Chapter 13

Environmental Sustainability Through Optimal Energy Consumption Using IoT-Based Edge-Computing and Image Processing

Shaheen Khatoon
University of East London, UK

Azhar Mahmood
https://orcid.org/0000-0002-1490-7389
University of East London, UK

Tariq Akhtar
https://orcid.org/0009-0004-8713-0007
University of East London, UK

Bintao Hu
https://orcid.org/0000-0003-4821-0448
Xi'an Jiaotong-Liverpool University, China

ABSTRACT

This chapter explored various techniques and modelling methodologies designed to estimate occupancy level in indoor environments. It also introduces an innovative image-based occupancy detection system that leverages edge computing and machine vision to accurately detect and classify occupants and other objects in an indoor environment which requires a certain thermal comfort level. By enabling real-time adjustments to HVAC operations based on actual occupancy, it can significantly reduce unnecessary energy consumption in unoccupied areas, thus improving overall energy management. The integration of edge computing allows for local data processing, which not only minimizes the computational load on centralized servers but also addresses privacy concerns by reducing the need for external data transmission. This is particularly important in environments where sensitive information about occupants

DOI: 10.4018/979-8-3693-5748-4.ch013

may be captured. A case study is presented in the end to demonstrate and examines the performance of several object detection models in the context of academic office occupancy detection.

CHAPTER OVERVIEW

Energy efficiency in buildings is a crucial component in tackling global sustainability challenges, particularly as Heating, Ventilation, and Air Conditioning (HVAC) systems are among the largest consumers of electricity. These systems often account for nearly half of a building's total energy consumption, making their optimization essential for reducing energy waste and minimizing the environmental impact of buildings. Occupancy-based control (OBC) systems offer innovative solutions to reduce energy consumption in buildings via demand-response control mechanism. By using sensors (e.g., temperature, humidity, light sensors, as well as specialized sensors like PIR sensors, smart meters, and thermal cameras) such systems can perceive environmental changes and estimate buildings' occupancy for adapting heating, ventilation and cooling in real time. Traditional systems have combined environmental sensors with intelligent techniques, such as machine learning and statistical modeling, to establish relationships between environmental parameters and occupancy levels.

This chapter explored various techniques and modelling methodologies designed to estimate occupancy level in indoor environments. It also introduces an innovative image-based occupancy detection system that leverages edge computing and machine vision to accurately detect and classify occupants and other objects in an indoor environment which requires a certain thermal comfort level. By enabling real-time adjustments to HVAC operations based on actual occupancy, it can significantly reduce unnecessary energy consumption in unoccupied areas, thus improving overall energy management. The integration of edge computing allows for local data processing, which not only minimizes the computational load on centralized servers but also addresses privacy concerns by reducing the need for external data transmission. This is particularly important in environments where sensitive information about occupants may be captured.

A case study is presented to demonstrate and examines the performance of several object detection models in the context of academic office occupancy detection. A custom dataset is created, consisting of 1,728 images from office settings, manually annotated to include eight object classes: person, cell phone, printer, mouse, computer, laptop, keyboard, and tablet. The models trained on this dataset include YOLOv8n, YOLOv9c, and YOLOv10n, as well as baseline models such as Faster R-CNN. To overcome challenges such as high computational loads, occlusion, and lighting inconsistencies, the dataset is thoroughly prepared, augmented, and annotated to enhance model training. By analyzing the strengths and weaknesses of these models, the study highlights their capabilities in accurately detecting occupants and while ensuring robust performance in real-world scenarios. Ultimately, this framework aims to provide a scalable and privacy-conscious solution for managing energy use in buildings, contributing to the broader objective of environmental sustainability.

1. INTRODUCTION

Energy efficiency is a significant challenge in today's world due to the growing global demand for energy and the environmental impacts associated with its production and consumption. As economies expand and populations increase, more energy is required to power industries, transportation, homes, and technology. Additionally, outdated infrastructure and resistance to adopting new, energy-efficient technologies further exacerbate the problem. 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(Koebrich et al., 2019). This high energy usage results from the need to maintain comfortable indoor climates, especially in varying weather conditions. Also, a lot of 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 efficiently controlling HVAC units via demand-response control mechanisms. Occupancy-based control (OBC) offer innovative solutions to reduce energy consumption in buildings, by optimizing HVAC system usage. By using sensors and cameras equipped with computer vision technology, buildings can detect real-time occupancy patterns, 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. Occupancy detection can be integrated with advanced algorithms to predict occupancy trends and optimize energy use further, ensuring that systems operate only when necessary, lowering energy consumption and enhancing the overall energy efficiency of buildings.

There have been continuing advances in sensing technology for monitoring the interaction between the building system and occupants, e.g., motion sensors (Yun and Lee, 2014, Hashimoto et al., 1997, Agarwal et al., 2010), Infrared Proximity Sensors (PIR) (Rastogi and Lohani, 2020), CO2 concentration sensors (Wang et al., 1999), and thermal images (Beltran et al., 2013, Gomez et al., 2018, Griffiths et al., 2018). The motion and proximity sensor has limited range and fails to count people with increasing objects (Wahl et al., 2012). In addition, the detection likely fails for the static objects. CO2 concentration-based solution has a relatively poor real-time performance, and the measurement precision is likely to be significantly reduced when the doors or windows are opened. Thermal cameras usually cannot recognize the characteristics of the detected object (Anjomshoaa et al., 2018).

Video based occupancy detection methods are emerging as effective approaches for recognizing activity in buildings (Petersen et al., 2016). These methods utilize computer vision technology to analyze video, extracting information such as number of people, their location and even behaviors. Typically, video analysis involves detecting features like the head, face, body contour, or movement. However, these methods can face challenges, particularly in environments with many obstacles or when only parts of a person's body are visible. Detection methods that rely on a single body characteristic often struggle in real-world applications. Additionally, in terms of cost, users prefer to utilize existing surveillance footage rather than installing new cameras. A key challenge with video-based occupancy detection methods is the concern over privacy, as these systems capture sensitive personal data, raising issues around data protection, unauthorized access, and the need for anonymization and secure storage.

This chapter presents an image-based occupancy detection system utilizing edge computing, specifically designed to automatically monitor indoor spaces for energy-saving purposes. Using machine vision and RGB images for indoor occupancy detection offers several advantages, making it an effective solution for optimizing energy use and enhancing building management. 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. By leveraging edge computing, the system processes data locally, ensuring faster response times and enhanced privacy protection by minimizing the need for data transmission to external servers. The system efficiently detects and tracks occupancy in real time, enabling dynamic adjustments to energy consumption of HVAC based on actual usage, thus optimizing energy efficiency in typical indoor environments such as offices, homes, and commercial buildings, enhancing overall building management and sustainability.

There are numerous machine vision models that are ready for deployment in edge computing environments, suitable for efficient occupancy detection once properly trained. In this chapter, a case study is conducted to examine the performance of various object detection models specifically for occupancy detection in office environment while also aiming to ensure the models' universal applicability across various environments. Previous machine vision approaches have faced several limitations, including high computational loads, occlusion challenges, difficulties in detecting small objects, lighting inconsistencies, camera placement issues, and difficulties in system generalization. To address these challenges, our main objectives is preparing a diverse set of data, accurately annotating this data to enhance model training, and training multiple machine vision models, including YOLOv8n, YOLOv9c, and YOLOv10n. We also compare their performance against baseline models like Faster R-CNN. This comprehensive approach aims to create robust, efficient, and adaptable occupancy detection systems in the next phase that can effectively manage energy consumption and improve building management in real-world scenarios.

2. OCCUPANCY DETECTION TECHNIQUES AND METHODS

Occupancy information in a building is critical in terms of indoor environmental quality, energy consumption and building energy simulation. In this section, we review the techniques and methods along modelling methodologies in occupancy estimation in buildings. Also discuss the advantages and limitations of these approaches proposed in recent past, providing insights for their potential application in Occupant-Based Control (OBC) systems.

2.1 Sensor-based Occupancy Detection Techniques and Methods

Most occupancy detection systems rely on the deployment of multiple environmental sensors, such as carbon dioxide (CO2), temperature, humidity, and light sensors, as well as specialized sensors like PIR sensors, smart meters, and thermal cameras (Rueda et al., 2020). Existing studies have combined these technologies with intelligent techniques, such as machine learning and statistical modeling, to establish relationships between environmental parameters and occupancy levels. For instance, Hobson et al. (2019) use Wi-Fi access point, CO2 sensors, PIR motion detector to collect data stream from an academic office environment to develop occupancy count estimation for HVAC control purpose. Various

ML based techniques such as linear regression and ANN are used to estimate occupancy information. In their case study linear regression models outperforms other methods. The study found Wi-Fi enabled device counts are useful for occupancy-count estimations achieving a mean R^2 of 80.1–83.0% when compared to ground truth counts during occupied hours.

Similarly, Dorokhova et al.(2020) used rule-based based system for occupancy forecasting using sensors such as Temperature, CO2, Humidity and Luminosity. It explores both supervised models with an accuracy up to 98.3 on cross validation and unsupervised algorithms with a cross-validation accuracy of 96.7 for predicting occupancy with high accuracy. In a case study of a mid-sized building in Portugal, a potential energy saving of 15.4% is demonstrated. The methodology includes extensive feature engineering to enhance data pre-processing and employs machine learning models like SVM, ANN, and LSTM for occupancy forecasting. The highest performing algorithms is LSTM for unsupervised and Feedforward ANN for supervised algorithms. Similarly, (Jiang et al., 2016) developed an indoor occupancy estimator to estimate real-time occupancy based on CO2 measurements, using a dynamic model of occupancy levels. The proposed Feature Scaled Extreme Learning Machine (FS-ELM), an improved version of the standard ELM, is introduced for this purpose. Since CO2 data often contains significant spikes, pre-smoothing the data is found to significantly enhance estimation accuracy. Given that real-time globally smoothed data isn't always available, the study proposed using locally smoothed data instead. The occupancy estimator, tested in an office setting with 24 cubicles and 11 open seats, achieved an accuracy of up to 94% with a tolerance of four occupants.

Dong and Lam (2014a) has presented a methodology reduce energy consumption based on prediction of occupant behaviour patterns and local weather conditions using CO2, acoustics, motion and lighting as input features. Adaptive Gaussian Process, Hidden Markov Model, Episode Discovery and Semi-Markov Model are modified and implemented into this study. Similarly, (Sayed et al.) developed an efficient and non-intrusive system for detecting occupancy in indoor spaces. The proposed method utilized an environmental sensing board to collect ambient data, which included temperature, humidity, pressure, light level, motion, sound, and CO2 levels. This data is then processed using a DL model, specifically a CNN, deployed on an edge device to ensure low-cost computing and enhanced data security. The accuracy achieved in this study is about 99.76%.

Abdel-Razek et al.(2022) used thermal comfort indices to estimate whether a room is occupied or empty. Data from fluctuations in light, CO2, and humidity levels are analyzed to assess the reliability of occupancy prediction. Additionally, kNN, ANNs, and DTs are applied as classification techniques. The results indicated that the kNN model outperformed both DT and ANN, achieving an accuracy of 99.50%.

Kou et al. (2021) investigated residential demand response (DR) implementations based on HVAC, focusing on effective control algorithms to coordinate the operating schedules of multiple HVAC devices. They developed both model-based and data-driven HVAC control strategies aimed at minimizing customers' electricity costs, discomfort, and utility-level load violations, using input data such as outdoor temperature (weather forecasts), indoor temperature, and non-HVAC electrical consumption. The model-based approach employed a thermal resistance-capacitance (RC) model and a distributed solution algorithm to establish optimal day-ahead HVAC operation schedules, achieving a 22% cost savings compared to the data-driven approach. In contrast, the data-driven approach, utilizing neural networks to interact with the environment during training for online decision-making, did not require outdoor temperature forecasts and is 46 times faster in computational speed, although it incurred higher electricity and discomfort costs.

2.2 Edge and Machine Learning Methods

There are studies using similar environmental sensors technologies as discussed in section 2.1, however, the deployment is conducted on the edge node. For instance, Rastogi et al. (2020) developed a methodology for estimating indoor occupancy using Linear and Quantile (QR) algorithms for indoor environment using CO2, temperature, relative humidity, and motion levels. Since the data generated by the setup is vast, sending it all to the cloud for processing may result in delayed prediction; hence, the suggested models are designed to be executed on an edge device.

In these studies, Zemouri et al. (Zemouri et al., 2019, Zemouri et al., 2018) addressed the effectiveness of machine learning algorithms used to predict human occupancy in closed office areas using temperature and humidity data as occupancy predictors. They employed a Raspberry Pi as an edge device to perform real-time occupancy detection. The findings revealed that kNN outperformed the other algorithms in all performance measures.

2.3 Vision-Based Techniques and Methods

Since mostly studies focused on sensing occupancy information through the count of occupants and their distribution in a certain environment, there is limited research and its material on sensing the actual objects which requires certain thermal comfort level. This is necessary to allow HAVC to dynamically adjust the heating and cooling requirements. Computer vision and AI- based occupancy detection solutions are emerging as effective approaches for recognizing activity in buildings using that can be implemented into building HVAC systems for higher accuracy monitoring and control. For Instance, Hu et al. (2022) presented a deep-learning-based approach for building occupancy detection using CCTV cameras. The method involves feature extraction through a deep convolutional neural network that constructs feature pyramids and a three-stage detection process using sequential detectors with increasing Intersection over Union (IoU) thresholds. Data is gathered by recording CCTV videos from a university building over five weeks, resulting in nearly 10,000 labeled images. Experimental results demonstrated that the proposed model achieved high detection accuracy with a Precision of 89.7%, a Recall of 89.2%, and an overall mean average precision (mAP) of 44.5%. The study highlights the effectiveness of this approach for real-time occupancy detection in complex indoor environments.

Tien et al.(2020) used thermal Cameras for detecting and predicting occupancy heat emissions to enable demand-driven control solutions in building energy management systems (BEMS). Developed a convolutional neural network (CNN) which detect and classify occupant activities from images such as sitting, standing, walking, and napping, generating real-time occupancy heat emission profiles that can optimize HVAC systems. Data is gathered by using images from online sources and captured images of various occupant activities in office spaces, with real-time live detection tests conducted in an office space in Nottingham, UK. The study achieved an average detection accuracy of 80.62%, with the highest detection accuracy being 89.39% for static images and about 93.7% for occupancy detection. This approach showed potential for better managing building energy loads and improving indoor environmental quality by accurately detecting occupant activities and adjusting HVAC systems accordingly.

Similarly, Wang et al.(2023) proposed indoor occupancy detection system using an advanced YOLOv5 model. The method involves improving the YOLOv5 algorithm with a decoupled prediction head to better manage classification and regression tasks separately, which enhances the detection accuracy. A self created dataset is prepared with various indoor scenes, and the training involved significant image preprocessing, resizing, and categorizing images into classes such as sitting, standing, walking, napping, and none. The machine learning algorithm used in this study is a modified version of YOLOv5, specifically the DFV-YOLOv5. The improved YOLOv5 model achieved high accuracy, with the best value being 86.3% for one of the key performance indicators. This result demonstrates significant improvements over other baseline models used for comparison in the study.

Lee et al., (2018) presented a method for counting people using a stereo camera mounted on the NVIDIA Jetson TX2 to improve accuracy in crowded environments by addressing occlusion issues. The process involves camera calibration, stereo matching using the Semi-Global Matching algorithm, background subtraction with a Gaussian Mixture Model (GMM), and Kalman filter-based tracking. Data is gathered using IMX 185 cameras installed at heights of 3 to 5 meters, recording 23 high-definition video sequences. The method achieved over 95% accuracy in most scenarios, with an overall accuracy of 98.95%, and demonstrated real-time processing capabilities with HD resolution at 12.5 frames per second, highlighting its effectiveness for surveillance and crowd monitoring.

Likewise, Sun et al. (2022) aims to enhance indoor occupancy measurement accuracy by combining motion detection and static estimation techniques. The method involves a four-stage algorithm: occupancy detection, static estimation using Fully Convolutional Head Detector (FCHD), motion detection at room entrances using GMM and Camshift tracker, and a fusion estimation method that utilizes the Kalman filter and Occupancy Frequency Histogram to integrate results from the two techniques. Data is gathered using two USB cameras installed at the room entrance and interior, capturing video data over two days. The motion detection events and static head detections are processed using Nvidia Jetson Nano, employing OpenCV for computer vision tasks and Pytorch for deep learning computations. The machine learning algorithms used are FCHD for head detection, GMM for background subtraction, Camshift tracker for motion tracking, and Kalman filter for fusing motion detection and static estimation results. The study achieved an occupancy detection accuracy of 97.8%, with an occupancy estimation score ranging from 78.52% to 79.18% and a mean square error (MSE) between 0.21 and 0.23, demonstrating high accuracy and efficiency in real-world applications.

Simliary, Paidi et al. (2020) used thermal cameras to acquire vehicle occupancy information in an open parking lot using deep learning techniques. Frames from these videos are extracted and manually labeled due to the lack of pre-labeled thermal images. Multiple deep learning networks, including Yolo, Yolo-conv, GoogleNet, ResNet18, and ResNet50, are evaluated for vehicle detection. Data is gathered using an Axis Q1942-E thermal camera installed on a two-storey building, capturing videos in different weather conditions such as snow, rain, darkness, and brightness. The videos are stored locally, and frames representing diverse conditions are manually labeled to prepare the dataset. The machine learning algorithms used include Yolo, Yolo-conv, GoogleNet, ResNet18, and ResNet50. These networks are tested for their efficiency in detecting vehicles in thermal images. Among these, ResNet18 performed the best, achieving an average precision of 96.16% and a log-average miss rate of 19.40. The study highlights the effectiveness of using thermal cameras and deep learning for real-time vehicle occupancy detection, suitable for varying illumination and environmental conditions.

Likewise, (Yu et al., 2023) presented YOLOv5s model for face mask recognition in heterogeneous IoT environments. The proposed method includes embedding a Coordinate Attention (CA) mechanism, integrating a Bidirectional Feature Pyramid Network (BiFPN) block, adding an Adaptive Spatial Feature Fusion (ASFF) layer, and merging Scaled Intersection over Union (SIoU) to improve detection accuracy and computational efficiency. Data is gathered from diverse sources, including the AIZOO face mask dataset, an internet-sourced dataset, and a self-built dataset incorporating photos taken on campus merged with the Real-World Masked Face Dataset (RMFD). The study employed the YOLOv5 model, enhanced with the aforementioned components to address specific challenges in face mask detection across different computing platforms. The improved YOLOv5s model achieved superior accuracy compared to baseline models, with a detection accuracy of up to 95.4% on heterogeneous IoT platforms. This indicates a significant improvement in face mask recognition, making it suitable for real-time applications in various environments.

3. NOVELTY AND GAPS IN KNOWLEDGE

Most of the existing studies on building occupancy estimation has used motion sensor (Yun and Lee, 2014, Hashimoto et al., 1997, Agarwal et al., 2010, Hobson et al., 2019), Infrared Proximity Sensors (PIR) (Rastogi and Lohani, 2020, Hobson et al., 2019), CO2 concentration sensors (Wang et al., 1999, Dorokhova et al., 2020), and thermal images (Beltran et al., 2013, Gomez et al., 2018, Griffiths et al., 2018). The motion and proximity sensor has limited range and fails to count people with increasing objects (Wahl et al., 2012). In addition, the detection likely fails for the static objects. CO2 concentration-based solution has a relatively poor real-time performance, and the measurement precision is likely to be significantly reduced when the doors or windows are opened. Thermal cameras usually cannot recognize the characteristics of the detected object (Anjomshoaa et al., 2018). Furthermore, all these methods are effective in determining whether a room is occupied but are unable to accurately estimate the exact number of occupants. Additionally, they are sensitive to environmental factors such as airflow and solar radiation. Environmental sensors like carbon dioxide and temperature sensors also have limitations in real-time detection, as their performance can be compromised by open windows and doors.

While vision-based methods (Yu et al., 2023, Wang et al., 2023, Tien et al., 2020, Paidi et al., 2020) for detecting the number and locations of occupants and recognizing their activities or behaviors show promise, they face significant challenges. Identifying individuals in complex academic office environments, where obstacles like furniture, equipment, and partitions are common, is difficult, especially, detecting occupant activities when parts of the body are obscured or occlusion issues in densely populated areas. Moreover, these methods raise serious concerns about data privacy. Many existing studies focus primarily on improving the performance and accuracy of deep learning models for human presence detection and activity classification. However, less attention has been given to addressing the issues of computational load and privacy protection. This leaves a gap in developing solutions that minimize unnecessary computational burdens and safeguard privacy.

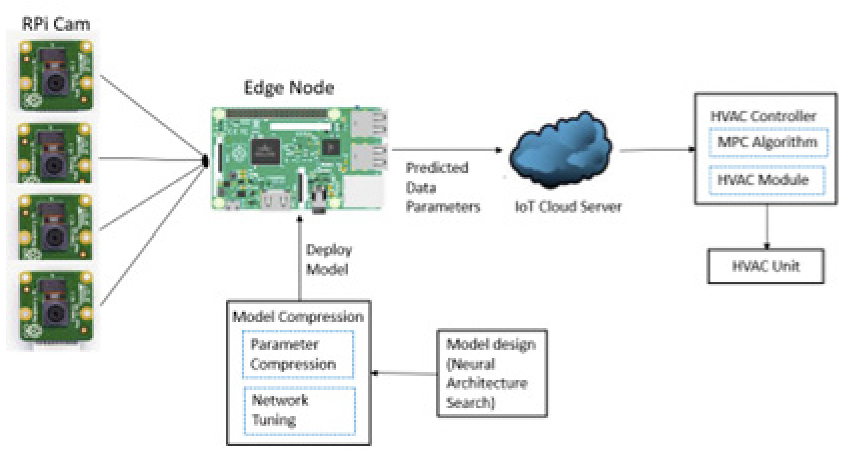
Further exploration is necessary to refine these methods, addressing both privacy concerns and the computational demands of real-time processing. In response to these challenges, we propose a generalized edge-based framework that aims to reduce both privacy risks and computational load, making the system more efficient and secure. This work focous on an image-based deep learning framework for occupancy detection in typical indoor spaces, particularly for energy-saving purposes. The proposed end-to-end edge-based architecture utilizes RGB images from video streams along with edge-optimized deep learning models, to classify indoor objects into distinct categories. Since various objects, beyond people, also require specific thermal comfort conditions, the goal is to detect all objects that require such comfort such as people, computer, laptops, servers, etc. This will allow for the automated control of HVAC systems, optimizing energy use by ensuring that thermal comfort is maintained only where necessary.

By using edge-based processing, the system ensures that sensitive data is handled locally, reducing privacy risks associated with transmitting images to cloud servers. Additionally, this approach minimizes computational overhead by leveraging lightweight models and localized processing, making it ideal for real-time applications in energy-efficient building management.

4. DESIGN OF THE IOT-ENABLED OCCUPANCY-BASED HVAC CONTROL SYSTEM

The overall architecture of the IoT-based control system is shown in Figure. 1. It consists of a set of RPi Cameras that streams images to an edge node. We plan to set up four RPi Cam to capture the room from different angles. RPi Cam will be configured with the edge device to transmit image stream and run the DNNs module to predict a room occupancy in terms of people and other objects that require certain thermal comfort and their count. The analysis results will be sent to the IoT cloud. The HVAC control unit will deploy Model Predictive Control (MPC) to communicate with the IoT cloud server to receive occupancy prediction results and run the MPC algorithm. The HVAC modules modify the temperature of the HVAC unit according to the decisions taken by the MPC algorithm.

Figure 1. The overall architecture of the proposed IoT-based solution



To implement embedded intelligence at the edge device following research questions will be addressed:

1. **What type of sensing technology is available for object identification in an indoor environment?**

Raspberry Pi Cameras (RPi Cam) for image streaming from an indoor space with high accuracy will be used. RPi Cam supports MIPI-CSI designed to use low power, smaller size, faster bandwidth, higher resolution and reduced latency. These features confirm to the resource constrained specification of the edge node.

2. **What type of edge technology can be used to deploy intelligent solutions?**

Previously researchers have used cloud-based systems for streaming raw data from an end device to the centralized servers or IoT cloud for analysis and send the results back to the end device (Ryu and Moon, 2016, Dong and Lam, 2014b, Carli et al., 2020). With the availability of low-cost and low-power IoT devices with edge computing features, it is possible to run advanced artificial intelligence algorithms at the edge where data originates. In the line of edge computing, previous studies have mainly designed edge nodes using few Micro Controller Units (MCUs) boards aiming to run AI algorithms at the local node (Metwaly et al., 2019). However, the configuration of the sensing device with the edge node is complex and needs manual calibration. Furthermore, cloud storage is required for intensive image computation. The adoption of Edge AI has led to developing specialized devices capable of performing AI inferencing efficiently, e.g., Nvidia Jetson Nano[1], Google Coral[2], and AWS DeepLens[3]. In the proposed edge computing architecture, the RPi Cam streams the data to an edge instead of uploading it directly to the cloud. The RPi Cam will be configured with the edge device using the NVIDIA Jetson Nano snapshot board. The Snapshot is the edge AI video capture device designed to run multiple neural networks in parallel for image classifications, object detection, and speech processing applications. Furthermore, the accompanying AGX Xavier Developer Kit provides tools and libraries to develop Edge AI applications. One device can support up to four RPi Cam connected via Wi-Fi or HDMI. Therefore, the snapshot board is more feasible and ready to use as an edge node to deploy deep learning-based occupancy estimation models.

3. **What are the most appropriate deep learning models that can be embedded at edge nodes for perceiving real-time environmental changes?**

We will develop and train multiple deep-learning models to detect occupancy and perceive environmental changes in real-time. We will explore various deep-learning-based models such as Feedforward Neural Networks (Moons et al., 2019), Convolutional Neural Networks (CNNs) (Gomez et al., 2018), Recurrent Neural Networks (RNNs) (Chung et al., 2014), and YOLO for image processing and object detections. At first, we will use GPU instances to train and test these models for realizing the proof of concept. Then we will use a tiny version of the most accurate model through model compression and network tuning so that the model can be deployed on the resource constraints microcontroller. To train any Neural Network, a large set of tagged training data is required. We plan to collect a dataset targeted at object recognition in the context of any office/classroom and tag data using Amazon Mechenical Turk[4], a crowdsourcing service that can be used to tag a huge amount of data.

5. CASE STUDY FOR OCCUPANCY DETECTION

In order to assess the performance of deep learning models, a case study is conducted, specifically for occupancy detection in an academic office environment. Previous machine vision approaches have faced several limitations, including high computational loads, occlusion challenges, difficulties in detecting small objects, lighting inconsistencies, camera placement issues, and difficulties in system generalization. To address these challenges, our main objectives is preparing a diverse set of data including people and other categories which requires certain thermal comfort, accurately annotating this data to enhance model training, and training multiple machine vision models, including YOLOv8n, YOLOv9c, and YOLOv10n and comparing their accuracy with baseline model, Faster R-CNN. This comprehensive approach aims to create robust, efficient, and adaptable occupancy detection systems in the next phase that can effectively manage energy consumption and improve building management in real-world scenarios. The dataset is prepared using Roboflow, which allowed us to efficiently create dataset by merging multiple office settings into a single cohesive dataset. For our purposes, the data needs to be annotated—a task easily accomplished with Roboflow. Additionally, Roboflow offers various functions for preprocessing the data.

5.1 The Occupancy Dataset Description

A custom dataset is used consisting of 1728 images from various office settings for the models training, with various possible classes for object recognition. The images are extracted at 1fps (frame per second) from 16 videos which are found on free websites like Pixabay. Afterwards the images are manually annotated into one of eight classes such as person, cell phone, printer, mouse, computer, laptop, keyboard and tablet. To prepare the dataset for training, the images are resized to 640x640, and a static crop is applied to retain the central 25%-75% for each image. Greyscale is applied and the images are tiled in 2 rows x 2 columns. Furthermore, the augmentation is applied to enhance the robustness, as shown in Table 1. After augmentation, the dataset is split into training, validation and testing. The annotated dataset is saved in both COCO and YOLO annotation, ensuring compatibility with multiple deep learning frameworks. The prepared dataset is then used to train the selected models, enabling them to recognize objects and occupants for effective HVAC system automation.

Table 1. Augmentation settings for the selected dataset

Augmentation Steps	Step Setting
Flip	Horizontal and Vertical
Noise	Of 0.1% of pixel
Bounding box rotation	-5% to +5%
Bounding box sheer	-10% to +10%

The image samples in the dataset are shown as an example in Figure.2. The dataset is created with the intention of overcoming limitations such as occlusion, camera angle issues, universal viability and ability to detect other heat emitting objects such as computers, cell phones and tablets. Looking through all the pictures they are all the same sorts of images but in different settings which should give the model a good understanding of how to perform in a variety of office layouts and thereby making the model universally viable. The matter of occlusion is tackled by carefully annotating the persons or object for example 1st image shows 2 full people and part of a 3rd person, image 2 shows 6 people 5 of them are clearly visible and the 6th is occluded. To tackle the issue of camera placement, when the dataset is created, we carefully chose different images where the camera is placed in different angles and positions. Lastly all other objects like computers, cell phones etc. where also annotated to pick these up whilst running object detection. Small devices such as cell phones or tablets are not likely to make a huge difference to the room temperature however, an office floor with a large number of computers could make a significant difference.

Figure 2. Examples of sample images in the dataset

5.2 Object Detection Models: Performance and Evaluation

First, its trained on different object detection such as YOLOv8n, YOLOv9c, and YOLOv10n against the Faster R-CNN baseline, emphasizing detection speed, computational efficiency, and small object detection. The trained models are evaluated based on two key metrics: Mean Average Precision (mAP) and computational efficiency. The following evaluations are conducted:

- **mAP:** The models are evaluated using mAP@50 and mAP@50-95 to measure their accuracy at different Intersection over Union (IoU) thresholds.
- **mAP@50:** Measures precision at a 50% IoU threshold, which is more lenient and captures larger objects.
- **mAP@50-95:** A stricter metric that measures precision across various IoU thresholds, providing insight into the model's performance with smaller or more occluded objects.
- **Detection Speed:** The time taken to process each image is measured to evaluate the model's suitability for real-time detection.
- **Resource Efficiency:** The models are compared in terms of the number of parameters and computational load, with a focus on reducing resource usage while maintaining high detection accuracy.

The performance of YOLOv8n, YOLOv9c, and YOLOv10n is compared against Faster R-CNN, which is chosen as baseline models due to its popularity in object detection research. Faster R-CNN, though accurate, is slower and more resource-intensive.

5.2.1 Faster R-CNN

The Faster R-CNN model achieved a mAP of 87.4% at IoU threshold 0.50 and 60% at IoU thresholds ranging from 0.50 to 0.95, indicating strong object detection performance, especially at lower IoU thresholds. Figure. 3 shows epochs along the x-axis, with mAP and mAP @50-95 on the y-axis, tracking the performance of the Faster R-CNN model as training progresses. Initially, at the lower epochs, both mAP and mAP@50-95 start relatively low, indicating that the model is still learning and hasn't achieved high precision yet. As the epochs increase, the lines representing mAP and mAP@50-95 gradually rise, demonstrating the model's improving ability to correctly detect and classify objects. Around the midpoint of the graph, you may notice a steady increase, showing that the model is refining its predictions and approaching its peak performance. Toward the later epochs, the graph begins to plateau, meaning the model has learned most of what it can from the data. At this point, the mAP and mAP@50-95 stabilize, indicating convergence. Any slight dips or oscillations in the curves might suggest overfitting The need for further tuning, but overall, the upward trend reflects the model's successful learning process over time.

Figure 3. mAP for Faster R-CNN

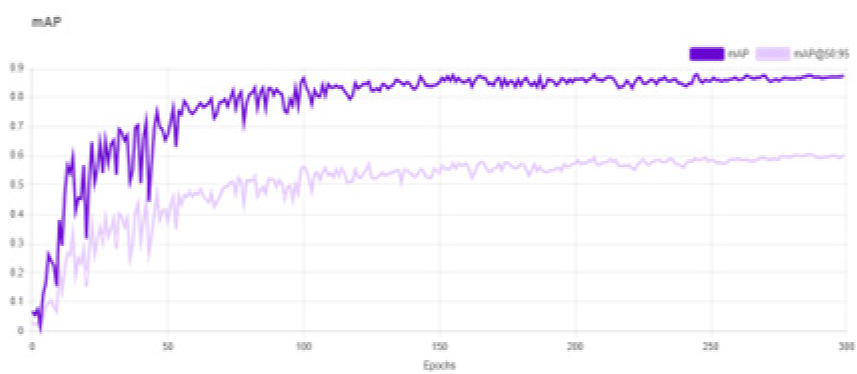
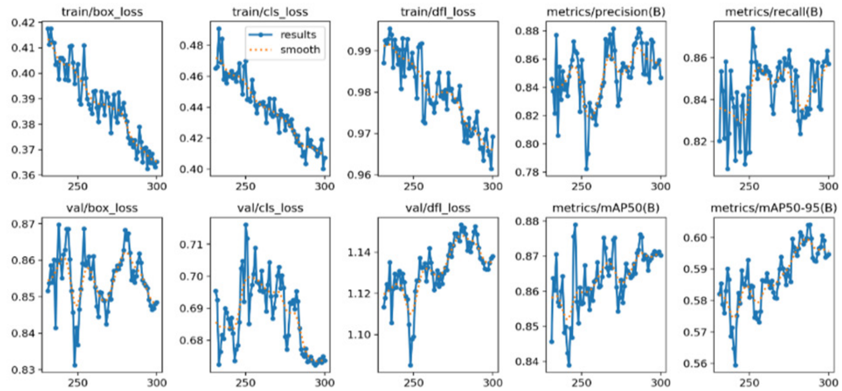


Figure. 4 provides a summary of the Faster R-CNN model's training and performance, showing various loss and metric trends across epochs. The train/box loss, train/cls loss, and train/dfl_loss all show a clear downward trend, indicating that the model is improving in its ability to predict object bounding boxes and classify objects accurately during training. However, on the validation side, both the val/box_loss and val/cls_loss show fluctuations, though with a slight overall decrease. This suggests that while the model is learning, its performance on unseen data remains inconsistent. Similarly, the val/dfl_loss fluctuates more significantly, indicating that the model struggles with object localization on validation data.

For the metrics, precision(B) and recall(B) show variable patterns but maintain relatively high values, reflecting good, though inconsistent, object detection and classification performance. The mAP50(B) and mAP50-95(B), which measure average precision, show an overall upward trend, suggesting improving detection accuracy, though with some variability.

Figure 4. Training visualization for Faster R-CNN



5.2.2 YOLO v8n

The YOLOv8n model achieved a mAP of 87.2% at IoU threshold 0.50 and 59.4% at IoU thresholds ranging from 0.50 to 0.95, demonstrating competitive object detection performance, particularly at lower IoU thresholds. Figure. 5 shows the training progress of YOLO V8 Nano across different metrics. The training box loss, classification loss, and distribution focal loss all exhibit sharp downward trends, particularly in the early epochs, indicating that the model is learning quickly and reducing errors in bounding box predictions and classifications. By around 100 epochs, the losses stabilize, suggesting the model has converged and learned the underlying patterns in the data. The precision and recall metrics improve steadily throughout training, with precision approaching 0.9 and recall nearing 0.85 by the final epoch, indicating that the model is becoming increasingly accurate and consistent in its predictions. Similarly, both mAP50 and mAP50-95 rise steadily, showing that the model's accuracy is improving not just at the 50% IoU threshold but across stricter IoU thresholds, reflecting robust performance. The validation losses follow a similar pattern as training, but with some fluctuations, suggesting occasional overfitting but generally strong generalization to unseen data.

Figure 5. Training Visualization for YOLO V8 Nano

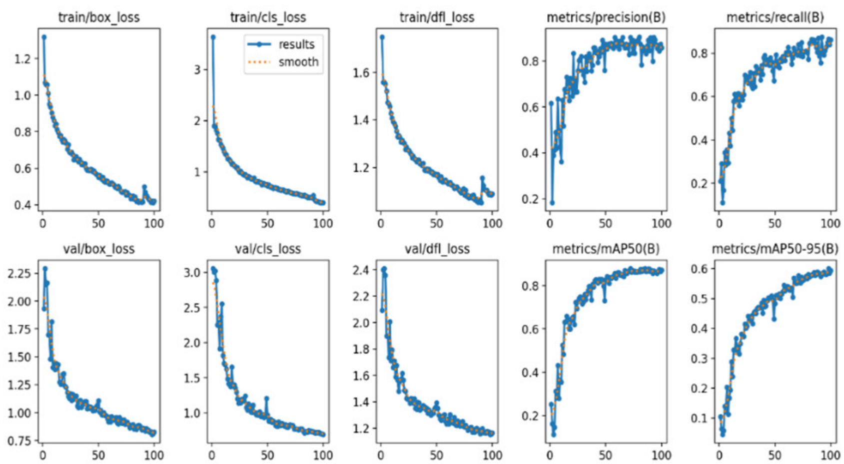


Figure.6. shows the model's precision across different recall levels for various object classes. Each line represents a different object class, with scores close to 1 indicating high precision and recall. The “Cell Phone” class performs the best, with almost perfect precision (0.988), showing that the model is highly accurate and consistent at detecting cell phones. “Laptop” and “Computer” also show strong performances with precision scores of 0.884 and 0.865, respectively. However, the “Tablet” class shows the lowest performance, with a precision of 0.761, indicating the model's difficulty in accurately detecting tablets, possibly due to confusion with similar objects. Overall, the average precision across all classes is 0.874 mAP@0.5, suggesting the model is performing well but has some difficulty with specific categories.

Figure 6. Precision and Recall Curve YOLO V8 Nano

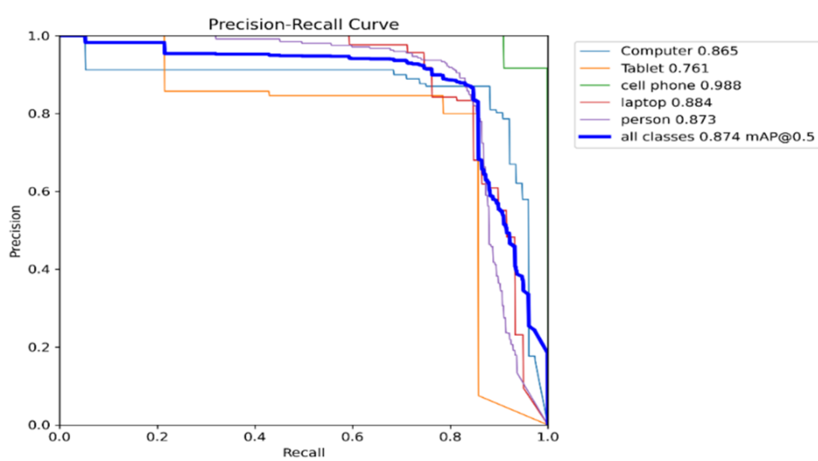


Figure.7. shows how well YOLO V8 Nano is able to classify various objects during testing. Along the diagonal, we observe high values, indicating strong performance in correctly classifying objects. For example, the model correctly classifies “Computer” 92% of the time, and “Tablet” 86% of the time. It perfectly classifies “Cell Phone” 100% of the time, which is a strong indication of the model's ability in that class. For “Person,” there is some confusion, with an 86% classification accuracy, but also a 12% misclassification as “Background.” This misclassification shows the model struggles in distinguishing between persons and backgrounds. Other small confusions are visible with categories like “Laptop” and “Mouse” where small percentages of incorrect classifications exist. Overall, the model performs well but has some difficulty with background distinction.

Figure 7. Normalized Confusion Matrix YOLO V8 Nano

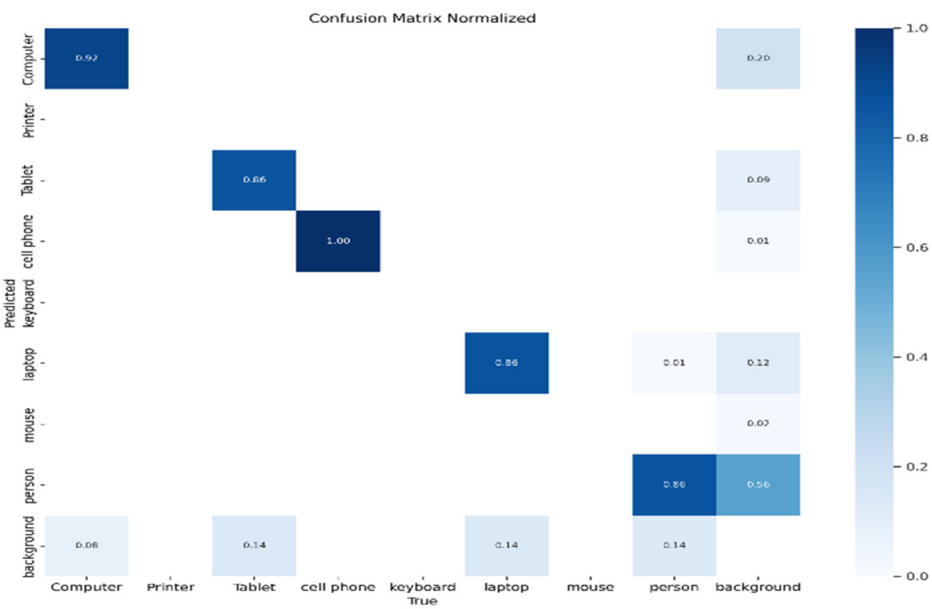


Figure. 8 depicts YOLO V8 Nano's predictions on a test set, where it successfully detects and labels various objects such as “person,” “laptop,” “tablet,” and “cell phone.” Each object is enclosed in a bounding box with a confidence score. The model correctly identifies multiple instances of people and laptops, with confidence scores ranging from 0.6 to 1.0. For instance, it detects a “person” with 0.9 confidence in the center 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 (like people and laptops) but faces challenges with smaller or more similar-looking items like “tablets” and “cell phones.”

Figure 8. YOLO V8 Nano prediction on Test Data

5.2.3 YOLO v9c

The YOLOv9c model achieved a mean Average Precision (mAP) of 88.0% at IoU threshold 0.50 and 59.8% at IoU thresholds ranging from 0.50 to 0.95, indicating strong object detection performance with a notable improvement at the lower IoU threshold compared to previous models. Figure. 9 shows various training metrics and loss curves over 100 epochs for YOLOv9c. The train/box_loss and train/cls_loss curves depict a sharp decrease, starting from approximately 1.2 and 2.5 respectively, and leveling out around 0.4 and 0.5. This indicates that the model's ability to predict bounding boxes and classify objects improves as training progresses. The train/dfl_loss curve, representing the distribution focal loss, follows a similar trend, starting around 1.6 and stabilizing near 1.1. The val/box_loss and val/cls_loss curves show more variability but decrease consistently, which suggests that the model is generalizing fairly well to unseen data, although some fluctuations may indicate slight overfitting. Metrics like precision, recall, and mAP (mean Average Precision) show steady improvement over epochs. Precision rises to around 0.88, and recall climbs above 0.75, demonstrating that the model's accuracy and ability to retrieve correct detections are improving over time. Both mAP_0.5 and mAP_0.5:0.95 curves also show significant growth, with mAP_0.5 peaking at around 0.8 and mAP_0.5:0.95, a stricter evaluation metric, peaking around 0.6, reflecting consistent improvements in detecting objects with more stringent overlap conditions.

Figure 9. Training visualization of YOLOv9c

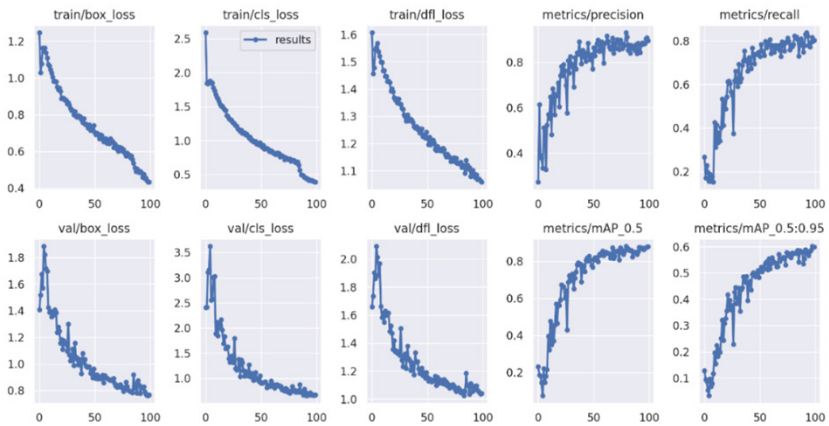


Figure.10 visualizes how YOLOv9c performs across different object classes. Each line represents a class, with performance evaluated based on precision versus recall. The cell phone class performs best, achieving near-perfect precision at 0.968, showing that the model is highly accurate in identifying cell phones. Other classes such as computer and laptop also exhibit strong performance, with precision values of 0.885 and 0.865, respectively. The tablet class performs the weakest, with a precision of 0.798, suggesting that the model has more difficulty detecting tablets compared to other objects. The overall mAP@0.5 for all classes is 0.874, which is a strong indication of the model's good performance across most categories, though some object classes still show room for improvement.

Figure 10. Precision and Recall curve of YOLOv9c

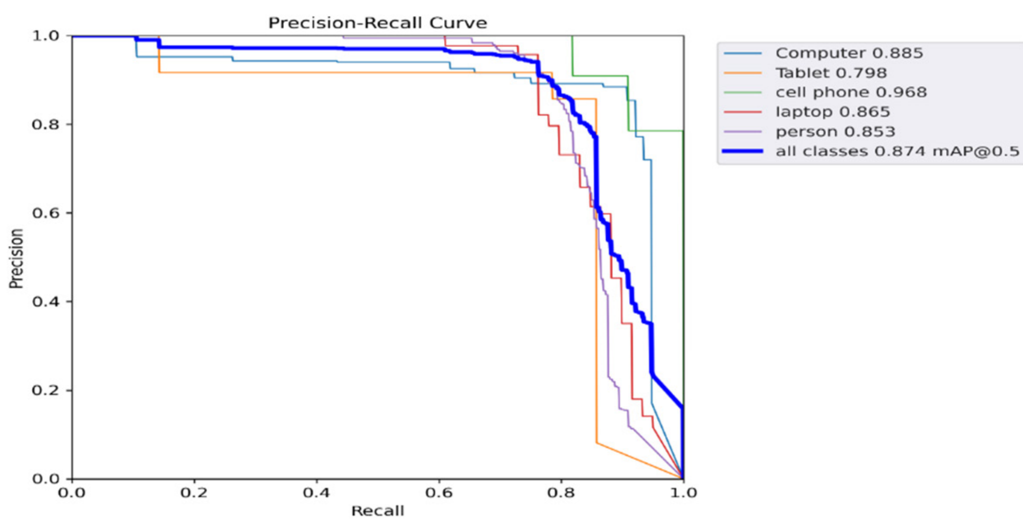


Figure. 11 shows how well YOLOv9c classifies different objects by comparing predicted and true labels. The diagonal values represent correct classifications, with the highest accuracies seen for "Computer" at 91%, "Cell Phone" at 91%, and "Person" at 82%. However, there is some confusion, particularly for the "Person" category, which is misclassified as "Background" 16% of the time. Additionally, the "Mouse" category has an accuracy of 80%, with some confusion possibly occurring with similarly shaped objects or backgrounds. For "Background," the model has a false positive rate of 18% when identifying it as "Person." These off-diagonal values indicate areas where the model could be improved to reduce misclassification, particularly for background and person distinctions.

Figure 11. Confusion Matrix of YOLOv9c

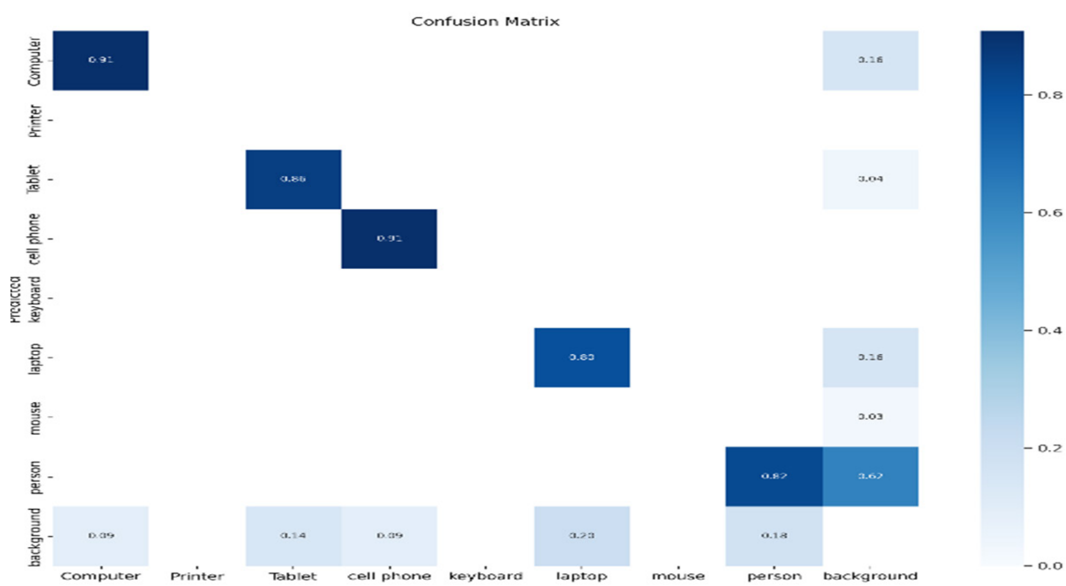


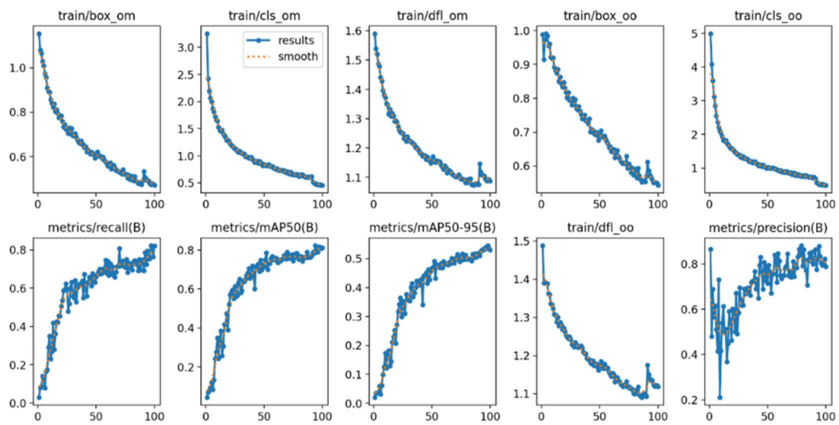
Figure. 12 shows the predictions made by YOLOv9c on a test dataset. The model correctly identifies objects such as "Computer," "Person," "Cell Phone," and "Laptop," with confidence scores generally ranging from 0.8 to 1.0. For example, multiple "Computer" instances are detected with high confidence (0.9), 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.

Figure 12. Prediction of YOLOv9c on Test Data

5.2.4 YOLO v10n

The YOLOv110n model achieved a mean mAP of 81.6% at IoU threshold 0.50 and 54.5% at IoU thresholds ranging from 0.50 to 0.95, indicating a lower object detection performance compared to earlier YOLO models, particularly at higher IoU thresholds. Figure.13 illustrates the YOLOv10n model's training progress over 100 epochs. The train/box_loss, train/cls_loss, and train/dfl_loss all exhibit significant decreases over time, reflecting that the model is effectively learning to localize objects and classify them correctly. Similarly, the metrics/mAP50(B) and metrics/mAP50-95(B) curves show steady improvements, with mAP50 approaching 0.8 and mAP50-95 (a stricter metric) rising to around 0.55. The metrics/precision(B) and metrics/recall(B) also improve consistently, with precision approaching 0.8 by the end of training and recall nearing 0.75. These results suggest that the model is learning effectively and improving in both precision and recall, although the performance can still be refined, particularly for challenging object classes.

Figure 13. Training Visualization for YOLOv10n



The precision-recall curve provides insight into the trade-off between precision and recall for YOLOv10n across different object categories. Person and Laptop classes perform well, with high precision and recall values of 0.862 and 0.855, respectively. Computer also performs well with a precision of 0.853. However, the Tablet class has the lowest precision at 0.679, indicating that the model struggles to consistently identify tablets. The overall mean Average Precision (mAP) across all classes at a 0.5 IoU threshold is 0.816, which shows that the model performs reasonably well but can be improved, especially for classes like “Tablet” and “Cell Phone.”

Figure 14. Precision and Recall for YOLOv10n

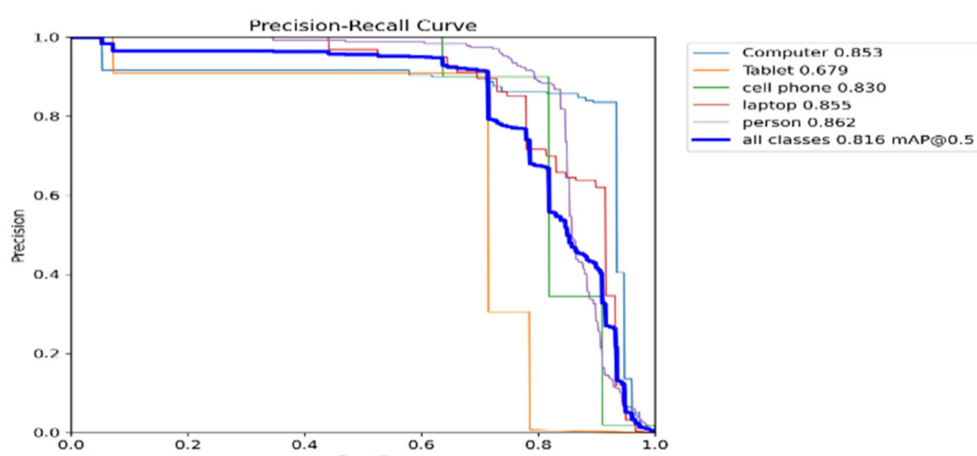
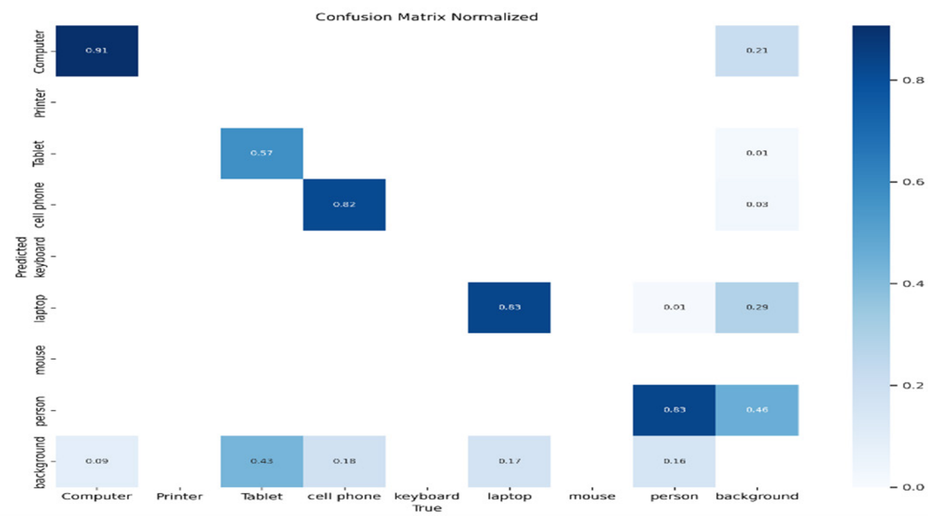


Figure. 15 shows a normalized confusion matrix which shows the model's performance in classifying objects during testing. The diagonal cells represent the percentage of correct predictions, and the non-diagonal cells represent misclassifications. YOLOv10n achieves 91% accuracy in detecting “Computer” and 83% for “Person.” However, it struggles with detecting “Tablet,” with only 57% accuracy, and with “Background,” where there is significant confusion, particularly for “Person,” with a 17% misclassification rate. The model also confuses “Cell Phone” and “Tablet” quite often, with only 82% accuracy for “Cell Phone.” This highlights areas where the model's performance could be improved, particularly in distinguishing between visually similar objects like “Tablet” and “Cell Phone,” and in separating objects from the background.

Figure 15. Confusion Matrix for YOLOv10n



In Figure.16 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 with identifying 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.

Figure 16. Prediction of YOLOv10n on Test Data

5.2.5 Comparison with Baseline Methods

In the result section, the performance of YOLOv8n, YOLOv9c, and YOLOv10n is compared against Faster R-CNN and YOLOv5, based on findings from recent studies. Faster R-CNN, as reported in studies like (Kou et al., 2021), typically achieves mAP@50 scores between 85-90% and an mAP@50 -95 around 40-60%. While these results demonstrate high precision, especially with small or complex objects, the model's significant drawbacks include its detection time of 150-200 milliseconds per image and a large model size of around 42 million parameters. This makes it unsuitable for real-time detection tasks, despite its accuracy.

On the other hand, YOLOv5 (Yu et al., 2023, Wang et al., 2023) achieves mAP@50 values between 87-90% and mAP@50-95 scores between 55-60%, with detection times ranging between 10-30 milliseconds per image. While this model is much faster and more efficient than Faster R-CNN, it still has a higher computational cost than newer models like YOLOv8n and YOLOv9c, requiring around 12 million parameters in its medium-sized configureurations.

In comparison, the models trained in this study shown remarkable improvements in both speed and resource efficiency. YOLOv8n achieves a mAP@50 of 87.2% and an mAP@50-95 of 59.4%, while requiring only 7 million parameters and having a detection time of 10-20 milliseconds per image, making it highly suitable for real-time applications. YOLOv9c, with a mAP@50 of 88.0% and a mAP@50-95 of 59.8%, offers the best balance between accuracy and efficiency, outperforming both Faster R-CNN and YOLOv5 in terms of precision and resource usage. YOLOv10n, though slightly less accurate with a mAP@50 of 81.6% and mAP@50-95 of 54.5%, still provides a reasonable trade-off between speed and precision, with 10 million parameters and detection times of 15-30 milliseconds.

6. DISCUSSION AND FUTURE WORK

The results of this study demonstrate significant advancements in occupancy detection models, particularly when comparing the performance of YOLOv8n, YOLOv9c, and YOLOv10n against established benchmarks such as Faster R-CNN and YOLOv5. Faster R-CNN, while recognized for its high precision in detecting small and complex objects, falls short in real-time applications due to its detection time of 150-200 milliseconds per image and its considerable model size of approximately 42 million parameters. Despite achieving mean Average Precision (mAP) scores ranging from 85-90% at IoU 0.50 and 40-60% at IoU 0.50-0.95, these drawbacks make it impractical for environments that require immediate occupancy detection.

YOLOv5 demonstrates a better balance between speed and accuracy, achieving mAP@50 scores between 87-90% and mAP@50-95 scores between 55-60%. With detection times ranging from 10-30 milliseconds, YOLOv5 is more efficient than Faster R-CNN, though it still computational intensive with approximately 12 million parameters in its medium-sized configureurations. YOLOv8n, YOLOv9c, and YOLOv10n—show remarkable improvements in both speed and resource efficiency. YOLOv8n stands out with a mAP@50 of 87.2% and a mAP@50-95 of 59.4%, requiring only 7 million parameters and achieving detection times of 10-20 milliseconds per image. This combination of accuracy and speed positions YOLOv8n as highly suitable for real-time occupancy detection applications. YOLOv9c further enhances this performance, boasting a mAP@50 of 88.0% and a mAP@50-95 of 59.8%, making it the model with the best balance between accuracy and efficiency in this study. This model outperforms both Faster R-CNN and YOLOv5 in terms of both precision and resource usage, solidifying its role as a preferred choice for real-time applications that require prompt decision-making. YOLOv10n shows slightly lower accuracy with a mAP@50 of 81.6% and a mAP@50-95 of 54.5%, it still provides a reasonable trade-off between speed and precision, making it a viable option in scenarios where computational resources are limited, or where the emphasis is placed on processing speed.

These findings show that further exploration in the field of occupancy detection should focus on several key areas to enhance the capabilities and applications of the models presented in this study. Integrating multi-modal data sources, such as PIR, infrared, CO2 sensors, alongside video analysis, can improve detection accuracy in complex environments. Additionally, prioritizing privacy protection through advanced techniques like data anonymization and federated learning will ensure occupant privacy while still harnessing valuable occupancy data. Long-term performance monitoring in diverse real-world settings will provide insights into occupancy patterns, informing better HVAC control strategies and maximizing energy savings. Implementing adaptive learning algorithms will enable models to continuously update and refine themselves based on new data, thereby enhancing their effectiveness over time. Finally, assessing the cost-effectiveness and scalability of these systems will be crucial for their widespread adoption, particularly in retrofitting existing buildings. By addressing these areas, future research can significantly contribute to the development of more robust, efficient, and sustainable occupancy detection systems, ultimately advancing the optimization of energy use in indoor environments and promoting environmental sustainability

FUNDING STATEMENT

This work is supported in part by XJTLU Research Development Fund under Grant No. RDF21-01-015.

REFERENCES

Abdel-Razek, S. A., Marie, H. S., Alshehri, A., & Elzeki, O. M. (2022). Energy efficiency through the implementation of an AI model to predict room occupancy based on thermal comfort parameters. *Sustainability (Basel)*, 14(13), 7734. DOI: 10.3390/su14137734

Agarwal, Y., Balaji, B., Gupta, R., Lyles, J., Wei, M., & Weng, T. (2010, November). Occupancy-driven energy management for smart building automation. In *Proceedings of the 2nd ACM workshop on embedded sensing systems for energy-efficiency in building* (pp. 1-6).

Anjomshoaa, A., Duarte, F., Rennings, D., Matarazzo, T. J., Desouza, P., & Ratti, C. (2018). City scanner: Building and scheduling a mobile sensing platform for smart city services. *IEEE Internet of Things Journal*, 5(6), 4567–4579. DOI: 10.1109/JIOT.2018.2839058

Beltran, A., Erickson, V. L., & Cerpa, A. E. (2013, November). Thermosense: Occupancy thermal based sensing for hvac control. In *Proceedings of the 5th ACM Workshop on Embedded Systems For Energy-Efficient Buildings* (pp. 1-8).

Carli, R., Cavone, G., Ben Othman, S., & Dotoli, M. (2020). Iot based architecture for model predictive control of hvac systems in smart buildings. *Sensors (Basel)*, 20(3), 781. DOI: 10.3390/s20030781 PMID: 32023965

Chung, J., Gulcehre, C., Cho, K., & Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555.

Dong, B., & Lam, K. P. (2014, February). A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. []. Springer Berlin Heidelberg.]. *Building Simulation*, 7, 89–106.

Dong, B., & Lam, K. P. (2014, February). A real-time model predictive control for building heating and cooling systems based on the occupancy behavior pattern detection and local weather forecasting. []. Springer Berlin Heidelberg.]. *Building Simulation*, 7, 89–106.

Dorokhova, M., Ballif, C., & Wyrsch, N. (2020). Rule-based scheduling of air conditioning using occupancy forecasting. *Energy AI*, 2, 100022. DOI: 10.1016/j.egyai.2020.100022

Gomez, A., Conti, F., & Benini, L. Thermal image-based CNN's for ultra-low power people recognition. *Proceedings of the 15th ACM International Conference on Computing Frontiers*, 2018. 326-331. DOI: 10.1145/3203217.3204465

Griffiths, E., Assana, S., & Whitehouse, K. (2018). Privacy-preserving image processing with binocular thermal cameras. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4), 1–25. DOI: 10.1145/3161198

Hashimoto, K., Morinaka, K., Yoshiike, N., Kawaguchi, C., & Matsueda, S. (1997, June). People count system using multi-sensing application. In Proceedings of International Solid State Sensors and Actuators Conference (Transducers' 97) (Vol. 2, pp. 1291-1294). IEEE.

Hobson, B. W., Lowcay, D., Gunay, H. B., Ashouri, A., & Newsham, G. R. (2019). Opportunistic occupancy-count estimation using sensor fusion: A case study. *Building and Environment*, 159, 106154. DOI: 10.1016/j.buildenv.2019.05.032

Hu, S., Wang, P., Hoare, C., & O'Donnell, J. (2022). Building occupancy detection and localization using CCTV camera and deep learning. *IEEE Internet of Things Journal*, 10(1), 597–608. DOI: 10.1109/JIOT.2022.3201877

Jiang, C., Masood, M. K., Soh, Y. C., & Li, H. (2016). Indoor occupancy estimation from carbon dioxide concentration. *Energy and Building*, 131, 132–141. DOI: 10.1016/j.enbuild.2016.09.002

Koebrich, S., Chen, E. I., Bowen, T., Forrester, S., & Tian, T. (2019). *2017 Renewable Energy Data Book: Including Data and Trends for Energy Storage and Electric Vehicles. National Renewable Energy Lab*. NREL. DOI: 10.2172/1581221

Kou, X., Du, Y., Li, F., Pulgar-Painemal, H., Zandi, H., Dong, J., & Olama, M. M. (2021). Model-based and data-driven HVAC control strategies for residential demand response. *IEEE Open Access Journal of Power and Energy*, 8, 186–197. DOI: 10.1109/OAJPE.2021.3075426

Lee, G.-C., Lee, S.-H., & Yoo, J. (2018). 3-D people counting with a stereo camera on gpu embedded board. *Applied Sciences (Basel, Switzerland)*, 8(11), 2017. DOI: 10.3390/app8112017

Metwaly, A., Queralta, J. P., Sarker, V. K., Gia, T. N., Nasir, O., & Westerlund, T. (2019, October). Edge computing with embedded ai: Thermal image analysis for occupancy estimation in intelligent buildings. In *Proceedings of the INTelligent Embedded Systems Architectures and Applications Workshop 2019* (pp. 1-6).

Moons, B., Bankman, D., & Verhelst, M. (2019). *Embedded deep learning. Circuits for Always-on Neural Network Processing* (1st ed.). Springer.

Paidi, V., Fleyeh, H., & Nyberg, R. G. (2020). Deep learning-based vehicle occupancy detection in an open parking lot using thermal camera. *IET Intelligent Transport Systems*, 14(10), 1295–1302. DOI: 10.1049/iet-its.2019.0468

Petersen, S., Pedersen, T. H., Nielsen, K. U., & Knudsen, M. D. (2016). Establishing an image-based ground truth for validation of sensor data-based room occupancy detection. *Energy and Building*, 130, 787–793. DOI: 10.1016/j.enbuild.2016.09.009

Rastogi, K., & Lohani, D. (2020). IoT-based indoor occupancy estimation using Edge Computing. *Procedia Computer Science*, 171, 1943–1952. DOI: 10.1016/j.procs.2020.04.208

Rueda, L., Agbossou, K., Cardenas, A., Henao, N., & Kelouwani, S. (2020). A comprehensive review of approaches to building occupancy detection. *Building and Environment*, 180, 106966. DOI: 10.1016/j.buildenv.2020.106966

Ryu, S. H., & Moon, H. J. (2016). Development of an occupancy prediction model using indoor environmental data based on machine learning techniques. *Building and Environment*, 107, 1–9. DOI: 10.1016/j.buildenv.2016.06.039

Sayed, A. N., Bensaali, F., Himeur, Y., & Houchati, M. (2023). Edge-based real-time occupancy detection system through a non-intrusive sensing system. *Energies*, 16(5), 2388.

Sun, K., Zhao, Q., Zhang, Z., & Hu, X. (2022). Indoor occupancy measurement by the fusion of motion detection and static estimation. *Energy and Building*, 254, 111593. DOI: 10.1016/j.enbuild.2021.111593

Tien, P. W., Wei, S., Calautit, J. K., Darkwa, J., & Wood, C. (2020). A vision-based deep learning approach for the detection and prediction of occupancy heat emissions for demand-driven control solutions. *Energy and Building*, 226, 110386. DOI: 10.1016/j.enbuild.2020.110386

Wahl, F., Milenkovic, M., & Amft, O. A distributed PIR-based approach for estimating people count in office environments. 2012 IEEE 15th International Conference on Computational Science and Engineering, 2012. IEEE, 640-647.

Wang, C., Zhang, Y., Zhou, Y., Sun, S., Zhang, H., & Wang, Y. (2023). Automatic detection of indoor occupancy based on improved YOLOv5 model. *Neural Computing & Applications*, 35(3), 2575–2599. DOI: 10.1007/s00521-022-07730-3 PMID: 36068815

Wang, S., Burnett, J., & Chong, H. (1999). Experimental validation of CO2-based occupancy detection for demand-controlled ventilation. *Indoor and Built Environment*, 8(6), 377–391. DOI: 10.1177/1420326X9900800605

Yu, F., Zhang, G., Zhao, F., Wang, X., Liu, H., Lin, P., & Chen, Y. (2023). Improved YOLO-v5 model for boosting face mask recognition accuracy on heterogeneous IoT computing platforms. *Internet of Things : Engineering Cyber Physical Human Systems*, 23, 100881. DOI: 10.1016/j.iot.2023.100881

Yun, J., & Lee, S.-S. (2014). Human movement detection and identification using pyroelectric infrared sensors. *Sensors (Basel)*, 14(5), 8057–8081. DOI: 10.3390/s140508057 PMID: 24803195

Zemouri, S., Gkoufas, Y., & Murphy, J. (2019, August). A machine learning approach to indoor occupancy detection using non-intrusive environmental sensor data. In *Proceedings of the 3rd international conference on big data and internet of things* (pp. 70-74).

Zemouri, S., Magoni, D., Zemouri, A., Gkoufas, Y., Katrinis, K., & Murphy, J. (2018, September). An edge computing approach to explore indoor environmental sensor data for occupancy measurement in office spaces. In 2018 IEEE International Smart Cities Conference (ISC2) (pp. 1-8). IEEE.

KEY TERMS AND DEFINITIONS

OBC: Occupancy-based control.
HVAC: Heating, ventilation, and air conditioning.
PIR: Infrared pro proximity sensors.
RC: Resistance capacitance.

FCHD: Fully convolutional head detector.
mAP: Mean average precision.
BEMS: Building energy management systems.

ENDNOTES

[1] https://www.gumstix.com/manufacturer/nvidia/nano-snapshot.html

[2] https://coral.ai/

[3] https://aws.amazon.com/deeplens/

[4] https://www.mturk.com/