MIE 1050 Course Report
Contact – Free Occupancy Detector

By

Andy Xin: andy.xin@mail.utoronto.ca

Linju Arangassery Jose: linju.arangassery@mail.utoronto.ca

UNIVERSITY OF
TORONTO

MIE Department

Supervised by Prof. Ardevan Bakhtari

December 17, 2025

TABLE OF CONTENTS

1. Introduction

A Contact-Free Occupancy Detector is an electronic system designed to identify the occupancy level of an indoor space, such as a classroom, classifying it as Low, Medium, or High without any physical interaction. This intelligent system plays a key role in advancing smart building technology and improving energy management strategies. Efficient monitoring of room occupancy is essential for intelligent building management systems, particularly for optimizing HVAC operation, lighting control, and energy usage.

Traditional occupancy detection methods, such as manual counting, badge-based access, or camera-based systems, often suffer from high costs, installation complexity, or maintenance issues. To address these challenges, the Contact-Free Occupancy Detector combines acoustic sensing and ultrasonic doorway crossing events, supported by advanced signal processing and machine learning algorithms, to accurately infer occupancy levels in a classroom environment.

1.1. Problem motivation

Existing indoor occupancy detection technologies exhibit fundamental limitations when applied to dynamic classroom environments. Passive Infrared (PIR) sensors are highly dependent on motion and therefore perform poorly when occupants remain stationary for extended periods. Which leads to false occupancy detection. Camera-based systems can provide high accuracy but introduce significant privacy, ethical, and data governance concerns, in addition to high installation and computational costs. CO_2-based sensing methods rely on indirect measurement of human presence and suffer from long response times due to air mixing and HVAC operation, making them unsuitable for real-time occupancy estimation. Wearable or badge-based approaches require active user participation and compliance, which is impractical in classrooms. These limitations highlight the need for an alternative sensing approach that is accurate, low-cost and responsive to rapid occupancy changes.

1.2. Problem Addressed

Contact-Free Occupancy Detector addresses the challenge of estimating indoor occupancy levels from noisy, high-rate sensor data in a non-intrusive and privacy-preserving manner. The problem involves mapping continuous classroom audio signals and discrete doorway crossing measurements into stable and interpretable occupancy tiers (Low, Medium, High). The current system fuses audio-derived activity and speech features with ultrasonic detections. which aligns per-second headcount ground-truth labels to fixed 10-second decision windows using majority voting. The system trains a time-aware supervised classifier augmented with temporal smoothing. This approach enables reliable detection of both stationary occupancy (via speech activity) and dynamic flow (via doorway crossings). Thus Contact Free Occupancy Detector estimates occupancy suitable for real-time and offline applications.

2. System Design

2.1. Hardware

The hardware platform is based on the UoTMIE1050 Rev1.0 electronic board, built with an ESP32-S3 microcontroller, which provides integrated analog to digital conversion. To improve acoustic sensing performance, an external microphone (S-MD-5P) was integrated into the board due to the limited sensitivity of the onboard microphone. The S-MD-5P microphone was powered using the regulated 3.3 V supply available on the board and the analog output of the microphone is connected to the ESP32-S3 through analog input IO9 which is bridged to the internal ADC of ESP32 chip, as shown in Fig. 1. The microphone output is conditioned using an analog front-end that provides appropriate biasing and amplification prior to digitization, ensuring compatibility with the ADC input range and minimizing clipping and noise effects.

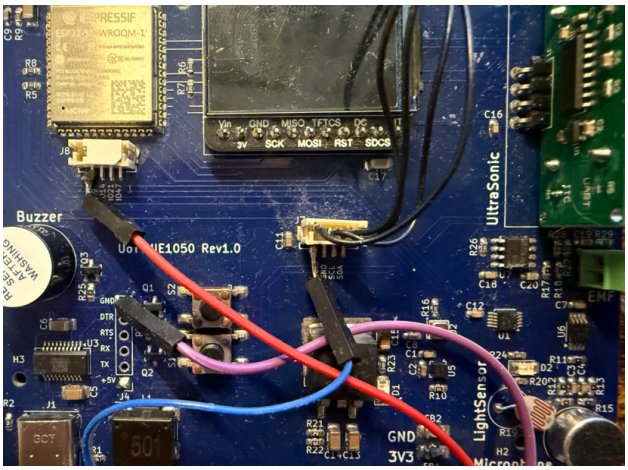
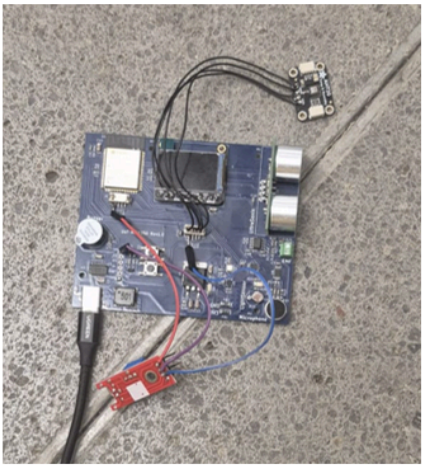


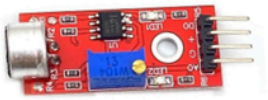
Figure 1. Integration of S-MD-5P microphone to UoTMIE1050 Rev1.0 electronic board

2.2. Experimental Setup

The experimental setup was done in a real classroom environment to capture representative occupancy behavior under normal operating conditions. An electronic board (UoTMIE1050 Rev1.0 electronic board) was installed near the classroom entrance to monitor doorway activity. Due to the low sensitivity of the initially installed onboard microphone, an external microphone sensor (S-MD-5P) was integrated into the electronic board. However, to ensure reliable and high-quality acoustic data, continuous audio recordings during active lecture periods were ultimately captured using a laptop microphone. The ultrasonic range sensor was mounted and angled across the doorway such that its line of signal intersected the typical walking path, resulting in a stable baseline distance under idle conditions and distinct distance excursions during crossing events. The electronic board, together with all integrated sensors, was installed near the door, as shown in Fig. 2. An onboard accelerometer was also integrated and mounted to measure floor vibrations associated with student movement. All sensor data were timestamped relative to the start of the experiment to ensure accurate temporal alignment across modalities. Manual headcount measurements were recorded at discrete time intervals and subsequently resampled to generate continuous ground-truth occupancy labels for data analysis and model training.



MIE 1050 electronic board



S-MD-5P

Figure 2. Experimental setup

2.3. Software & Data collection

During the data collection stage, the input data stream contains temperature, relative humidity, gas resistance, the acceleration on 3 dimensions, light sensor reading and the ultrasonic readings from the board. The input stream is read through a serial port using Python script and recorded in a csv file with timestamps associated with sensor readings. In addition, since the required sampling rate for the external microphone is much higher than the normal sensors, which means it will be space-consuming and time consuming to read if saved in the same csv format. Subsequently, the mic raw information is stored in a binary format, which contains both timestamp and raw readings. This specific format takes much less space compared to a csv format and also guarantees a fast reading and processing speed. At the same time, the audio information is also collected using the internal microphone in the laptop, which is powerful and suitable for the extraction of high-level audio features for machine learning usage. In order to retain the raw audio signal to its original format, a waveform audio file (WAV) format is used to store the uncompressed audio data. This format is able to preserve the audio information as much as possible while still retaining high interpretability and processability in the following analysis. The pulse code modulation (PCM) encoding protocol with 16-bit sampling depth and 48kHz sampling rate is used to transform an analog signal to digital signal. Giving the context of feature extraction used for speech analysis and/or detection, this precision is more than enough. Regarding the design of data collection pipeline, it should be noted that the incoming data stream exhibits different characteristics in sampling rate and information density, which means specific measures need to be applied on both Arduino and the Python logging script to make sure the data are recorded timely and accurately. Regarding the code uploaded to ESP32 sensor board, the external microphone data and other sensor data are collected in 2 separate loops to accommodate the high sampling rate required by the external microphone. For sensor data including temperature, humidity and acceleration, the sampling rate is

limited by the BME680 sensor which can only provide data at a minimum time interval of more than 300ms, which means at most times the sensor board can only achieve a sampling rate of 2Hz. On the other hand, the extra external microphone can be easily sampled at more than 200Hz, which is almost 100 times faster compared to the other sensors. Therefore, if the data is read and sent through the serial port at the same program loop, the raw microphone will be greatly delayed by the slow sensors. As a result, the data from the microphone is put in a separate loop which only handles the microphone data, while the other data from slow sensors are only visited and recorded at a fixed interval of 400 milliseconds. By separating the slow and fast sensors, this specific setting guarantees that the data from the external microphone can be collected at a sample that is high enough and also prevents saving duplicated data from slow sensors. The serial port will output the external mic data most of the time while the slow sensors will only appear when a certain time has passed from the last slow sensor output. In the Python script, since the data comes in at the high sampling rate, the program should be able to handle the data promptly. However, the audio data comes in at a high bitrate which means it will take a longer time to process and encode. Therefore, instead of putting these 2 tasks in a synchronized and ordered execution program, these contradictory requirements from different tasks are addressed by creating different threads. Specifically, 3 threads are created apart from the main thread. The thread separates the saving (writing) step of data processing from the acceptance of incoming data. In other words, for both audio data from laptop internal microphone and serial data from sensor board, the program assigns 2 threads including the main thread to record and put the data in a “Queue” object at whatever sampling rate that is required by the nature of the data. The other 2 threads are only responsible for pulling data from the Queue object and storing them in the corresponding file format according to the type of data. Note that the main thread is only taking care of the audio recording, which theoretically consumes the most computational resources. As a result, when one experiment for data collection is completed, 3 separate file are generated: one csv file containing readings from slow sensor with a maximum sampling rate of 2 Hz; one binary file containing only the raw readings from the external microphone with a minimum sampling rate of 50Hz; one WAV file with uncompressed audio signal with a sampling rate of 48kHz and a bit depth of 16. This configuration keeps the interpretability, accuracy, and efficiency of the data collection process while significantly reducing the complexity of further processing and testing. Also, aside from the data used for prediction, the ground truth label information is also attained with a manual head count of the people in the room.

2.4. Data Selection

The multiple sensor modalities were initially evaluated for their information content and suitability for occupancy estimation. The parmeters which are measured such as acoustic signals, ultrasonic range measurements, accelerometer data, and environmental parameters such as gas concentration, temperature, and relative humidity. The onboard microphone and new microphone sensor (S-MD-5P) shows insufficient sensitivity and signal-to-noise ratio for reliable speech feature extraction. The microphone signal from the laptop exhibits high temporal variability with distinct amplitude spikes corresponding to speech bursts and human activity. These signals indicate a strong response to short-term occupancy-related events. The gas sensor signal shows a slow transient rise followed by gradual drift over tens of minutes, reflecting cumulative environmental effects and HVAC air circulation rather than instantaneous changes in student presence. The absence of sharp transitions or repeatable patterns aligned with known occupancy changes indicates low temporal resolution and poor suitability for real-time occupancy inference.

The temperature signal displays a rise followed by saturation, consistent with thermal control system behavior (HVAC), and lacks the responsiveness required to detect short-term occupancy variations. The collected data are shown in the following Fig. 3. These environmental sensors provide low variance at short time scales and weak correlation with ground-truth occupancy labels, making them ineffective for discriminating between Low, Medium, and High occupancy states within the targeted 10-second decision windows.

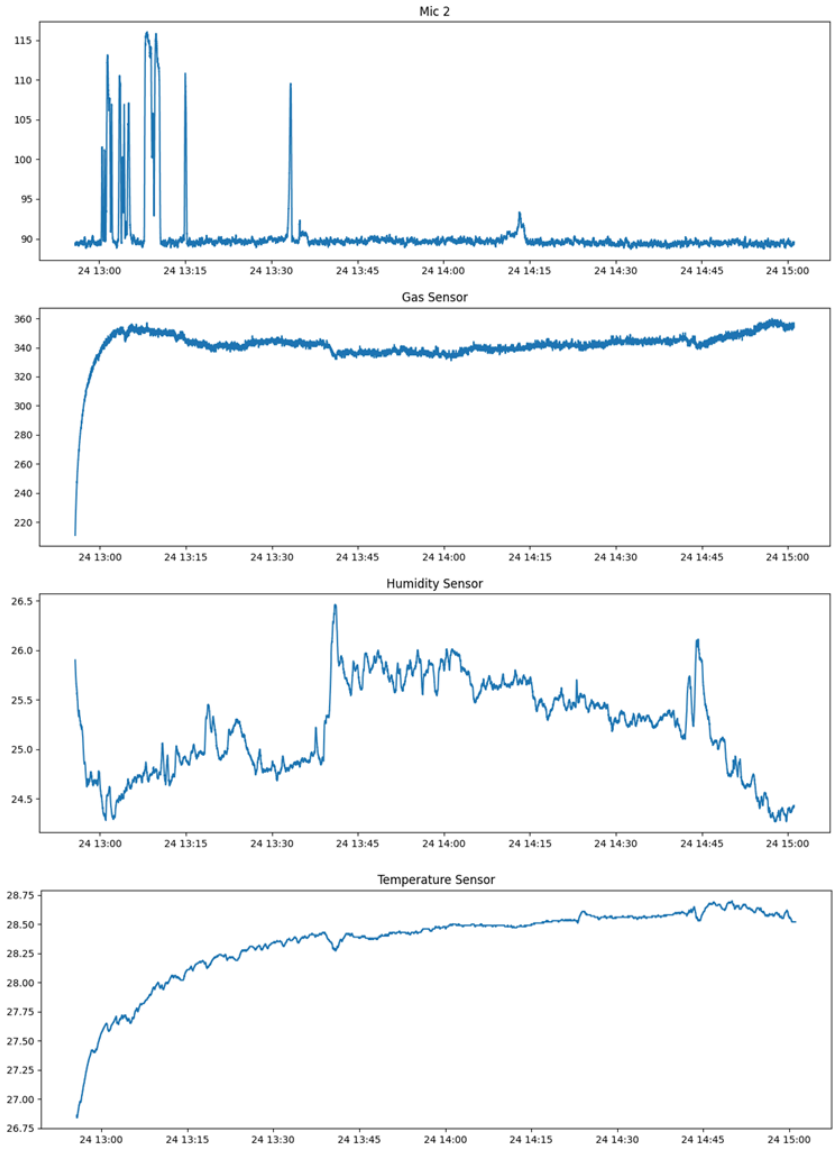


Figure 3. Multi-sensor time domain data collected during classroom experiment

The ultrasonic range sensor, positioned across the doorway, produces a crossing event high-confidence distance excursions when a student crosses the entrance. Ultrasonic crossing event details are shown in Fig 4. These excursions are detected using threshold-based event extraction, yielding precise cross timestamps that are robust to background noise and environmental disturbances.

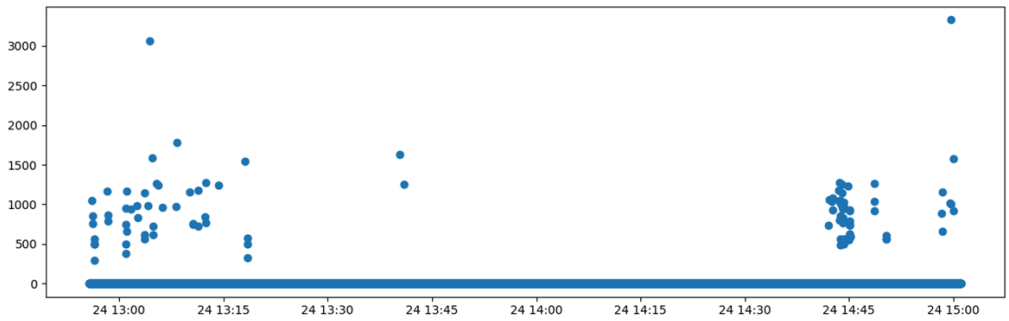


Figure 4. Ultrasonic doorway crossing events over time

Ground-truth occupancy labels are derived from manual headcount logs recorded during data collection. Raw headcounts are normalized by room capacity and discretized into categorical occupancy tiers such as Low, Medium, and High.

We also checked the built-in accelerometer by recording vibrations from student movements and people walking. We calibrated it by testing if it worked properly, measuring the noise level, and checking its response to known movements. After applying a rolling window with a size of 50, the tested data is plotted in Fig. 5 using the mean value in each window. The test data showed that vibrations from the floor were too weak or too inconsistent exhibiting little deviation from background noise. Limited by the sensitivity of the accelerometer, these readings could not create clear, repeatable patterns and thus the accelerometer did not give reliable information for detecting occupancy. As a result, we did not use it for occupancy detection.

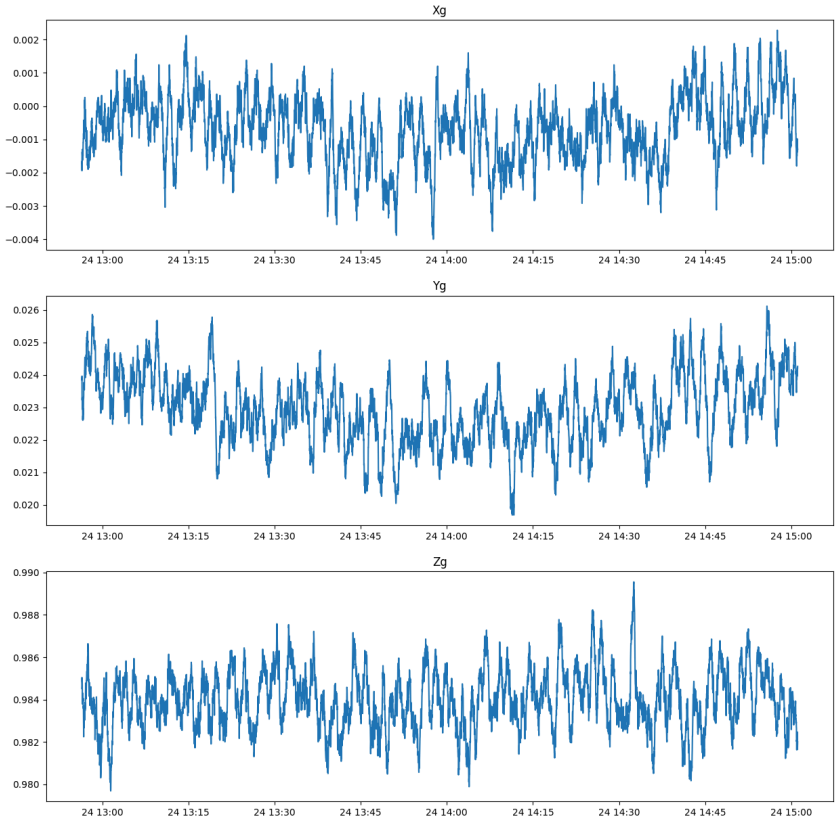


Figure. 5. Accelerometer readings in X,Y,Z directions by taking mean of rolling window of size 50

We found that the built-in microphone on the board was not sensitive enough and typically exhibits little reaction to outside stimuli. As a result, we added an external microphone sensor called S-MD-5P to the electronic board. We calibrated it by checking the signal strength, noise level, and how well it picked up speech and room sounds. This new sensor worked better than the original microphone, however, it still had a low signal-to-noise ratio (SNR) and uneven frequency response in the classroom. It seems that the external microphone has low sensitivity and only reacts to large pulses, which provides little context for the audio information. The plot for the external signal is shown in Fig. 6. Because of these issues, we decided the onboard microphone was not good enough and did not use it in the final data selection.

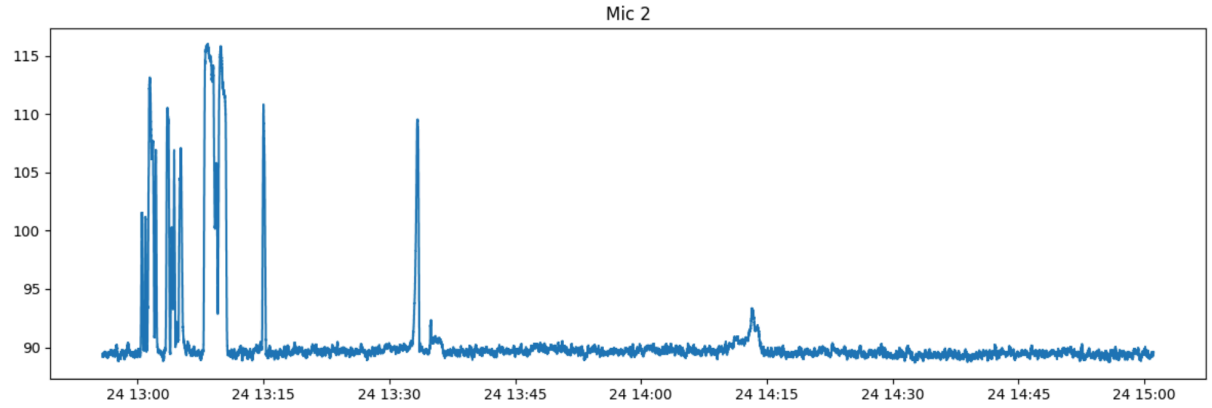


Figure 6. External microphone readings with only occasional bumps observed

Based on these observations, a principled data reduction strategy was applied to exclude sensor streams whose statistical characteristics were incompatible with the temporal requirements of the classification task. Retaining slow-response signals would introduce redundancy, temporal misalignment. This ultimately degrades model performance. The final dataset therefore prioritizes acoustic signals from laptop mic, which capture stationary occupancy through speech activity, and ultrasonic crossing events, which provide sparse but reliable indicators of dynamic crossing. This data-driven selection process ensures that the machine learning model is trained on sensor modalities with high signal-to-noise ratio, appropriate temporal resolution, and direct relevance to human presence, resulting in a more robust and interpretable occupancy estimation system.

2.5. Feature Extraction

Given the limitations of sensor readings related to atmospheric characteristics, dynamic features, and the external microphone, the final model will be constructed using the range sensor as well as the audio file attained using laptop microphones. Conceptually, the audio as well as ultrasonic readings will be expanded into multiple features which together compile into a high dimensional feature vector, which will be used in the training of logistic regression models. Since the project is looking at the changing of the occupancy level inside the room, the model has to be aware of a temporal context, which means when constructing the dataset for training, there is a need to design a method to equip the model with temporal awareness. With this concept in mind, the audio data is first grouped in frame level: each piece of data accounts for a window of 20 milliseconds, where spectral features are calculated and described. The data sample used for training (i.e. decision windows) are defined in 10 seconds length. Note that a

50-percentage overlap is applied for both time scales. Namely, half of each sample in either time scale overlaps with the sample after it. For example, in a 20ms scale, the next sample starts at the half of the last sample, which is at 10ms starting from 0 of the first sample, and the same applies in the 10 seconds time scale. This configuration can also be seen as an overlapped rolling window used to provide more responsive update by setting intervals between each sample as 5 seconds and to produce smoother behavior without sacrificing time resolution. Note that the information provided by the range sensor will only be considered in the 10-second window-level features and some audio features in the 10-second scale are directly related to the frame level features.

The detailed processing procedure of the ultrasonic reading is first discussed. For the experiments, the sensor board will be set near the door placed perpendicular to the route that people will generally take when they enter the room. The distance between the range sensor and the obstacle will be set so that during no pass-by, the reading will reside near 0 since it exceeds the detection limits. When people enter the classroom in front of the sensor, it will temporarily occlude the ultrasonic beam and hence create an actual reading above 0. By filtering the readings above a certain threshold, it is straightforward to yield the relative timestamp of the crossing events. By using the relative timestamp of crossing events, several features can be drawn regarding the window level used for prediction. The “CrossCount” feature measures the number of crossing events observed in the current window that lasts for 10 seconds. The “CrossPerMinute” feature provides information on the crossing happens in a 1-minute horizon built in a rolling window format, which can be treated as a more comprehensive description of the crossing event that spans a longer interval compared to the plain “CrossCount” feature. The dwell ratio, on the other hand, is constructed by taking the ratio of the time of all crossing events in a given window with respect to the total time of the entire window, which is 10 seconds. This specific parameter is helpful for distinguishing between crossing events triggered by a single person and the events caused by a group of people or consecutive passings. Aside from the parameters focuses on each individual window, a temporal context is generated by calculating $\Delta\mathrm{Cross}$, which represents the difference of “CrossCount” in the current window and previous window.

Before extracting features from the audio file, several pre-processing is required. After loading the .wav file, it is first transformed to a NumPy array object and converted to single channel audio from stereo format. The audio data is then normalized to the range of [-1,1] and down sampled from 48kHz to 16kHz to reduce the load of future processing and model training/inference. The audio signal is then passed through a Butterworth style IIR filter to band-pass the signal from 300-3000Hz which is the primary frequency band where human speech falls in. Such band-pass signals will be used to calculate some specific audio features in the following pipeline. Note that when applying the IIR filter the digital filter is applied twice in the forward and backward direction to create zero-phase shift and avoid signal distortion. Before diving into the calculation of frame-level features extracted from audio signals, it has to be mentioned that any frame that locates within 2 seconds from a crossing event is ignored to avoid the interference of door slamming on audio feature generation. The frame-level audio features contains RMS energy, band ratio, spectral flatness, and the voice action fraction (VAF) calculated using zero-crossing rate (ZCR), spectral tilt and loudness level. It should be noted before applying the fast Fourier transform (FFT) before calculating features, a Hann window function is first applied on the frame-level signal to help mitigate spectral leakage brought by the rectangle-window sampling of a particular frame. For audio features in the frame, RMS energy is an intuitive parameter used to measure the average

loudness. Before introducing the following features, it is worthy to bring in a specific parameter—power spectrum density (PSD), which quantifies how the power of a signal is distributed across different frequency components. Specifically, it is a statistical metric that describes the average power of a signal as a function of frequency. The PSD is calculated by taking the square of absolute values of the frequency band after FFT on the real part and normalized by the sample length. A two-band spectral tilt measures the ratio of PSD at 300-1000 and 100-3000Hz band in a log scale and expressed in dB. The spectral tilt is used to verify that a frame's spectrum isn't low-heavy brought by door, footstep or HVAC and actually has speech-band content above 1 kHz. The zero crossing rate (ZCR) of the audio signal measures how often the signal passes through zero, which is a representative feature used to reflect the overall oscillation behavior of the audio and such behavior is closely tied to the existence of speech in the room. By composing three features mentioned above: ZCR, RMS, and 2-band spectral tilt in a certain logic, a mask can be generated accordingly to determine if speech is observed for the particular sample within the frame. When determining the existence of speech for a given audio sample, the following rules are applicable: the sample will be regarded as "speech" if it satisfies following conditions: (1). The sample has a RMS level larger than threshold; (2). a ZCR value lies between prescribed lower and upper bound; (3). 2-band spectral tilt value larger than given, which means the 1000-3000Hz band contains more energy compared to the lower band. By dividing the number of samples with speech active to the total samples in a given frame, the voice active fraction (VAF) can be easily attained. Since the frame level duration is still short: only 20ms, in order to prevent creating flickering results, a persistence check is applied to each sample in the frame, which means the sample can be only considered as speech active if 3 consecutive trues are observed. Aside from the metrics used to calculate frame-level VAF, the band ratio and spectral flatness are also included. The band ratio describes a similar feature to 2-band spectral tilt with the only difference that band ratio yields a plain ratio instead of a log scale dB-like metric for frequency band 50-150Hz and 300-3000Hz. Note that this specific feature is calculated with the raw audio signal without going through a band pass filter. The last feature calculated is the spectral flatness, which is also based on the PSD calculated previously. The spectral flatness deals with the ratio between geometric mean and arithmetic mean of the PSD in the voice band (i.e. 300-3000Hz). The geometric mean of a specific dataset is given by:

$$\mathrm{GM} = \sqrt[n]{x_1 x_2 x_3 \dots x_n},$$

which is more sensitive to small values compared to the traditional arithmetic mean. In other words, if the voice band energy is dominated by the sound around a certain frequency, the geometric mean of PSD will be much smaller than the arithmetic mean and thus create a small spectral flatness value. On the contrary, if the voice energy follows a more even distribution along different frequencies, the flatness level will be higher. This parameter serves as an indicator for either a tone like or a noise like sound piece.

With regard to the window level that spans for 10 seconds, some features are simply calculated by aggregating the frame-level features while the others are not seen in the old feature set. For the VAF at window-level, it is aggregated by taking the mean of frame level VAF, and both band ratio and spectral flatness is done by taking the median of the corresponding frame level feature. One crucial factor that is newly proposed is the modulation energy at 3-8Hz which measures how much the speech-band energy fluctuates at a syllable-like rate. Instead of focusing on fluctuation of raw signal, this parameter puts

attention on the changing rate of loudness level. The amplitude envelope is calculated by taking the absolute value of the analytic signal of the raw produced by Hilbert transform, which is given by:

$$\hat{x}(t) = \mathcal{H}\{x(t)\} = \frac{1}{\pi}\mathrm{p.v.}\int_{-\infty}^{\infty}\frac{x(\tau)}{t-\tau}\mathrm{d}\tau,$$

and in frequency domain, the transform becomes:

$$\mathcal{F}\{\hat{x}(t)\} = -j\mathrm{sgn}(f)X(f),$$

$$\mathrm{sgn}(f) = \begin{cases} +1, & \mathrm{f} > 0 \\ 0, & \mathrm{f} = 0 \\ \text{-}1, & \mathrm{f} < 0 \end{cases}$$

where $\mathrm{p.v.}$ stands for the Cauchy principle value. This envelope is treated as a new signal and then down-sampled to around 100Hz for FFT, which yields the amplitude spectrum. The power between 3 and 8 Hz of the spectrum is integrated and then normalized by total envelope power, which directly quantifies the component of the loudness envelope that falls into 3-8Hz, which is also a typical syllabic rate of human speech. Intuitively, this parameter gives a direct reflection on the fraction of the loudness level that is changing at 3-8Hz, which a higher value typically implies that the sound is dominated by human speech from multiple speakers. The loudness level RMS as well as the "CrossCount" from range sensor is recalculated using the window data. The "CrossPerMinute" parameter is updated at each window corresponding to a rolling window fashion. Since there is no actual start and end timestamp for each passing event, each event is assumed to last for 0.8 seconds, and the dwell ratio is the proportion between total occluded time within a window and the total window length, which is 10 seconds. Finally, several features on temporal context are constructed to describe short term behavior changes. $\Delta\mathrm{Cross}$ and $\Delta\mathrm{VAF}$ deals with the change of the corresponding feature with respect to the last window. The RMS loudness of the audio as well as the VAF is smoothed using exponential moving average (EMA) filter, which is given by $x'_t = \alpha x_t + (1-\alpha)x_{t-1}$, where α is the EMA coefficient. These parameters equip the model with some ability to see what happens in the previous window and make adjustments accordingly.

3. Sensor Calibration

3.1. Calibration Procedure

The ultrasonic range sensor (RCWL-1601) was calibrated to evaluate its measurement accuracy, usable range, and robustness under varying atmospheric conditions. The sensor was installed with the UoTMIE1050 Rev1.0 electronic board based on an ESP32-S3 microcontroller, and distance measurements were compared against known reference distances obtained using a calibrated measuring tape as shown in Fig. 7. Calibration experiments were conducted across four distinct environments such as standard room, hot-dry, hot-humid, and cold-dry to capture the influence of temperature and relative humidity on the speed of sound. During calibration, the sensor and microcontroller were fixed in position while the target object was placed at varying distances, ensuring repeatable geometry and

alignment. Temperature and relative humidity were recorded concurrently, while atmospheric pressure was assumed constant due to sensor insensitivity and experimental constraints.

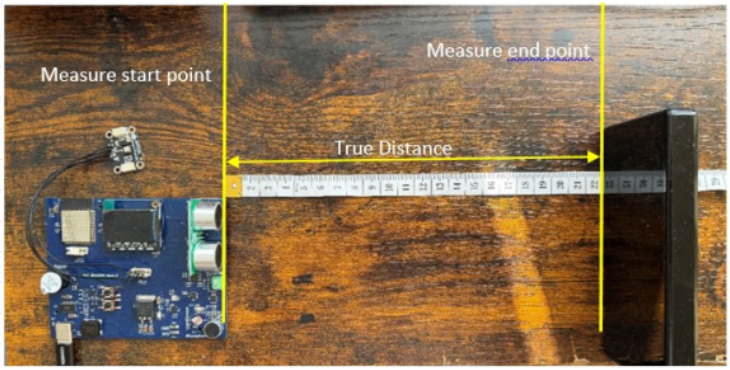


Figure 7. Ultrasonic range sensor calibration set up

For calibration of other atmospheric sensors, the sensor is placed in an indoor environment near the HVAC with the HVAC thermometer set as 73 Fahrenheit, which is equivalent to 22.8 Celsius. Since the sensor board is placed near the intake of the HVAC, it can be safely assumed that the temperature around the board is kept stable, which corresponds to the readings of the internal thermometer from the HVAC. The window of the room is kept closed to eliminate any potential interruption coming from the weather changing. In addition, human activities are avoided to keep the testing environment undisturbed and ensure the accuracy of calibration experiments. The experiments lasted for about 3 hours, while the data was collected and stored in csv files. Plots are generated from the calibration data for inspection.

3.2. Calibration Result

The calibration model accounts for environmental effects on the speed of sound using Cramer's equation, enabling correction of time-of-flight measurements prior to distance estimation. After compensating for temperature and humidity effects, a polynomial regression model was fitted between the corrected ultrasonic distance and the true reference distance to mitigate sensor drift and random noise. A fourth-order polynomial was selected as an optimal balance between model complexity and accuracy, yielding an adjusted R^2 of 0.9993 and a standard error of 6.79 mm. Post-calibration validation showed a typical accuracy of approximately ±7 mm, with a mean bias of 0.32 mm and a 95% confidence interval of ±13.6 mm. The calibrated sensor demonstrated reliable performance over a range of approximately 20 mm to 3.5 m, which comfortably satisfies the requirements for doorway crossing

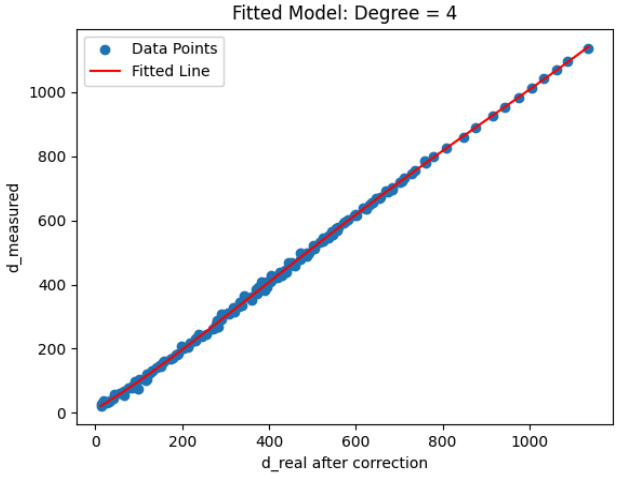


Figure 8. Collected data points scattered plot with fitted line

detection rather than precise absolute ranging. The data points as well as the fitted line is shown in Fig. 8.

The BME680 environmental sensor was evaluated to determine its suitability for occupancy estimation based on changes in gas concentration, temperature, and relative humidity. The temperature output was compared against this reference condition using the recorded time-series data shown in Fig. 9. The BME680 shows temperature, humidity, and gas signals exhibit slow response dynamics characterized by gradual transients and long settling times governed primarily by HVAC cycling and air mixing effects as shown in Fig. 3. From the calibration data, the temperature sensor requires about 30 minutes to become stable and gas readings require about 15 minutes to settle down, which means only readings after these time thresholds are meaningful and reflective of the real environment. Note that there is a drop in both gas sensor and temperature sensor which are caused by unknown reason, and therefore only the stable readings are considered for discussion. By inspecting the calibration data, it can be seen that the temperature and gas readings mostly remain steady after the transient state, which means the fluctuation seen in Fig. 3 is caused by the environment. However, time-series analysis showed that these signals exhibited slow dynamics dominated by HVAC operation resulting in weak correlation with short-term occupancy changes. Although the sensor readings were internally consistent, their low temporal responsiveness made them unsuitable for detecting rapid transitions between occupancy states.

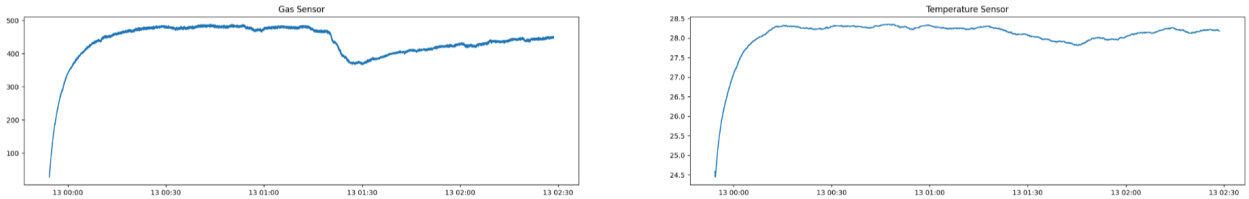


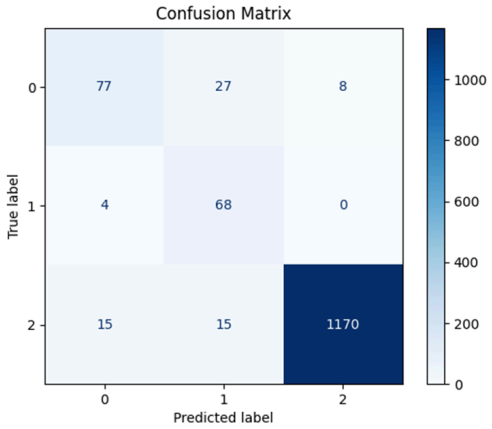
Figure 9. BME680 sensor outputs for calibration

4. Test Results

Before diving into the training and test results of the system, it is worth mentioning that since the label is collected in head counts with total occupancy recorded, the labels are first mapped to integers representing the occupancy level: 0-low, 1-medium, and 2-high. The occupancy level is defined as such: low occupancy level between 0 occupant and one third of the total occupancy; medium occupancy level between one third occupants and two thirds of the total occupancy; high level between two thirds of full and completely full. Also, in order to accommodate for the training samples that span 10 seconds each, the manually collected head count with irregular time interval is unsampled by forward filling method into the same frequency as the decision window. In other words, if a head count is recorded as medium at 5 min and a new one is recorded as high at 7 min, the labels will be filled as medium start from 5 minutes until 7 minutes. As a result, the label information is properly aligned with the training samples, and they are guaranteed to share the same length. Also noted that the sample localization and alignment is completed by explicitly including the center second of each window as a timestamp, which is convenient for debugging and visualization. A dataframe is built using the aforementioned features for both audio and range sensor, where each sample can be seen as a 11-dimensional row vector describing the following features: "VAF", "BandRatio", "SpecFlat", "Mod3_8Hz", "RMS_dB", "CrossCount", "DwellRatio", "DeltaVAF", "DeltaCross", "EWMA_VAF", "EWMA_RMSdB", and each feature is normalized to [0,1] for better converging in the machine learning model. With regard to the machine learning model, a

multimodal logistic regression model is used to handle the temporal context and provide an embedded temporal awareness for post-processing. A L2 regularization is used to prevent overfitting and a softmax activation function is used for final prediction probabilities. Class weights are introduced in model initialization to handle the class imbalance scenario in the data. During inference, the model will directly take-in the pretrained parameters and the audio data as well as ranger sensor readings will be fed into the pipeline on a per minute basis. Note that since the system aims at make predictions on the changing occupancy level for a complete lecture, during training and testing, instead of splitting the experimental data recorded for an entire lecture, 2 sets of data from complete lectures will be used for training and validation, while the third lecture data will be treated as the test set. For simplicity and consistency, all three sets of data are collected in the same room. Instead of directly selecting the class with the largest softmax probability as the result, the raw predicted probabilities are passed through an EMA-based filter which leverages information provided by the "CrossPerMin" feature from the range sensor. In this filter, the probabilities are calculated by $x'_t = \alpha x_t + (1-\alpha)x_{t-1}$, where α is an adaptive EMA coefficient controlled by the "CrossPerMin" feature. Namely if the "CrossPerMin" feature stays below a certain threshold, a larger EMA coefficient (α = 0.85) will be applied to increase the hysteresis of the system, which drives entire system to largely rely on the prediction at the previous step. On the contrary, when people are frequently passing by the range sensor, the EMA coefficient will be reduced to 0.15, allowing the system quickly to adapt to the changing environment. This specific filter proves to be effective in reducing flickering results and increasing system adaptability. After the filter, a simple state-aware algorithm is implemented to provide a contextual background to the system—it will make decisions using probabilities from multiple classes given the information of the previous state. For example, it will enter from low to medium level when the sum probability of medium and high exceeds a certain limit and will decrease from medium to low if the sum drops below another threshold. Similar gating mechanisms are also carried out for the case of high-medium. Note that no jumping and diving between windows is allowed since there is little probability for the occupancy level to increase from low to high or drop from high to low within 5 seconds. By coupling the EMA filter and state-awareness gates, the model is told with current status and can react promptly and accurately to the changing environment and is able to adapt to the cases of both entering and leaving. The performance of the trained pipeline is evaluated on the post-possessed predictions, and the confusion matrix as well as the precision, recall, and F1 score for the test set is shown below in Fig. 10:

	Precision	***Recall***	***F1 Score***
Low	0.802	0.688	0.74
Medium	0.618	0.944	0.747
High	0.993	0.975	0.984



The classification label and predicted labels are expressed in 0,1, and 2, which corresponds to low, medium and high levels of occupancy. It could be observed that the model performs ideally on high-level predictions, with a F1 score of 0.984 for high occupancy cases. On the other hand, the performance drops for low and medium predictions. Specifically, 27 of low occupancy samples are mistaken for medium, which drags down the recall on low and precision on medium. The difficulty in getting stable predictions on low and medium could be resulted from the relatively low number of training samples as well as the atypical behavior of the audio samples during low or medium stages. Also, when the lecture ends and people are exiting the room, only a small time interval will be regarded as medium level occupancy, which could potentially add more uncertainty to the prediction. Finally, when the lecture ends, the crowd will start moving and packing up belongings which will create a large noise that overrides the speech noise and thus disturb the predictions that are largely depended on audio characteristics and such examples are shown in a visualization program and illustrated in Fig. 11:

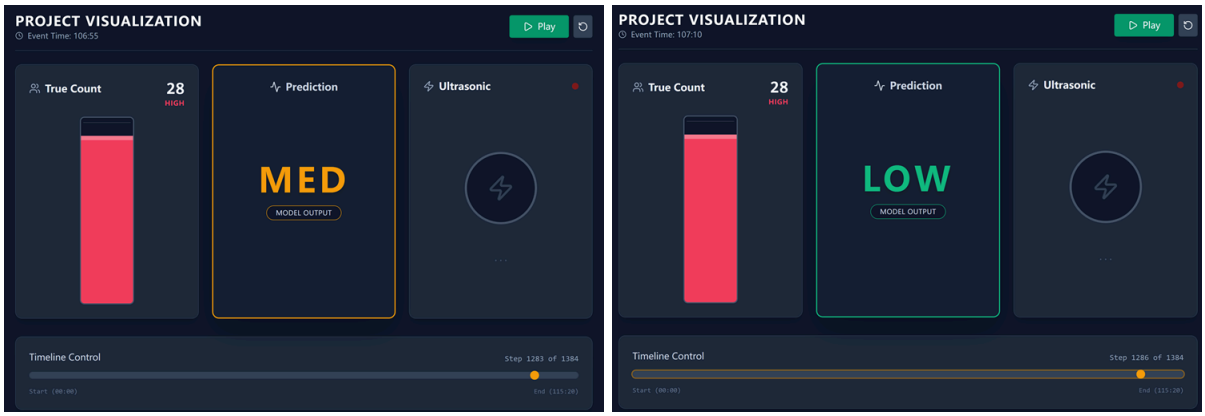


Figure 11. Illustration of test results

5. Conclusion

The project proposed a practical contact-free occupancy detector for classroom environments achieved by fusing acoustic sensing with crossing events yielded from ultrasonic range sensors to infer categorical occupancy levels (Low/Medium/High). This approach allows easy integration with HVAC systems, automatic lighting controls, and room-usage monitoring tools, resulting in better overall energy savings. If integrated with HAVC, the system is able to dynamically adjust HVAC settings such as airflow and temperature based on detected occupancy levels. Compared to traditional schedule-based or fixed HVAC controls, this detector responds much faster to actual room usage, achieving 20 to 40% energy reduction while maintaining or improving occupant comfort.

During system prototyping, a data-driven sensor selection process demonstrated that slow-response environmental signals and low-SNR onboard/external microphone data streams were not suitable for occupancy classification that requires a quick adaptation to the environment. Hence, the final pipeline switched to laptop-recorded audio and ultrasonic crossings for speech analysis and dynamic flow detection. On the sensing side, the ultrasonic range sensor was calibrated with environmental compensation as well as polynomial fitting, which achieves an adjusted R^2 of 0.9993 and an accuracy of approximately ±7 mm, which is sufficient for robust doorway event detection. With regard to prediction

and inference, a multimodal logistic regression model followed by adaptive EMA smoothing and a state-aware gating algorithm is implemented to improve stability while maintaining the responsiveness during rapid entry/exit periods. The final test results demonstrate strong performance for high-occupancy detection, while dropping to a moderate performance for medium and low level where most errors came from low samples being predicted as medium.

5.1. Limitations and Future Works

Some limitations are observed on distinguishing between entry and exit when only relying on the ultrasonic sensor, which means the speech information is the only source to support predictions for state identification. This requires the machine learning model to capture meaningful features from audio samples which has a strong correlation to the medium and low level occupancy, and this is precisely where the current model is struggling. Limited by fewer diverse low or medium training examples as well as the non-speech noise during end-of-lecture which is highly likely to override speech cues and thus disturb audio-driven features, the model sees more inaccurate predictions during the preparation and packing stage of a lecture. The ultrasonic readings generally react similarly for single passes or multiple passes, especially when a group of people cross by the sensor in a row. In the current algorithm, a passing event stemming from a valid reading corresponds strictly to a single person, whereas such an assumption can be misleading if multiple people pass by at the same time or if the range sensor didn't react promptly.

Based on the limitations mentioned above, some improvements could focus on collecting more varied datasets that span across multiple rooms with different class styles with more microphones introduced on different positions of the classroom. In addition, increasing the sampling rate of the ranger sensor can be useful in identifying the start and end time of a certain passing interval, which provides valuable contextual information on the crossing event and thus yield a more accurate and reflective estimation of the flow of people. Extra measures that concentrate on explicit movement or noise detection as well as the development of more advanced temporal models can be employed to add robustness to transient non-speech events without adding too-much cost to the system.

Reference

[1] “Pulse-code modulation,” Wikipedia, https://en.wikipedia.org/wiki/Pulse-code_modulation (accessed Dec. 14, 2025).
[2] “Threading - thread-based parallelism,” Python documentation, https://docs.python.org/3/library/threading.html (accessed Dec. 14, 2025).
[3] S. Butterworth, “On the Theory of Filter Amplifiers,” *Experimental Wireless & the Wireless Engineer*, vol. 7, pp. 536–541, Oct. 1930.
[4] J. W. Cooley and J. W. Tukey, “An algorithm for the machine calculation of complex Fourier series,” *Mathematics of Computation*, vol. 19, no. 90, pp. 297–301, Apr. 1965, doi: 10.1090/S0025-5718-1965-0178586-1.
[5] R. B. Blackman and J. W. Tukey, “The measurement of power spectra from the point of view of communications engineering—Part I,” *Bell System Technical Journal*, vol. 37, no. 1, pp. 185–282, Jan. 1958.
[6] J. Sohn, N. S. Kim, and W. Sung, “A statistical model-based voice activity detection,” IEEE Signal Process. Lett., vol. 6, no. 1, pp. 1–3, Jan. 1999.
[7] S. Dubnov, “Generalization of spectral flatness measure for non-Gaussian linear processes,” IEEE Signal Process. Lett., vol. 11, no. 8, pp. 698–701, Aug. 2004.
[8] Y. Ma and A. Nishihara, “Efficient voice activity detection algorithm using long-term spectral flatness measure,” EURASIP J. Audio, Speech, Music Process., vol. 2013, no. 87, pp. 1–15, Jul. 2013.
[9] L. Ding and S. S. Narayanan, “Cepstrum derived from differentiated power spectrum for robust speech recognition,” Speech Commun., vol. 41, nos. 2–3, pp. 469–484, Oct. 2003.
[10] S. Leong, “Acoustic correlates of the syllabic rhythm of speech: Modulation spectrum or local features of the temporal envelope,” Hear. Res., vol. 428, 108690, 2023.
[11] B. Ludusan and P. Wagner, “Speech, laughter and everything in between: A modulation spectrum-based analysis,” in *Proc. 10th Int. Conf. Speech Prosody 2020*, Tokyo, Japan, May 2020, pp. 995–999, doi: 10.21437/SpeechProsody.2020-203.
[12] S. W. Roberts, “Control Chart Tests Based on Geometric Moving Averages,” *Technometrics*, vol. 1, no. 3, pp. 239–250, Aug. 1959, doi: 10.1080/00401706.1959.10489860.
[13] Cramer, O. (1993). “The variation of the specific heat ratio and the speed of sound in air with temperature, pressure, humidity, and CO_2 concentration.” *Journal of the Acoustical Society of America*, 93(5), 2510–2516.
[14] K. J. Burch, “Measurement and Calibration,” in Measurement and Instrumentation: *Theory and Application*, 2nd ed.