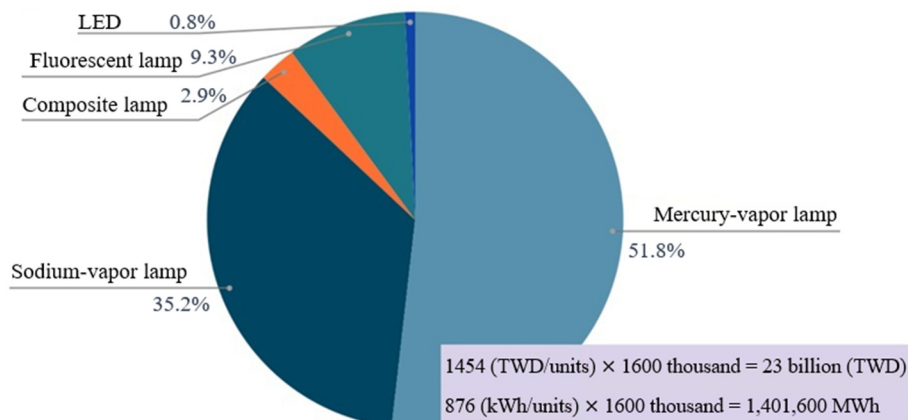
1.1. Background

Fig. 1 The proportion and resource consumption of street light types across Taiwan

Streetlights are the most widely installed infrastructure and are essential equipment to assure public safety. Due to their pervasive distribution and strict requirements for lighting direction and brightness, the power and specifications of streetlights are rigorously regulated. According to [1-2], as shown in Fig. 1, statistically, about 1.6 million streetlights are installed in

* Corresponding author. E-mail address: lyp@cycu.edu.tw

Taiwan, of which mercury streetlights account for 51.8%, high-pressure sodium lamps standing at 35.2%, metal halide lamps for 2.9%, fluorescent lamps for 9.3%, and LED street lights for 0.8%, respectively. Assuming all street lights are 200W traditional lights (non-LED) and are illuminated for 12 hours per day annually, they consume approximately 1,401,600,000 kWh (1,401,600 MWh) of electricity per year, with an annual cost of more than NT$2.3 billion (NT$1454 per light × 1.6 million lights).

Despite the truth lying in the satisfaction of people's need for lighting, the government confronts a huge financial burden, and reports have profusely emanated and discussed the inability of local governments to undertake the cost of streetlights. In addition, in recent years, the conflicts among energy supply, economy, environmental protection, and people's livelihoods have continued to emerge. Despite the ostensible insignificance, the economic and energy pressures incurred by streetlights have osmotically impacted daily life.

1.2. Motivation

Currently, despite the presence of research concerning mmWave radar [3-4], still, related products, research, and applications concerning energy-saving streetlights are insufficient. The recent practice is "AI Island" in Songdo, South Korea. However, artificial intelligence (AI) is widely applied to various products and services, and many electronic products increasingly seek to incorporate the internet of things (IoT). Given this trend, such a phenomenon is inevitable to develop a small device that can intelligently recognize and operate at the edge.

Furthermore, in comparison, the most suitable sensor for detecting objects at night is the millimeter wave radar. Millimeter wave radar technology is gradually maturing and is equipped with a certain level of software and hardware knowledge. Millimeter wave radar remains flexible functions for college students to use various data processing techniques for object detection in varied circumstances, accurately recognize target objects, and ultimately achieve the desired purpose through controlled hardware and components.

1.3. Goal

To address the issue of energy consumption of street lights in a state of being sustainably idle and illuminating at full brightness, this study proposes a method wherein the lights turn off after a 3-minute countdown in the absence of passing vehicles and pedestrians. Millimeter-wave radar will be deployed to detect specific targets, as illustrated in Fig. 2, and activate the lights to ensure driving safety. As a result, the project aims to achieve energy saving, carbon reduction, and cost reduction.

(a) System schematic diagram

"A car is driving on a road section without intersections."
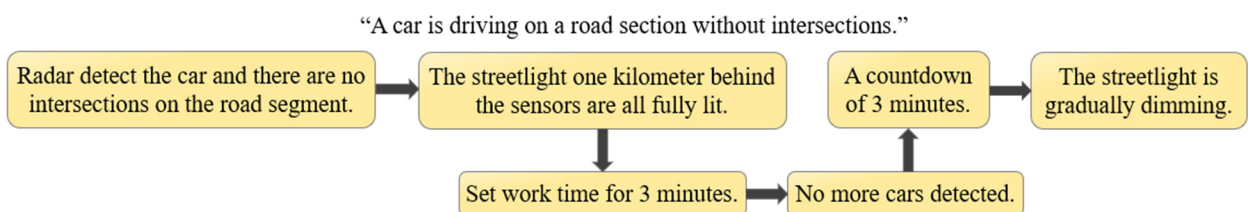
(b) System schematic flowchart

Fig. 2 Smart streetlight energy saving system schematic diagram

2. Research Methods

The implementation of smart street lighting systems is pervasive in large cities. Meanwhile, the millimeter-wave radar is perceived to be the most suitable sensor for nocturnal detection because of its long detection range and ability to identify objects under any nocturnal and meteorological conditions. In this section, the project approach will be further explored as follows.

2.1. Research methods

The project is mainly divided into four parts:

(1) Collecting and building a millimeter wave radar database on the Windows 10 operating system.

(2) Building and training a long short-term memory (LSTM) neural network model using PyTorch on the Windows 10 operating system.

(3) Developing a control program for HT32F52352 streetlights on the Windows 10 operating system.

(4) Integrating the millimeter wave program in the Ubuntu system of NVIDIA Jetson Nano and communicating with HT32F52352 to control streetlights.

In this project, the TI IWR6843 Single-Chip mmWave Sensor millimeter wave radar, which is mounted on the Batman BM501 mmWave EVM Kit [5], is used to collect multiple sets of 50-frame millimeter wave data on the Windows 10 operating system. Three noise reduction methods including Doppler, DBSCAN, and averaging over thousands of data points in each frame, are used to remove a large amount of background noise.

Next, an LSTM neural network model is built using PyTorch, and the pre-processed multiple datasets are sent to the LSTM neural network for training in a concatenated matrix form. After adjusting the parameters appropriately, the trained module can be exported for real-time recognition on the NVIDIA Jetson Nano later.

The streetlight control program runs on the HT32F52352 based on the ARM architecture and is developed using Keil v5. The main function is to use an interrupt function to calculate the time the light is on and control the brightness of the LED light board through pulse-width modulation (PWM).

The final step involves integrating the PyTorch-trained LSTM model with the millimeter wave radar on the NVIDIA Jetson Nano hardware. This integration allows the pre-processed millimeter wave data to be analyzed in real-time by the trained model to identify the direction of the target's movement. The resulting output is then sent to the GPIO on the NVIDIA Jetson Nano and forwarded to the HT32F52352 to activate the streetlights along the path. In cases where no target is detected, the lights are dimmed or turned off sequentially to conserve energy.

2.2. System architecture

The steps of the system process are enumerated as follows:

(1) The millimeter-wave radar on NVIDIA Jetson Nano receives detection data.

(2) The data is organized and denoised on NVIDIA Jetson Nano, and the trained LSTM model is used to perform calculations.

(3) The result is transmitted to HT32F52352 via GPIO.

(4) HT32F52352 controls the LED lights based on the recognition result.

The system architecture diagram is depicted in Fig. 3. The NVIDIA Jetson Nano is connected to the Batman BM501-PCR mmWave sensor through a USB port to collect point cloud data and perform recognition using the AI model, and the recognition results are then transmitted to the HT32F52352 via GPIO, while the HT32F52352 utilizes PWM to control the brightness of street lights.
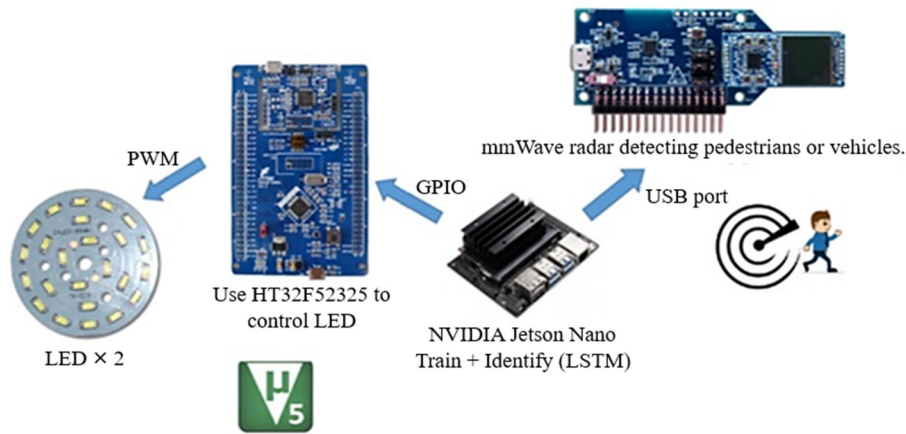
Fig. 3 System architecture diagram

2.3. TI IWR6843 single-chip mmWave sensor

The CNN image recognition has been employed for pedestrian and vehicle detection in some studies [6-8]. However, given that street lights are only turned on at night, using traditional cameras under low light conditions will result in lower recognition rates, as shown in Table 1. Even with infrared and thermal cameras, the effective range is limited to a maximum of 50 meters. Therefore, the decision is made to utilize a millimeter-wave radar capable of effectively detecting objects in any situation, with a working range of up to 250 meters.

Table 1 Comparison chart of common sensors [9]

	Radar	Lidar	Ultrasonic	Camera	Laser	Infrared
Range	Long	Medium	Short	Short	Short	Short
Range accuracy	High	High	High	Medium	High	Poor
Angle accuracy	Medium	High	None	High	High	Poor
Speed measurements	Yes	No	No	No	No	No
Dust/Fog/Smoke robustness	High	Medium	Poor	Poor	Poor	Poor
Dark/Light robustness	High	High	High	Medium	Poor	Medium
Efforts to integrate into machines	Small	High	Small	Medium	Medium	Small
Cost factor	Small to medium	Extremely high	Small	Medium to high	High	Small

Stepwisely, the process can be divided into three steps. First, the millimeter-wave radar emits radio waves and receives the reflected signals from the target. Second, the relative distance, velocity, angle, and motion direction of the target are calculated. Third, the aforementioned data is returned to the computer for processing and decision-making. Currently, medium-range radar (MRR), operating at 24 GHz, and long-range radar (LRR), operating at 77 GHz, are the main types of millimeter-wave radar used ubiquitously, as shown in Table 2.

Table 2 Advantages and disadvantages of millimeter-wave radars in different frequency bands

	Advantages	Disadvantages
40 GHz	Wide detection angle and cheap price.	The maximum detection range is approximately 50 meters.
60 GHz	The radar with a detection range of about 100 to 150 meters, with a medium price and a wider detection angle than 77GHz.	The detection distance is not farther than that of a 77GHz radar.
77 GHz	The radar with the farthest detection distance is up to 250 meters.	The radar with a narrow detection angle and is expensively priced.

The 24 GHz millimeter-wave radar used in autonomous driving, automatic parking, and other applications can only detect distances of approximately 50 meters. In the case of employing the 24 GHz millimeter-wave radar for the streetlights, a false activation after the passage of vehicles may incur. Although the 77 GHz millimeter-wave radar can detect longer distances and higher speeds, it demands a relatively high price. Therefore, the decision is made to utilize the IWR6843AOP Single-Chip 60-

to 64-GHz mmWave Sensor-Batman BM501, which has a detection range of approximately 100 to 150 meters, a lower price than purchasing only a 77 GHz radar, and a wider detection angle than 77 GHz. Relevant specifications are shown in Fig. 4 and Table 3 [5, 10].

Fig. 4 BM501 module on carrier board [5, 10]

Table 3 Configuration parameters [5, 10]

	Parameter	Configuration
1	Start frequency	60 GHz
2	Stop frequency	64 GHz
3	Bandwidth	4 GHz continuous bandwidth
4	TX power	15 dBm
5	RX noise figure	14 dB
6	Phase noise at 1 MHz	-92 dBc/Hz
7	Number of transmitters	3
8	Number of receivers	4
9	Azimuth field of view	120°
10	Elevation field of view	120°
11	The heights of 3 subjects	171, 180 and 182 cm

The output of the millimeter-wave module is segmented into Key data and Raw data. Key data has less data and uses a baud rate of 115200/8/n/1, while Raw data has more data and uses a baud rate of 921600/8/n/1. This project uses Raw data and opens the Jetson Nano's USB-UART to read data, enabling dial-out permission. The BM501-PCR will transmit three packets: v6, v7, and v8, which correspond to the point cloud, target object, and target index, respectively.

Concerning the classification and processing, the raw v6 (point cloud) data is deployed. The v6 packet contains seven pieces of information: frame number, type, elevation, azimuth, Doppler, range, and snr. In other words, the information represents frame number, data type, elevation angle, azimuth angle, velocity, distance to the radar, and signal-to-noise ratio of each point in the point cloud, respectively. By using the point cloud's elevation (ψ), azimuth (θ), and range (R), the x, y, and z positions of each point can be calculated. Elevation, azimuth, and range are in spherical coordinates, which are nuanced from the commonly used x, y, and z coordinates, but they can be transformed into point cloud's x, y, and z coordinates using a mathematical formula.

A 60 GHz millimeter-wave radar is employed, commencing with the detection of individuals within a 6-meter and 120-degree range. The team will repeatedly walk past in front of the millimeter-wave radar at the same distance and height. The 60 GHz radar transmitter generates radio frequency signals, which are then converted to low-frequency signals by the receiver. The signal is subsequently transmitted to the signal processor, which extracts information such as distance, velocity, and angle, and eventually is returned to the Jetson Nano for processing.

2.4. Millimeter-wave radar data collection

The output data of the BM501-PCR mmWave sensor can be processed into some data structures using the mmWave Python SDK. The v6 point cloud data is utilized to process and classify data. The data structure of v6 point cloud data is depicted in Fig. 5 [11]. The v6 point cloud data structure contains ten data items, from which the values of sx, sy, sz, and doppler can be used as input features for training the deep learning LSTM model to recognize object movement.

```
V6 data structure: (point cloud)
  fN            int64    #frameNum
  type          object   #v6
  elv           float64. #Elevation in radians
  azimuth       float64  #Azimuth in radians
  doppler       float64  #Doppler in m/s
  range         float64  #Range in meters
  snr           float64  #SNR, ratio
  sx            float64  #point cloud position in X, m
  sy            float64  #point cloud position in Y, m
  sz            float64  #point cloud position in Z, m


            ----------------- v6 dataFrame --------
      fN type  elv azimuth  ...       snr        sx        sy        sz
  0  2352   v6 0.04    0.09  ...  0.640000  0.034059  0.377413  0.015166
  1  2352   v6 0.04    0.10  ...  0.640000  0.037832  0.377054  0.015166
  2  2352   v6 0.04    0.11  ...  0.640000  0.041600  0.376656  0.015166
  3  2352   v6 0.04    0.12  ...  0.640000  0.045365  0.376222  0.015166
  4  2352   v6 0.04    0.13  ...  0.640000  0.049124  0.375749  0.015166
```

Fig. 5 Data structure of v6 point cloud [11]

Considering the position of the radar on the road, the range of data collection, and the ease of collecting data on the target for students, data collection on pedestrians is conducted, as shown in Fig. 6. The radar is installed at a height of 1.5 meters above the ground, and the range presents rectangularly with a length of 3 meters and a width of 2 meters centered on the radar. The pedestrian will walk straight 1 meter away from the radar at a speed of 2 m/5s, and the direction of the movement is either left or right.
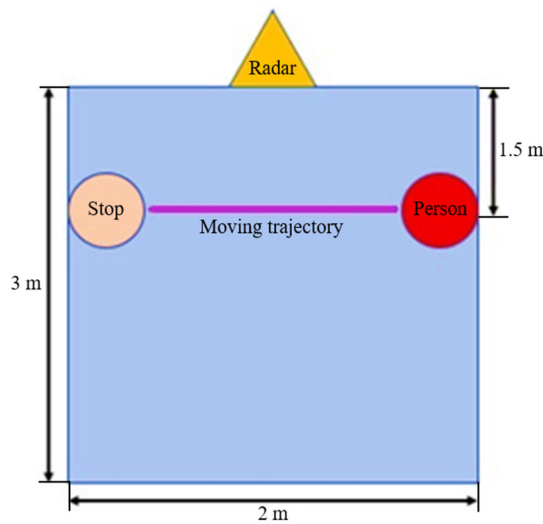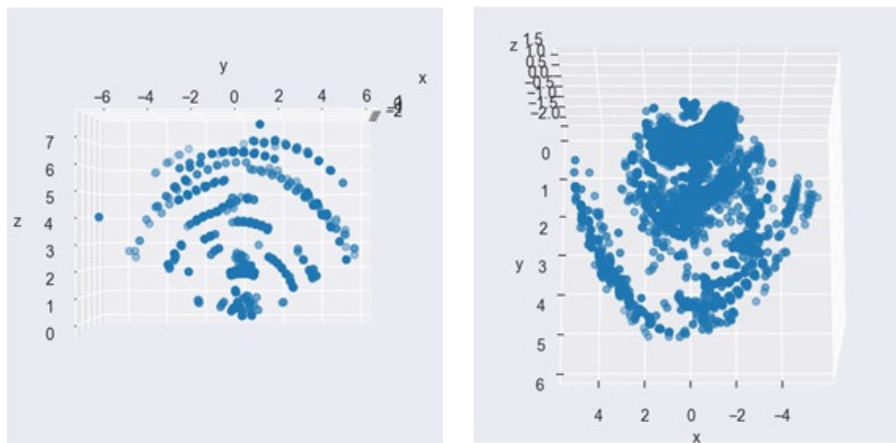

Fig. 6 Schematic of data collection range

(a) An originally single frame   (b) Superimposed frames

Fig. 7 Data before and after superimposing

It is noteworthy that partially observing point cloud data in each frame will incur the inability to target the location of the detected object and the inability to identify the object concerning the attributes, moving state, and additional information. Therefore, by superimposing multiple frames of point cloud data, as shown in Fig. 7, there will be considerable dense point cloud data in the detected place, which is helpful for position identification data processing and object classification.

2.5. Denoising

The collected point cloud data is initially noisy, with over 10,000 point clouds containing noise and all information about the target objects within the detection time and range of 50 frames. Thus, to filter out all irrelevant noise and extract detailed data of each target object in each frame, three denoising methods are used sequentially:

(1) Doppler filtering [12]

The point cloud data type provides Doppler data for each point, and Doppler is exactly defined as the speed of a certain point. By using Doppler, many static background noises are easily filterable, and only the point cloud data of the "walking person" that is desired to be retained is preserved. The condition is "if |doppler| < 0.1" which is shown in Fig. 8, the data of that point is eliminated.

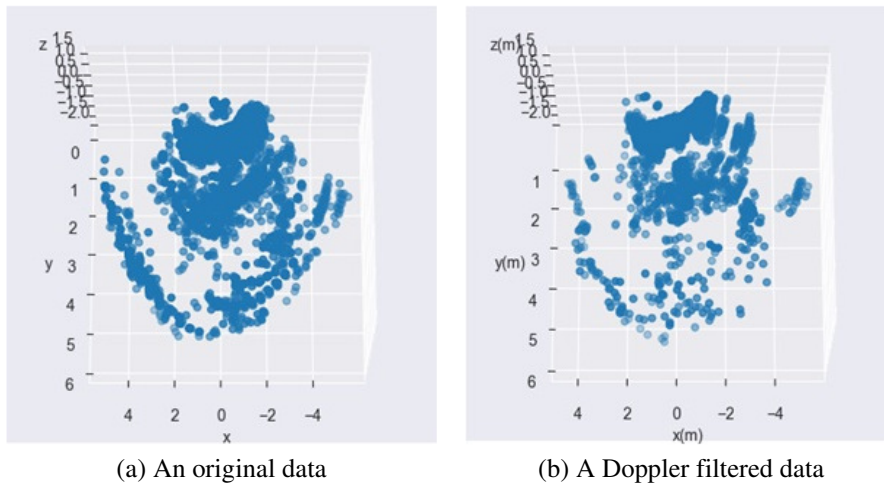(a) An original data　　　　(b) A Doppler filtered data

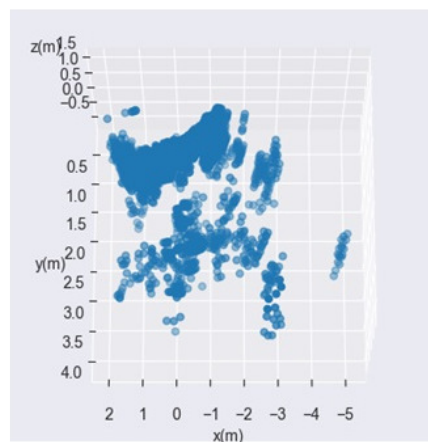Fig. 8 Data before and after Doppler filtering

(2) DBSCAN

Fig. 9 A data after DBSCAN

Despite the addition of the Doppler filter, millimeter-wave radar is somewhat susceptible to misidentifying stationary noise as moving. Such misidentification is environmentally caused by diffraction and interference and subsequently generates an incorrect Doppler value. However, most of the remaining point cloud data is located on the target and has a high density.

As a solution, numerous clustering algorithms based on cluster density can be employed to further filter out noise. In this context, the algorithm utilized is DBSCAN (Density-Based Spatial Clustering of Applications with Noise) proposed in 1996 [13]. Technically, DBSCAN is a density-based algorithm [14] that defines the user's literal definition of "high density" by inputting two parameters: eps (the radius at which a data point looks for other points) and min_samples (the minimum number of points in eps to be considered non-noise). As shown in Fig. 9, the parameters are set to eps = 0.25 and min_samples = 12.

(3) Mean filtering in x, y, z direction

Notwithstanding undergoing two antecedent steps and the filtration of the majority of background noise, the point cloud data intersperses residual noise. Being cognizant of the majority of remaining points from pedestrians, their characteristic motion applies to further filter out noise. Specifically, a mean filter is applied in the x, y, and z directions to each frame of point cloud data. The result is shown in Fig. 10. This averaging process helps to highlight the distinctive features of pedestrians in motion.

$$(x_{mean}, y_{mean}, z_{mean}) = (\frac{\sum_{i=1}^{n} x_i}{n}, \frac{\sum_{i=1}^{n} y_i}{n}, \frac{\sum_{i=1}^{n} z_i}{n}) \tag{1}$$

Remark: n is the rest point cloud data counts of a single frame of 50 superimposed frames.
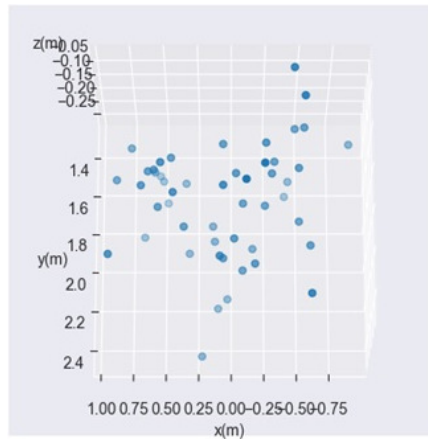

Fig. 10 A mean filtered data

By averaging all the points in this frame, the previously chaotic and noisy data will be converted into a point, which represents the position of the person in that frame. This also signifies the sequence information will be used later in LSTM deep learning. To better understand its temporal variation, Fig. 11 shows the position line chart having been averaged, and Fig. 12 shows the averaged variation of each frame.
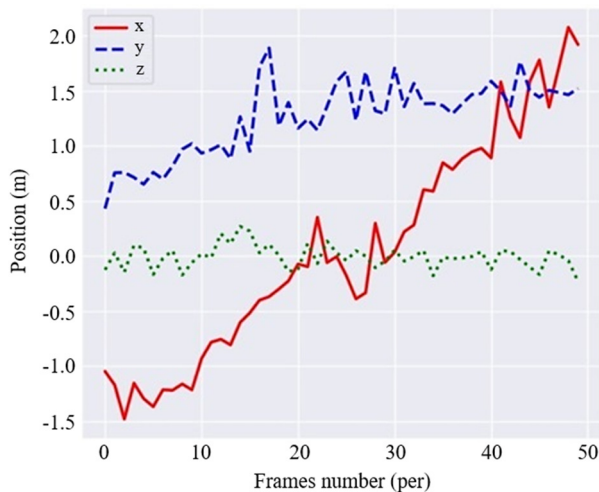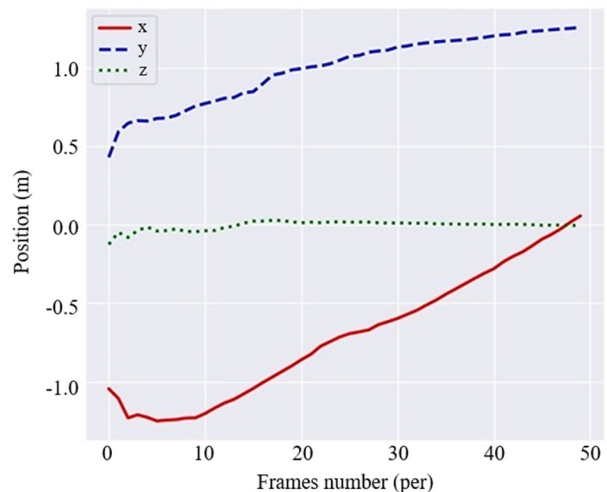

Fig. 11 Line chart of mean position

Fig. 12 Line chart of variation of the mean position

2.6. LSTM deep learning

After preprocessing the millimeter-wave radar v6 point cloud data through Doppler filtering, DBSCAN, and Mean filtering in x, y, and z directions to filter out all irrelevant noise and extract detailed data of each target object in each frame, the data is fed into an AI model. Since the point cloud data is sequential, an LSTM model is chosen for recognition. After multiple rounds of testing, the following neural network architecture is adopted.

(1) Model

As illustrated in Table 4, the first layer is constructed as LSTM, the third and fourth layers as linear layers, and the second layer in the middle with Relu (linear rectification function) to prevent gradient disappearance and gradient explosion. The input size of the LSTM input shape in the first layer is 4, which are the 4 feature values (x pos, y pos, z pos, doppler) of the millimeter wave radar detection [14]. The fifth layer is Softmax, which induces the sum of the two numbers in the matrix as 1, and the output is 2 for (1, 0) and (0, 1) respectively. If the matrix is close to (1, 0) for people walking to the right, the street light will turn on from left to right. If the result is close to (0, 1) for people walking to the left, the streetlights will turn on from right to left. If neither, the street light will not make any response.

Table 4 Layers of LSTM neural network

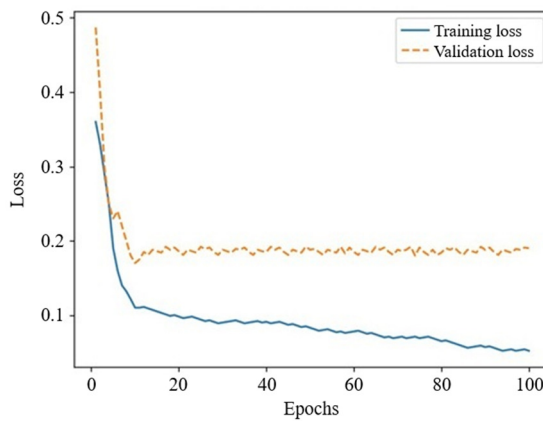| Layer	I/O	Configuration
LSTM	Input	(None, 50, 4)
	Output	(None, 50, 16)
Relu	Input	(None, 16)
	Output	(None, 16)
Linear	Input	(None, 16)
	Output	(None, 20)
Linear	Input	(None, 20)
	Output	(None, 2)
Softmax	Input	(None, 2)
	Output	(None, 2)

(2) Training

Fig. 13 LSTM training and validation loss

In this section, three people would pass 1.5 m-3 m in front of the millimeter wave radar for testing. Each person passed from left to right and right to left 110 times, with inconstant speed each time. A total of 660 records were collected. The learning rate is 0.001 and is inversely proportional to the change speed of the loss function. According to [15-16], 20% of the training data is used as validation data, and the data use "optimizer.zero_grad" to set the gradient to zero, i.e., changing the derivative of loss concerning weights to 0 before backpropagation. Then, "loss.backward" is called to start backpropagation, and, finally, "optimizer.step" is used to update the weights. A total of 100 epochs are performed, and the batch size is 20. The results are shown in Fig. 13 and Table 5.

Table 5 Confusion matrix of the proposed work

Class	Walk from left to right	Walk from right to left	Else	Total
Walk from left to right	324	1	5	330
Walk from right to left	0	327	3	330
Else	3	2	325	330

2.7. Streetlight controlling program

Fig. 14 is a flowchart for streetlight control. This chart purposively visualizes the steps of streetlight control to proffer the reader a comprehensive insight into the overall program. Initially, the system waits for a signal from the Jetson Nano, and if the signal is absent, no action will be taken. Then, the Jetson Nano transmits a signal indicating the correct direction of the illumination, which triggers the Key Process (Interrupt function) and the sleep function. When the system receives a signal from the Jetson Nano, the countdown variable is incremented by five, and when it reaches zero, the system continues to wait for the subsequent signal.
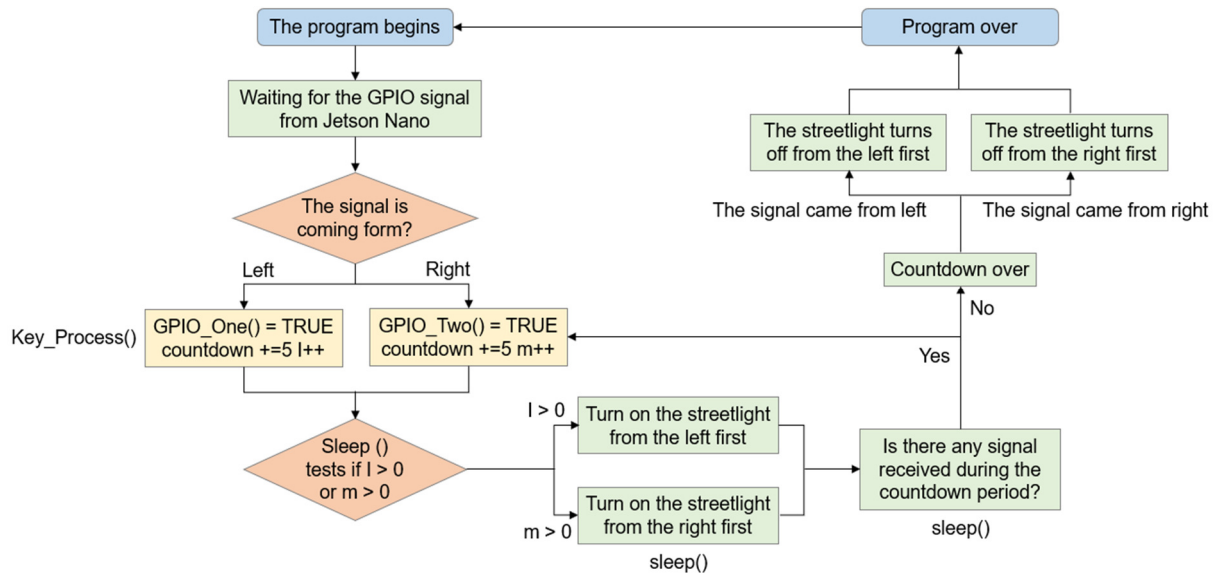

Fig. 14 The flowchart of the streetlight control program

The Key Process function is the interrupt function area, primarily used to send signals from the Jetson Nano to the HT32F52352 and determine the direction of the signal source. The sleep function is mainly used to turn on the LED light. Table 6 shows the PWM Duty Cycle variations that control the brightness of the streetlight, for readers to refer to.

Table 6 The duty cycle variation of PWM

The duty cycle when PWM is turned on and fully bright	100%
The variation in duty cycle during the PWM off period	75% => 50% => 25% => 0%

3. Result and Analysis

The model is first trained on the Windows 10 operating system using LSTM, and the ensuing trained model is deployed onto the NVIDIA Jetson Nano. The millimeter-wave radar is connected to perform object recognition, and the results are then transmitted to the HT32F52352 to control the streetlight switch. Assuming that the streetlights can halve their operating time every night, the analysis of power consumption and running costs of the proposed system signifies the importance of a smart lighting system. The results and analysis of the proposed system are demonstrated in the following sections.

3.1. Result

The results are shown in Figs. 15, 16, and 17. The trained LSTM model was saved using the torch.save and transferred as a path file to the Jetson Nano for loading. The weights in the file were used to compare with the data measured by the millimeter wave radar for recognition. The output after weight calculation is also a 1 × 2 matrix, where both values are between 0 and 1, and the sum of the two values is 1 (due to the softmax layer). The values between 0 and 1 represent probabilities, where 0 represents 0% and 1 represents 100%. Therefore, the 1 × 2 matrix represents the probabilities of two possibilities. If the left value is larger, such as [0.8, 0.2], the system determines that there is an 80% probability of moving to the right and a 20% probability of moving to the left. Thus, torch.max is used to select the higher value (higher probability) as the output. However, if both values do not exceed 0.6 (60%), such as [0.56, 0.44], the system will determine that there is no passing pedestrian, and the street light will not respond.
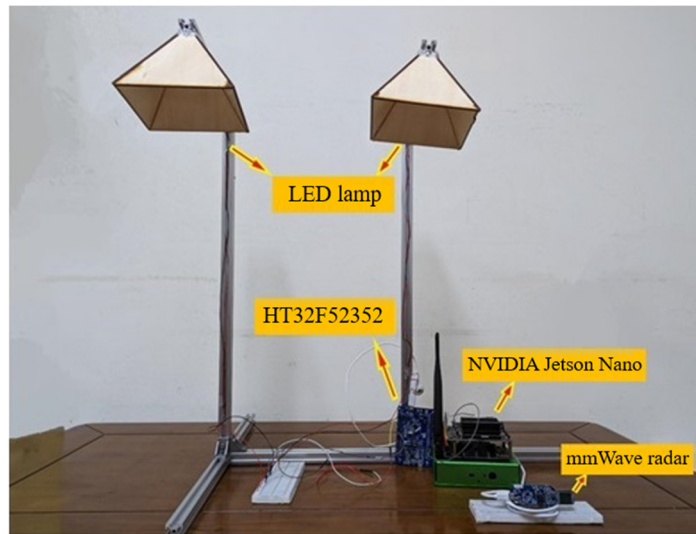

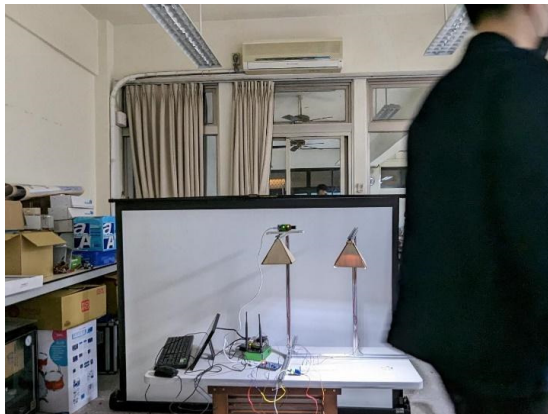Fig. 15 The smart streetlight energy-saving system

Fig. 16 The person passing by and the light turns on   Fig. 17 The light turns off after the person has gone for a while

3.2. Analysis

Through the use of millimeter-wave radar recognition, the system has achieved excellent results in recognizing people, walking direction, and identifying the target objects. This project is planned to install one system approximately every 500 meters. Streetlights are spaced at a distance of about 35 to 50 meters, i.e., at least 10 streetlights will be situated if this system is installed. The cost of this device is approximately 28,000 TWD. Each device can provide power for around 10 streetlights, which means the average cost per streetlight is 2,800 TWD. Table 7 presents the "device cost" of the "Smart Streetlight Energy Saving System" and common types of streetlights in Taiwan, along with their annual "energy consumption" and "electricity cost".

Table 7 The device cost and the annual energy consumption and electricity cost [17]

Name	Device cost	Power consumption	Energy consumption (11 Hours / Day × 1 year)	Electricity cost (1 year) (3.5 TWD / kWh)
Smart streetlight energy-saving system	28,000 TWD	10W	11 hours × 0.01 kW × 365 days × 10 units = 40 kWh	40 kWh × 3.5 = 140 TWD
LED	20,000 TWD × 10 units = 200,000 TWD	120W	11 hours × 0.12 kW × 365 days × 10 units = 4,818 kWh	4,818 kWh × 3.5 = 16,863 TWD
Sodium-vapor lamp	10,000 TWD × 10 units = 100,000 TWD	250W	11 hours × 0.25kW × 365 days × 10 units = 10,038 kWh	10,038 kWh × 3.5 = 35,133 TWD
Mercury-vapor lamp	5,000 TWD × 10 units = 500,000 TWD	400W	11 hours × 0.4kW × 365 days × 10 units = 16,060 kWh	16,060 kWh × 3.5 = 56,210 TWD

Due to the energy-saving effect of the "Smart Streetlight Energy Saving System", the operating time of the streetlights is reduced from 11 hours to 5 hours per day, as shown in Table 8. If the most power-consuming mercury-vapor lamp is considered, the "Smart Streetlight Energy Saving System" can save approximately 8,760 kWh annually. Even if the most energy-saving LED is used, based on the equipment cost of the proposed system, 2,628 kWh of electricity can be saved annually, which is equivalent to saving at least 9,058 TWD per year. The calculated investment payback period is 28,000 / 9,058 ≈ 3.09, and it will take approximately three years to recoup the equipment cost. Taiwan currently has about 1.6 million streetlights. Even if all streetlights in Taiwan were converted to LED, each group of 10 lights could save 2,628 kWh of electricity per year, and the total annual electricity savings for the entire country would be 2,628 kWh × 160,000 = 420,480 MWh.

Table 8 The energy-saving benefits and total savings after using the "Smart Streetlight Energy Saving System" [17]

Name	The system is in use or not	Energy consumption	Energy-saving benefits (1 years)	Electricity cost (3.5 TWD / kWh) # 140 TWD is the annual electricity cost for this System.	Total savings
LED	No	11 hours × 0.12 kW × 365 days × 10 units = 4,818 kWh	4,818 kWh - 2,190kWh = 2,628 kWh	4,818 kWh × 3.5 = 16,863 TWD	16,863 - 7,805 = 9,058 TWD
	Yes	5 hours × 0.12 kW × 365 days × 10 units = 2,190 kWh		2,190 kWh × 3.5 + 140 TWD = 7,805 TWD	
Sodium-vapor lamp	No	11 hours × 0.25 kW × 365 days × 10 units = 10,038 kWh	10,038 kWh - 4,563kWh = 5475 kWh	10,038 kWh × 3.5 = 35,133 TWD	35,133 - 16,109 = 19,024 TWD
	Yes	5 hours × 0.25 kW × 365 days × 10 units = 4,563 kWh		4,563 kWh × 3.5 + 140 TWD = 16109 TWD	
Mercury-vapor lamp	No	11 hours × 0.4 kW × 365 days × 10 units = 16,060 kWh	16,060 kWh - 7,300kWh = 8760 kWh	16,060 kWh × 3.5 = 56,210 TWD	56,210 - 25,690 = 30,520 TWD
	Yes	5 hours × 0.4 kW × 365 days × 10 units = 7,300kWh		7,300 kWh × 3.5 + 140 TWD = 25690 TWD	

4. Conclusion

This paper proposed a method that utilizes millimeter-wave radar combined with deep learning LSTM to identify targets for controlling the brightness of streetlights. The noticeable difference from conventional motion sensor lights lies in the proposed system approach of recognition rather than mere sensing. Therefore, the prerequisite of streetlight activation is the objects which are identified as pedestrians and vehicles. Such a function could conduce to the avoidability of frequent illumination of the streetlights, thereby achieving energy-saving benefits. While the system consumes power, the current architecture only consumes a maximum of 10W. The system is supposed to control traditional streetlights with approximately 10,200W, which is regarded as a mere 0.5% increase in total consumption. Therefore, ideally, the energy consumption caused

by the system can be considered negligible. Given the aforementioned findings and the pervasive installation of streetlights, the system has the potential to save approximately 420,480 MWh annually, which is equivalent to around 1.5 billion New Taiwan Dollars.

Acknowledgments

Conflicts of Interest

The authors declare no conflict of interest.

References

[1] G. Allen, "The Private Finance Initiative (PFI)," Economic Policy and Statistics Section, House of Commons Library, Research Paper 03/79, October 21, 2003.

[2] J. W. Selsky and B. Parker, "Cross-Sector Partnerships to Address Social Issues: Challenges to Theory and Practice," Journal of Management, vol. 31, no. 6, pp. 849-873, December 2005.

[3] S. Gupta, P. K. Rai, A. Kumar, P. K. Yalavarthy, and L. R. Cenkeramaddi, "Target Classification by mmWave FMCW Radars Using Machine Learning on Range-Angle Images," IEEE Sensors Journal, vol. 21, no. 18, pp. 19993-20001, September 2021.

[4] Y. P. Liao, F. K. Huang, Y. J. Xia, and H. Cheng, "Smart Speaker Based on Detection of Millimeter Wave," IEEE International Conference on Consumer Electronics-Taiwan, pp. 439-440, July 2022.

[5] Joybien Technologies Company Limited, DatasheetBM501MAC, 2022.

[6] R. Zhang and S. Cao, "Real-Time Human Motion Behavior Detection via CNN Using mmWave Radar," IEEE Sensors Letters, vol. 3, no. 2, article no. 3500104, February 2019.

[7] J. Jung, S. Lim, B. K. Kim, and S. Lee, "CNN-Based Driver Monitoring Using Millimeter-Wave Radar Sensor," IEEE Sensors Letters, vol. 5, no. 3, article no. 3500404, March 2021.

[8] X. Fan, C. Chen, Z. Huang, L. Tang, J. He, and Y. Jia, "Hand-gesture Recognition Based on Parallelism CNN and Multi-domain Representation for mmWave Radar," 8th International Conference on Signal and Image Processing, pp. 693-697, July 2023.

[9] J. Horne, "mmWave Radar, You Won't See It Coming," https://www.joshhorne.com/mmwave-radar-and-ambient-computing, February 01, 2022.

[10] C. Iovescu and S. Rao, "The Fundamentals of Millimeter Wave Radar Sensors," Texas Instruments, 2017.

[11] D. Salami, R. Hasibi, S. Palipana, P. Popovski, T. Michoel, and S. Sigg, "Tesla-Rapture: A Lightweight Gesture Recognition System from mmWave Radar Sparse Point Clouds," IEEE Transactions on Mobile Computing, vol. 22, no. 8, pp. 4946-4960, August 2023.

[12] B. Zhang, G. Xu, R. Zhou, H. Zhang, and W. Hong, "Multi-Channel Back-Projection Algorithm for mmWave Automotive MIMO SAR Imaging with Doppler-Division Multiplexing," IEEE Journal of Selected Topics in Signal Processing, vol. 17, no. 2, pp. 445-457, March 2023.

[13] M. Wang, F. Wang, C. Liu, M. Ai, G. Yan, and Q. Fu, "DBSCAN Clustering Algorithm of Millimeter Wave Radar Based on Multi Frame Joint," 4th International Conference on Intelligent Control, Measurement and Signal Processing, pp. 1049-1053, July 2022.

[14] S. Palipana, D. Salami, L. A. Leiva, and S. Sigg, "Pantomime: Mid-Air Gesture Recognition with Sparse Millimeter-Wave Radar Point Clouds," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 5, no. 1, article no. 27, March 2021.

[15] E. Stevens, L. Antiga, and T. Viehmann, Deep Learning with PyTorch, New York: Manning Publications, 2020.

[16] T. M. Breuel, "High Performance Text Recognition Using a Hybrid Convolutional-LSTM Implementation," 14th IAPR International Conference on Document Analysis and Recognition, pp. 11-16, November 2017.

[17] Y. J. Guo and Y. H. Chen, "Statistics and Analysis of Energy-Saving Application in LED Street Lighting for Newly Constructed Roads in Kaohsiung City," Construction Office Public Works Bureau Kaohsiung City Government, June 30, 2015. (In Chinese)