Occupancy Detection for HVAC Systems Using IoT Edge Computing and Vision-Based Image Processing

Tariq Akhtar
School of Architecture, Computing and Engineering
University of East London
London, United Kingdom
u2633833@uel.ac.uk

Shaheen Khatoon
School of Architecture, Computing and Engineering
University of East London
London, United Kingdom
s.khatoon@uel.ac.uk

Azhar Mahmood
School of Architecture, Computing and Engineering
University of East London
London, United Kingdom
a.mahmood3@uel.ac.uk

***Abstract*—Energy efficiency, particularly in Heating, Ventilation, and Air Conditioning (HVAC) systems, is a critical challenge in modern building management due to the increasing energy demands and environmental impacts. This paper focuses on developing optimized object detection models using machine vision for occupancy detection in office environments, aiming to improve HVAC efficiency. The primary objective is to compare three models—YOLOv8n, YOLOv9c, and YOLOv10n—against the Faster R-CNN baseline, emphasizing detection speed, computational efficiency, and small object detection. Data collection involved creating a custom dataset of 1,728 images from office environments, annotated with eight object classes, including persons and office devices. Preprocessing techniques such as grayscale conversion, image resizing, and augmentation improved the model's ability to detect objects under various conditions, including occlusion and varied camera angles. The models were evaluated based on mAP@50, mAP@50-95, and detection speed. YOLOv9c outperformed Faster R-CNN in speed and accuracy, achieving a mAP@50 of 88.0% and mAP@50-95 of 59.8%, making it the most balanced model. YOLOv8n demonstrated the fastest detection speed, ideal for real-time applications, while YOLOv10n, though less accurate, provided a strong trade-off between speed and precision. Despite these successes, challenges remain, particularly in small object detection and dataset size. Future work includes expanding the dataset to 100,000 images, improving detection of smaller objects, and integrating the object detection models into real-time HVAC control systems. Moreover, deployment on edge devices, transfer learning, and integration with Building Management Systems (BMS) for dynamic HVAC control represent promising areas for future research.**

***Index Terms*—Occupancy Detection, HVAC, YOLO, Energy Efficiency, Computer Vision**

I. Introduction

Energy efficiency is a significant challenge in today's world due to the growing global demand for energy and the associated environmental impact. Buildings are among the largest energy consumers in the U.S., with significant energy demand driven by their heating, ventilation, and air conditioning (HVAC) systems. In 2018, HVAC systems accounted for nearly 50% of the total electricity consumption in buildings across the country [1]. This high energy usage results from the need to maintain comfortable indoor climates, in varying weather conditions. Additionally, energy is wasted in large indoor areas due to centralized air conditioning, irrespective of the presence of occupants. With millions of residential, commercial, and industrial buildings relying on energy-intensive HVAC systems, this sector plays a major role in the overall energy footprint. Improving the efficiency of HVAC systems and adopting smarter building technologies could significantly reduce energy consumption, lower operational costs, and contribute to environmental sustainability by reducing greenhouse gas emissions. One way to reduce the energy consumption of HVAC is by controlling HVAC units via demand-response control systems. Machine vision and occupancy detection offer innovative solutions to reduce energy consumption in buildings, by optimizing HVAC system usage. By using edge technology equipped with cameras and machine vision technology, systems can detect real-time occupancy patterns in buildings, identifying when and where spaces are in use. This allows HVAC systems to adjust heating, cooling, and ventilation dynamically, supplying energy only to occupied areas and reducing waste in unoccupied ones. Accurate occupancy detection can be achieved by using machine vision [2]. RGB images provide rich visual data that can accurately identify occupants, differentiate between people and objects, and track movement within a space. This allows systems to make real-time adjustments to HVAC based on actual usage, reducing energy waste. Additionally, machine vision can work without requiring invasive or intrusive sensors on individuals, preserving privacy while maintaining functionality. Using the Internet of Things (IoT) with Edge Computing further enhances privacy, by allowing the images gathered to

979-8-3503-6720-1/24/$31.00 ©2024 IEEE
DOI 10.1109/UCC63386.2024.00074

be processed on-site rather than sending the raw data to a cloud server. This also reduces system latency and computing resource requirements. In this work, we will develop low-cost IoT-based solutions to optimize energy consumption in office environments by learning about environmental changes using sensors and image processing on edge computing devices to automate HVAC operations in real-time. However, this paper focuses on accurate occupancy detection using vision-based algorithms to estimate the number of people and other equipment in an indoor space. By acquiring reliable information on such objects, the next step is developing edge-based solutions for automatically adjusting the heating and cooling requirements of an indoor space while maintaining the thermal comfort of an environment. Such an edge- computing-based solution to energy conservation for the growing HVAC industry will help combat climate change and achieve millennium development goals. The rest of the paper is organized as follows. Section II contains the literature review. Section III explains the methodology from data collection to model evaluation. Section IV contains the results. Section V contains the conclusions and future work.

II. Related Work

Previous work on building occupancy estimation has used motion sensors [3,4,5], Infrared Proximity Sensors (PIR) [6], CO2 concentration sensors [7] and thermal images [8,9,10]. These methods had limitations such as latency, false detection and missing static objects or occupants [11]. Previous machine vision methods had limitations such as high computational loads, occlusion issues, small object detection, lighting issues, camera placement issues and system generalisation. [12,13]. This paper explores what data is available to create a model that can overcome these limitations.

III. Methodology

The methodology of this study includes data collection, preprocessing, model development, and evaluation to create and test optimized object detection models for occupancy detection.

A. Data Collection

A custom dataset was created to capture real-world office settings. Videos from sources like Pixabay provided diverse office layouts and lighting. Frames were extracted at 1 fps from 16 videos, yielding 1,728 images with eight object classes: person, cell phone, printer, mouse, computer, laptop, keyboard, and tablet. This ensured detection of occupants and heat-generating devices.

B. Data Pre-processing

Images were preprocessed using Roboflow for annotation, augmentation, and standardization. Key steps included:

- **Image Resizing**: Standardizing images to 640x640 pixels.
- **Cropping and Tiling**: Cropping each image by 25-75% and arranging into a 2x2 grid to emphasize areas of interest.
- **Grayscale Conversion**: Reducing computational load without affecting detection quality.
- **Augmentation**: Adding variety through flips, noise injection, and bounding box rotation.

C. Model Development

Three YOLO models (YOLOv8n, YOLOv9c, YOLOv10n) were developed and trained alongside Faster R-CNN as a baseline.

- **Faster R-CNN**: Used as a baseline model for comparison purposes. Faster R-CNN has a high accuracy but a slow detection time.
- **YOLOv8n**: Prioritized for speed with lightweight architecture and minimal parameters. Suited for edge device deployment and real time object detection.
- **YOLOv9c**: Focused on balancing speed and accuracy, making it ideal for scenarios requiring a balance between performance and computational load.
- **YOLOv10n**: Targeted at scenarios where higher speed is critical, though it sacrifices some detection accuracy.

Training Setup: Training took place on Google Colab using GPU resources, with models trained for 100 epochs (YOLO models) and 300 epochs (R-CNN).

D. Model Evaluation

Models were evaluated for precision, accuracy, speed, and computational efficiency.

- **mAP evaluation**: Evaluated at mAP@50 and mAP@50-95 for detection accuracy under various conditions.
- **Precision and Recall**: Precision assessed true positive accuracy, while recall indicated the capture of actual positives, critical for small/occluded objects.
- **F1-Score**: Balanced metric combining precision and recall.
- **Detection Speed**: Measured in ms/image to ensure suitability for real-time applications.
- **Computational Load**: Assessed model size, GPU memory use, and edge device compatibility for efficient deployment on low-power devices.

IV. Experimental Results

The experiment results highlight the performance of different object detection models, including Faster R-CNN and multiple versions of YOLO (YOLO V8n, YOLO V9c, and YOLO V10n), in detecting occupancy from office environments. These models were evaluated on a created dataset to measure their effectiveness in human detection under varying conditions.

A. Dataset Description

The dataset consists of 1728 images from various office settings, with many possible classes for object recognition. The images were extracted at 1fps from 16 videos which were found on free websites like Pixabay. Afterwards, the images were manually annotated, and 8 classes were defined. The 8 classes in this dataset are person, cell phone, printer,

mouse, computer, laptop, keyboard and tablet. The images then were resized to 640x640 followed by a static crop of 25%-75%. Greyscale was applied, and the images were arranged in a 2x2 grid (2 rows and 2 columns). After prepossessing and data augmentation, the dataset was split into training, validation and testing. The dataset was saved in COCO and YOLO annotation format to finally be trained on the selected models.Several object detection models such as Faster R-CNN and different versions of YOLO were implemented. The following subsection discusses the results of each model.

B. Model Evaluation

Faster R-CNN (Baseline): Fig.1 shows the training progression for the Faster R-CNN model. In this figure, the mean average precision (mAP@50) and mAP@50-95 metrics over the training epochs are displayed. The results indicate an mAP@50 of 87.4% and an mAP@50-95 of 60%. The graph shows a steady increase in precision during the early epochs, followed by a plateau in the later epochs, suggesting that the model converged as training progressed. Fig.2 shows the training performance of Faster R-CNN. It tracks loss functions such as train/box loss and train/classification loss, both of which, show a downward trend, indicating improvements in bounding box prediction and classification accuracy. Validation metrics fluctuate more but follow a downward trend, hinting at slight overfitting on unseen data. Fig.3 shows the inference on unseen data. The model accurately detected 3 occluded people and 1 laptop, but did not detect the other 4 people. Inference time was 200ms.

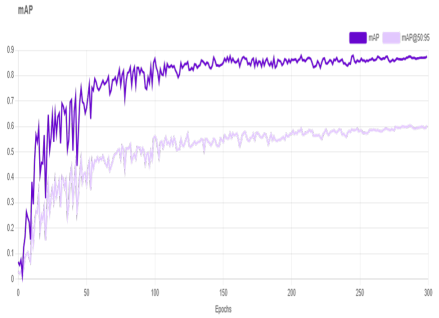

Fig. 1. Mean Average Precision (mAP) for Faster R-CNN

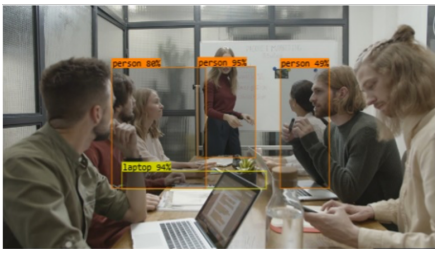

Fig. 3. Inference Faster R-CNN

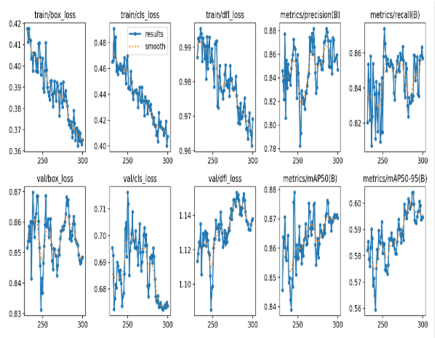

Fig. 2. Training Visualization for Faster R-CNN

YOLOv8n Model:

- mAP@50: 87.2%
- mAP@50-95: 59.4%

Fig.4 shows a sharp decline in training losses early on, with steady improvements in precision and recall throughout the training process. By around 100 epochs both mAP@50 and mAP@50-95 metrics stabilize, signalling model convergence. The precision-recall curve for YOLOv8n in Fig.5 shows varying performance across object classes. The cell phone class performed best with a precision of 0.988, while the tablet class had the lowest precision at 0.761, indicating the model's difficulty in distinguishing tablets from similar objects. Fig.7 shows the YOLO V8 Nano model's predictions on a test set. The model successfully detects and labels various objects such as "person," "laptop," "tablet," and "cell phone." Each object is enclosed in a bounding box with a confidence score. The model correctly identifies multiple instances of people and laptops, with confidence scores ranging from 0.6 to 1.0. For instance, it detects a "person" with 0.9 confidence in the centre and multiple "laptops" with confidence as high as 1.0. However, it seems to struggle with "tablet" and "cell phone" detections, with confidence scores as low as 0.3 to 0.5. This indicates the model is good at detecting larger, more distinctive objects but faces challenges with smaller or more similar-looking items like tablets and cell phones.

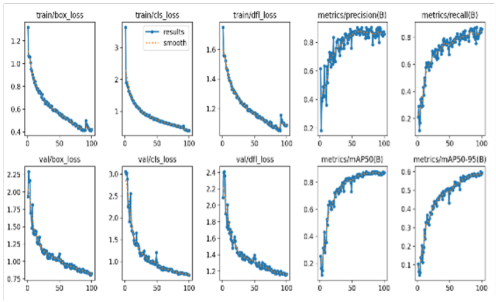

Fig. 4. Training Visualization for YOLO V8

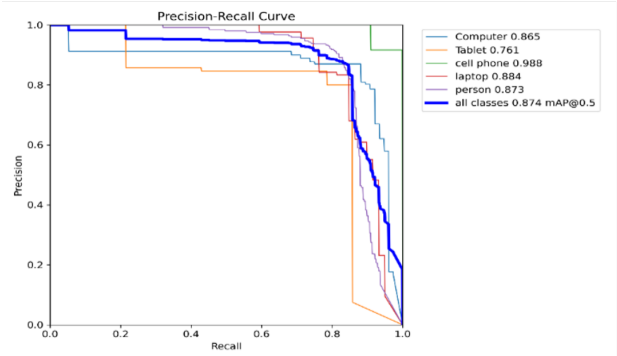

Fig. 5. Precision and Recall Curve YOLO V8 Nano

Fig. 6. YOLO V8 Nano Prediction on Test Data

YOLOv9c Model:

- mAP@50: 88.0%
- mAP@50-95: 59.8%

The training visualization of YOLOv9c in Fig.8 shows consistent improvements in training losses and evaluation metrics. The mAP@50 and mAP@50-95 metrics show significant growth over the epochs with precision nearing 0.88 and recall above 0.75, indicating strong performance in object detection. Fig.9, the precision-recall curve for YOLOv9c, indicates high precision for cell phone detection (0.968), while tablet detection is less precise at 0.798. The overall performance is strong across most object categories, with an average precision of 0.874 mAP@50. Fig.11 shows the predictions made by YOLOv9c on a test dataset. The model correctly identifies objects such as ”Computer,” ”Person,” ”Cell Phone,” and ”Laptop,” with confidence scores ranging from 0.8 to 1.0. For example, multiple ”Computer” instances are detected with high confidence (0.9), as well as ”Person” and ”Laptop” instances with confidence levels as high as 1.0 and 0.9. The ”Cell Phone” class is identified with confidence scores around 0.8, indicating reasonably accurate detection. Overall, YOLOv9c demonstrates strong detection capabilities, with consistently high confidence scores across most objects, though occasional low-confidence detections (e.g., ”Laptop” at 0.4) indicate the potential for further fine-tuning to improve recognition of specific items.

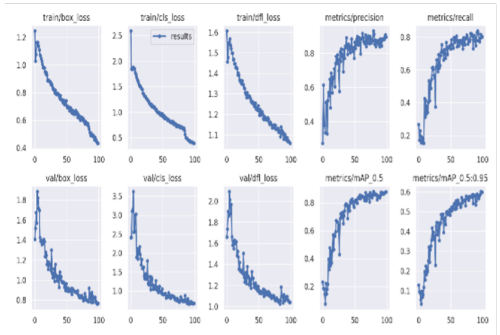

Fig. 7. Training Visualization of YOLOv9c

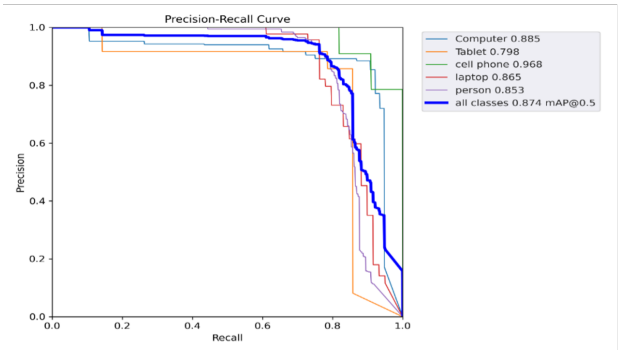

Fig. 8. Precision and Recall curve of YOLOv9c

Fig. 9. Prediction of YOLOv9c on Test Data

YOLOv10n Model:

- mAP@50: 81.6%
- mAP@50-95: 54.5%

The training visualization for YOLOv10n in Fig.12 demonstrates similar trends in loss reduction as the model progresses through training. Precision and recall improve consistently, though the performance does not match that of YOLOv9c, particularly in detecting smaller or occluded objects. The precision-recall curve for YOLOv10n in Fig.13 shows weaker performance for tablet detection, with a precision of 0.679. The model achieves better results for larger objects like persons and computers, with precision scores of 0.862 and 0.853, respectively. In Fig.15 YOLOv10n correctly detects and labels objects such as ”Computer,” ”Cell Phone,” and ”Person.” The confidence scores for these predictions range from 0.3 to 1.0. Notably, the model detects a ”Computer” with high confidence (up to 1.0 in several instances) and consistently identifies ”Person” with confidence ranging from 0.5 to 1.0. However, it struggles to identify the ”Cell Phone,” with confidence scores as low as 0.4. This suggests that while YOLOv10n is effective at detecting larger objects like ”Computer” and ”Person,” it faces challenges when detecting smaller or less distinct objects like ”Cell Phone,” leading to lower confidence scores in those instances.

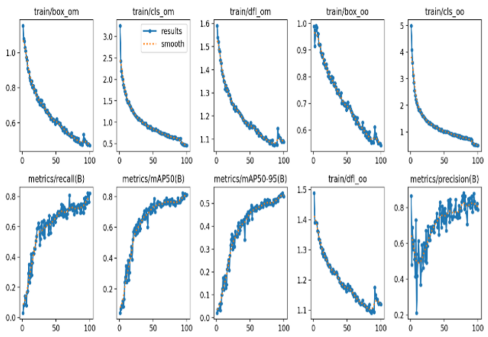

Fig. 10. Training Visualization of YOLOv10n

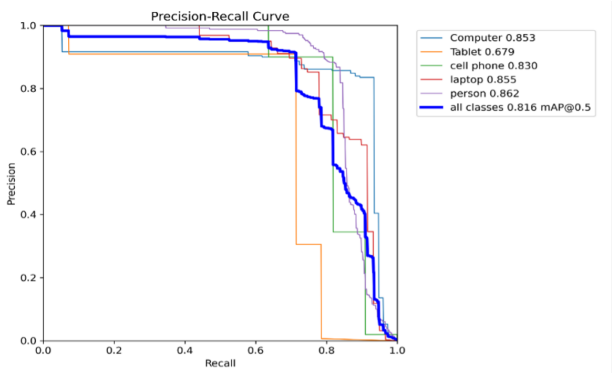

Fig. 11. Precision and Recall of YOLOv10n

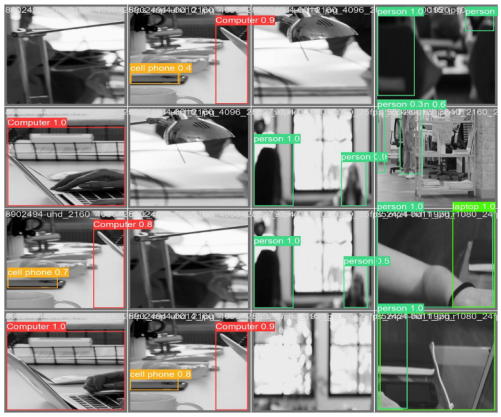

Fig. 12. Prediction for YOLOv10n on Test Data

C. Comparison of All Models

The performance of four models—Faster R-CNN, YOLO V8n, YOLO V9c, and YOLO V10n—was compared across various metrics, including mean average precision (mAP), recall, precision, F1 score, model parameters, and detection speed. The results are presented in Tables I, II, and III. As shown in Table I, the YOLO V9c model achieved the highest mAP@50 score of 88% closely followed by Faster R-CNN with 87.4% and YOLO V8n with 87.2%Ḣowever, YOLO V10n showed a lower performance, with an mAP@50 score of 81.6%Ḟor the more stringent mAP@50-95 metric, Faster R-CNN performed the best with 60% followed by YOLO V9c at 59.8% and YOLO V8n at 59.4%ẎOLO V10n trailed behind with 54.5%Ṫhese results suggest that while YOLO models, particularly YOLO V9c, excel at higher precision (mAP@50), Faster R-CNN provides more consistent performance across varying IoU thresholds (mAP@50-95). In terms of recall and

TABLE I
COMPARISON OF MAP OF ALL FOUR MODELS

Model	mAP@50	mAP@50-95
Faster R-CNN	87.4	60
YOLO V8n	87.2	59.4
YOLO V9c	88	59.8
YOLO V10n	81.6	54.5

precision (Table II), Faster R-CNN and YOLO V8n show identical scores, both achieving a recall of 0.86 and precision of 0.85, resulting in an F1 score of 0.855. YOLO V9c, while exhibiting the highest precision (0.89), had a lower recall (0.80), leading to a slightly lower F1 score of 0.84. YOLO V10n performed equally in both recall and precision, with a score of 0.81, resulting in a balanced F1 score of 0.81. These results suggest that YOLO V9c tends to favor precision over recall, while Faster R-CNN and YOLO V8n maintain a good balance between both metrics.

The number of parameters and detection time for each model are compared in Table III. Faster R-CNN, with 42 mil-

TABLE II
EVALUATION METRICS OF ALL FOUR MODELS

Model	Recall	Precision	F1 Score
Faster R-CNN	0.86	0.85	0.855
YOLO V8n	0.86	0.85	0.855
YOLO V9c	0.80	0.89	0.84
YOLO V10n	0.81	0.81	0.81

lion parameters, is by far the heaviest model, and its detection time ranges between 150-250 ms. In contrast, YOLO V8n, with only 7 million parameters, is the lightest model, achieving the fastest detection speed between 10-20 ms. YOLO V9c and YOLO V10n have moderate parameters, with 12 and 10 million, respectively, and their detection speeds are slightly slower than YOLO V8n, ranging from 15-30 ms.

TABLE III
COMPARISON OF PARAMETERS AND DETECTION SPEED

Model	Parameters (Million)	Detection Time (ms)
Faster R-CNN	42	150-250
YOLO V8n	7	10-20
YOLO V9c	12	15-25
YOLO V10n	10	15-30

Faster R-CNN stands out for its strong mAP@50-95 performance and consistent recall and precision, making it a robust choice when high accuracy across varying thresholds is needed. However, its large number of parameters and slower detection times make it less ideal for real-time applications like occupancy detection. YOLO V8n, while slightly behind in mAP and F1 scores, is the fastest model with the fewest parameters, making it highly efficient for real-time deployment. YOLO V9c provides a good balance, with the highest mAP@50 and precision scores, though its recall is slightly lower. YOLO V10n, while the least accurate in terms of mAP, still offers a reasonable tradeoff between speed and performance.

V. CONCLUSION

This research marks a significant step toward achieving a vision-based occupancy detection system for HVAC optimization. By developing edge-based models, it demonstrated that higher precision could be achieved compared to the baseline Faster R-CNN, particularly with YOLOv9c excelling in both detection speed and accuracy. However, several areas remain for improvement, such as expanding the dataset, incorporating GAN-based data augmentation, and further optimizing the models for edge devices. Addressing these challenges will be crucial in making the system more robust, enabling real-time occupancy detection that can efficiently contribute to energy-saving HVAC systems. This research marks a significant step toward achieving a vision-based occupancy detection system for HVAC optimization. By developing edge-based models, it demonstrated that higher precision could be achieved compared to the baseline Faster R-CNN, particularly with YOLOv9c excelling in both detection speed and accuracy. However, several areas remain for improvement, such as expanding the dataset, incorporating GAN-based data augmentation, and further optimizing the models for edge devices. Addressing these challenges will be crucial in making the system more robust, enabling real-time occupancy detection that can efficiently contribute to energy-saving HVAC systems.

REFERENCES

[1] S. Koebrich, E. I. Chen, T. Bowen, S. Forrester, and T. Tian, "2017 Renewable Energy Data Book: Including Data and Trends for Energy Storage and Electric Vehicles," National Renewable Energy Lab.(NREL), Golden, CO (United States)2019.

[2] 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.

[3] Y. Agarwal, B. Balaji, R. Gupta, J. Lyles, M. Wei, and T. Weng, "Occupancy-driven energy management for smart building automation," in Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building, 2010, pp. 1-6.

[4] K. Hashimoto, K. Morinaka, N. Yoshiike, C. Kawaguchi, and S. Matsueda, "People count system using multi-sensing application," in Proceedings of International Solid State Sensors and Actuators Conference (Transducers' 97), 1997, vol. 2, pp. 1291-1294: IEEE.

[5] J. Yun and S.-S. Lee, "Human movement detection and identification using pyroelectric infrared sensors," Sensors, vol. 14, no. 5, pp. 8057-8081, 2014

[6] K. Rastogi and D. Lohani, "IoT-based Indoor Occupancy Estimation Using Edge Computing," Procedia Computer Science, vol. 171, pp. 1943-1952, 2020.

[7] S. Wang, J. Burnett, and H. Chong, "Experimental validation of CO2-based occupancy detection for demand-controlled ventilation," Indoor and built environment, vol. 8, no. 6, pp. 377-391, 1999.

[8] 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.

[9] A. Gomez, F. Conti, and L. Benini, "Thermal image-based CNN's for ultra-low power people recognition," in Proceedings of the 15th ACM International Conference on Computing Frontiers, 2018, pp. 326-331.

[10] E. Griffiths, S. Assana, and K. Whitehouse, "Privacy-preserving image processing with binocular thermal cameras," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 1, no. 4, pp. 1-25, 2018.

[11] Rueda, J.C., Varas, A., Grimmer, J., Ponce, D. and Ruano, A.E., 2020. Deep learning-based occupancy detection system for energy management in smart building environments. Journal of Building Engineering, 31, p.101612.

[12] Sun, K., Zhang, X. and Bao, H., 2022. A robust real-time small object detection method for smart building management using YOLOv5. Sustainable Cities and Society, 76, p.103298.

[13] Hu, X., Li, Y., Wang, Z. and Zhao, H., 2023. Edge intelligence-based HVAC control system for smart buildings leveraging deep learning and IoT. Energy and Buildings, 278, p.112473.