electronics

MDPI

Article

Gas Leakage Detection Using Tiny Machine Learning

Majda El Barkani [1,*], Nabil Benamar [1,2], Hanae Talei [1] and Miloud Bagaa [3,*]

[1] School of Science and Engineering, Al Akhawayn University, Hassan II Avenue, Ifrane 53000, Morocco; n.benamar@aui.ma (N.B.); h.talei@aui.ma (H.T.)
[2] School of Technology, Moulay Ismail University of Meknes, Meknes 50050, Morocco
[3] Department of Electrical and Computer Engineering, University of Quebec at Trois-Rivières, Trois-Rivieres, QC G8Z 4M3, Canada
* Correspondence: m.elbarkani@aui.ma (M.E.B.); miloud.bagaa@uqtr.ca (M.B.)

Abstract: Gas leakage detection is a critical concern in both industrial and residential settings, where real-time systems are essential for quickly identifying potential hazards and preventing dangerous incidents. Traditional detection systems often rely on centralized data processing, which can lead to delays and scalability issues. To overcome these limitations, in this study, we present a solution based on tiny machine learning (TinyML) to process data directly on devices. TinyML has the potential to execute machine learning algorithms locally, in real time, and using tiny devices, such as microcontrollers, ensuring faster and more efficient responses to potential dangers. Our approach combines an MLX90640 thermal camera with two optimized convolutional neural networks (CNNs), MobileNetV1 and EfficientNet-B0, deployed on the Arduino Nano 33 BLE Sense. The results show that our system not only provides real-time analytics but does so with high accuracy—88.92% for MobileNetV1 and 91.73% for EfficientNet-B0—while achieving inference times of 1414 milliseconds and using just 124.8 KB of memory. Compared to existing solutions, our edge-based system overcomes common challenges related to latency and scalability, making it a reliable, fast, and efficient option. This work demonstrates the potential for low-cost, scalable gas detection systems that can be deployed widely to enhance safety in various environments. By integrating cutting-edge machine learning models with affordable IoT devices, we aim to make safety more accessible, regardless of financial limitations, and pave the way for further innovation in environmental monitoring solutions.

Keywords: CNN; cost-effective IoT solutions; EfficientNet-B0; IoT; MobileNetV1; TinyML

check for updates

Citation: El BarkaniEl Barkani, M.; Benamar, N.; Talei, H.; Bagaa, M. Gas Leakage Detection Using Tiny Machine Learning. *Electronics* **2024**, *13*, 4768. https://doi.org/10.3390/electronics13234768

Academic Editor: Marcin Witczak

Received: 21 October 2024
Revised: 21 November 2024
Accepted: 26 November 2024
Published: 2 December 2024

1. Introduction

The Internet of Things (IoT) has emerged as a transformative technology that fundamentally reshapes our interaction with the digital world. By seamlessly integrating sensors and actuators into everyday objects, the IoT has enabled a new era of connectivity and real-time data collection in various sectors. From enhancing home automation to revolutionizing industrial operations, the deployment of IoT technologies promises improved efficiency and responsiveness in real-time applications [1].

The merging of AI with the Internet of Things (IoT) expands the potential of IoT systems. This connection enables advanced data analytics and decision-making processes at network edges, eliminating the latency and bandwidth constraints associated with cloud computing. The AIoT not only improves decision making based on IoT data but also enables complex applications like predictive maintenance and smart energy management [2]. Despite the numerous benefits, deploying AIoT systems is not without problems. Some of the significant challenges include high implementation costs, complexity in integrating AI with current IoT infrastructures, and concerns about data privacy and security. Furthermore, the continual functioning of these systems requires sustainable energy, which can be a limiting factor, especially in environments where power availability is constrained [1,2].

Tiny machine learning (TinyML) has emerged as a potent solution to these challenges. It represents a fast-growing field of machine learning technologies and applications that are capable of performing on-device analytics at extremely low power. This capability is crucial for extending the battery life of IoT devices and reducing latency in data processing. TinyML enables the embedding of intelligence in the tiniest devices, making it feasible to process data directly on the device rather than relying on cloud services [2,3].

Recent studies further underscore TinyML's transformative impact on various applications. For instance, Abadade et al. [4] provided a comprehensive survey of TinyML, highlighting its advantages of energy efficiency, low latency, and privacy. The survey discussed TinyML's suitability for resource-constrained devices in sectors such as healthcare, environmental monitoring, and anomaly detection, showing how it brings machine learning capabilities to edge devices in diverse fields. Another study by Lin et al. [5] explored the challenges associated with deploying deep learning models on microcontrollers due to memory limitations. They emphasized the need for system-algorithm co-design to make TinyML feasible for real-time applications on ultra-low-power devices, enabling impactful IoT deployments across various industries.

In addition, recent research by Pimpalkar and Niture [6] presented a TinyML-based contactless elevator system that utilizes convolutional neural networks (CNNs) for person detection and keyword spotting. This innovative approach highlights TinyML's role in public health and safety applications by reducing physical contact, which is essential in contexts where hygiene and convenience are paramount. The study serves as a practical example of how TinyML can be implemented in real-world systems with low computational overhead and minimal infrastructural changes.

TinyML provides an efficient framework for deploying machine learning models on cost-effective and eco-friendly platforms such as the Arduino Nano 33 BLE. By leveraging techniques such as model quantization and on-device learning, TinyML ensures that real-time analytics can be performed while minimizing power consumption, making it a crucial technology for safety applications like gas leak detection. These techniques, including quantization and pruning, are integral in reducing model size and computational requirements, allowing deployment on resource-constrained devices without sacrificing accuracy [7,8].

Specifically, in the realm of safety and environmental monitoring, TinyML can be instrumental [9]. Gas leaks, particularly in industrial settings or residential areas, pose significant risks, including explosions and health hazards due to toxic exposures. Traditional gas detection systems often rely on central processing and are not capable of real-time responses. Utilizing TinyML, we propose a novel approach to gas leak detection, where IoT devices can instantly analyze environmental data and provide alerts without needing to communicate with a central server. For instance, research in acoustic monitoring has shown that embedding CNN-based TinyML models into edge devices can achieve high accuracy and low latency for detecting critical anomalies in sensor data [7]. This method enhances the speed and reliability of gas leak detection, which is crucial for safety and preventive measures [1].

To address the critical need for efficient, real-time gas leak detection in resource-constrained environments, this research aims to design and implement a TinyML-based gas detection system. The primary goal is to leverage low-power, edge-based machine learning models to achieve high accuracy and quick response times in detecting various gas types. Specifically, we focus on integrating the MobileNetV1 and EfficientNet-B0 models with the Arduino Nano 33 BLE Sense and the MLX90640 thermal camera to create a scalable, cost-effective solution for industrial and residential applications. This work not only demonstrates the feasibility of deploying TinyML models for safety monitoring but also contributes to advancing IoT-based environmental monitoring technologies by providing a low-cost, reliable solution for gas leak detection.

This study addresses three key research questions (RQs) on leveraging TinyML for efficient, real-time gas leak detection:

- **RQ1:** How effectively can TinyML models improve detection in resource-limited environments?
- **RQ2:** What balance can be achieved between accuracy, speed, and efficiency on edge devices?
- **RQ3:** How well do optimized models, such as MobileNetV1 and EfficientNet-B0, perform under various conditions and hardware setups?

These questions aim to bridge the gap between TinyML's theoretical advancements and practical, cost-effective safety solutions.

The remainder of this paper is organized as follows. Section 2 reviews related works, highlighting advances in the applications of IoT and TinyML for environmental safety. Section 3 describes the materials and methods, detailing the experimental setup for TinyML-based gas leak detection. Section 4 presents our findings, discussing the performance of the proposed system under various conditions. Finally, Section 5 concludes this paper with a summary of the obtained results and future research directions.

2. Related Works

Numerous studies have explored using machine learning models and TinyML technologies to enhance gas leak detection in various settings, showcasing advances in identifying leaks with precision and improving safety measures effectively.

Kopbayeva et al. [10] introduced an innovative method to detect and diagnose natural gas leaks, employing a combination of convolutional networks and bidirectional long- and short-term memory (BiLSTM) networks. The model impressively achieved a 92% accuracy rate in predicting leaks and accurately classifying their size, with just a 5% average error margin. However, the reliance on simulated data poses limitations and could overlook real-world intricacies. Furthermore, while the model effectively detected leaks within a 50-s time frame, its accuracy waned with shorter intervals, dropping below optimal levels when the window was reduced to less than 26 s.

Miao et al. [11] used unsupervised learning with the Wasserstein generative adversarial network with gradient penalty (WGAN-GP) and the Bayesian Gaussian mixture model (BGMM) to analyze real-time stress signals obtained from residual magnetic effects. WGAN-GP extracts features from normal stress data, which are then inputted into the BGMM to assess pipeline health via the WLP indicator. The results indicated that the relative error of the stress prediction model was within 3%, and the weighted logarithm probability (WLP) effectively distinguished between normal and leakage conditions. However, limitations include the need for further evaluation under high pressure and external interference conditions and consideration of additional potential leakage causes beyond those addressed in the model.

In their study, Doshmanziari et al. [12] introduced a model-based fault detection framework for gas pipeline leak detection, employing sensor fusion and a state-space representation approach. Using the extended Kalman Filter (EKF) for state observation, the methodology requires hydraulic pressure measurements at various points and recommends using sensor arrays over single sensors for improved leak estimation accuracy. When validated through simulations using the OLGA software, the approach effectively identified and located leaks, demonstrating notable precision. However, the performance is sensitive to factors such as spatial discretization, initial conditions, leak timing, and sampling rate, indicating areas for potential improvement.

Du et al. [13] developed a deep learning framework for detecting underground natural gas micro-leaks using hyperspectral imagery, grounded in a multi-branch deep learning network (Img-Spec-PGE model) that estimates the plant leaf area index (LAI) under gas stress. Despite achieving high LAI estimation accuracy and validating the method's efficacy through realistic simulation models, the study faced notable limitations. The approach's performance is particularly sensitive to the selection of spectral bands, implying that its applicability may be constrained across different vegetation types or environmental conditions.

In the study by Gkogkidis et al. [9], a novel application of TinyML technology was showcased through the development of a system designed for the real-time detection of hazardous gas leaks, utilizing edge computing for data processing directly on the device. This approach circumvents traditional privacy and data transmission concerns associated with IoT devices by enabling local analysis of sensor data to identify gas leaks or smoke presence. Despite achieving promising detection outcomes, with F1 scores of 0.77 for smoke and 0.70 for ammonia, the study acknowledged significant limitations, particularly the challenges associated with offline training or retraining of the system, security vulnerabilities in data transmission, and the potential need for more specialized sensors to enhance detection accuracy.

In their innovative research, Lorthong et al. [2] employed an Artificial Neural Network (ANN) within an IoT-based system to predict the risk of LPG gas leaks, harnessing environmental data captured by sensors. This ANN model, crucial for processing variables like LPG concentration, temperature, humidity, and oxygen levels, effectively identified complex patterns and relationships to classify leakage risk. Achieving an impressive accuracy rate of 96.05%, the model underscores the potential of machine learning in enhancing safety mechanisms in industrial settings. However, the research highlighted limitations, notably the dependency on high-quality, real-time sensor data and the challenges of applying the model across diverse and dynamic industrial environments.

In the study by Narkhede et al. [14], a multimodal AI-based sensor fusion approach was employed for gas detection, integrating a seven-semiconductor gas sensor array with a thermal camera. This early fusion technique achieved a testing accuracy of 96%, significantly outperforming individual sensor models (82% for gas sensors and 93% for thermal images). However, the method's reliance on both gas sensors and thermal imaging introduced complexity and increased costs, making it less practical for widespread deployment. The necessity of using both types of sensors complicates the system, and it can be prohibitively expensive, highlighting the need for more streamlined and cost-effective solutions in gas detection technology.

Wang et al. [15] introduced the RGB-Thermal Cross-Attention Network (RT-CAN) for gas detection, combining RGB and thermal imaging to enhance detection accuracy by 4.86%. They also developed Gas-DB, a dataset containing 1300 RGB-thermal images for training and evaluation. However, the dual-camera requirement increases costs and complexity, limiting the model's use in constrained settings. Additionally, the model's reliance on RGB data could impact performance in low-light conditions, and Gas-DB's limited environmental diversity may reduce generalizability.

Attallah and Elhelw [16] proposed a gas leak detection pipeline using multiple CNNs and thermal imaging, achieving 98% accuracy. This method enhances detection but requires high computational power, making it less suitable for real-time or low-power devices. The reliance on thermal imaging alone may reduce accuracy in complex environments where heat interference is present, limiting its adaptability across various gas types.

Sharma et al. [17] applied a multimodal approach and federated learning for gas detection, achieving a 96% accuracy rate while maintaining data privacy. However, federated learning depends on reliable network connectivity, which could limit deployment in remote locations. The complexity of processing multimodal data and the specialized setup for privacy-preserving frameworks may also impact scalability.

Tsoukas et al. [18] developed a TinyML-based gas detection device for real-time monitoring, achieving F1 scores of 0.77 for smoke and 0.70 for ammonia. While effective, the offline setup restricts model updates, and limited accuracy may reduce sensitivity in critical applications. Additionally, the system's focus on specific gases (smoke and ammonia) limits broader industrial use.

Shafin et al. [19] introduced a custom CNN-BiLSTM model specifically designed for gas leak detection on resource-limited IoT devices. The model, developed without using pre-trained networks, achieved a compact size of 575 KB and an efficient inference time of 0.255 ms. The combination of CNN layers for feature extraction and BiLSTM layers for temporal pattern recognition allowed the model to detect gas leaks accurately while maintaining low latency, making it suitable for real-time safety applications in edge environments where power and computational resources are limited. Both pre-trained and custom-designed models can be beneficial for such applications, with the choice depending on the specific deployment constraints and performance requirements.

In conclusion, the related works highlight the diversity of approaches in gas leak detection, as detailed in Table 1, which summarizes the models' accuracies and notes significant limitations such as dependency on simulated data, high computational demands, and sensitivity to environmental conditions. This comparison emphasizes the necessity of addressing these constraints to enhance the practicality and effectiveness of gas leak detection technologies in real-world applications.

Table 1. Comparison of gas leak detection models.

Reference	Model/Method	Accuracy	Notes
Kopbayeva et al. (2022) [10]	CNN & BiLSTM	92%	Effective for predicting leaks and classifying size, but reliant on simulated data.
Miao et al. (2022) [11]	WGAN-GP BGMM	Within 3% error	Suitable for stress signal analysis; needs further validation under high-pressure conditions.
Du et al. (2024) [13]	Multi-branch DL (Img-Spec-PGE)	High for LAI estimation	Hyperspectral imagery sensitive to spectral band selection; applicable to vegetation types under gas stress.
Gkogkidis et al. (2022) [9]	TinyML Edge Detection	F1 scores of 0.77 (smoke), 0.70 (ammonia)	Challenges in offline retraining and security vulnerabilities.
Lorthong et al. (2023) [2]	ANN-based IoT System	96.05%	Dependence on high-quality, real-time sensor data limits scalability across environments.
Narkhede et al. (2021) [14]	Multimodal AI-based sensor fusion	96%	Combines gas sensors and thermal imaging, which increases cost and complexity.
Wang et al. (2024) [15]	RGB-Thermal Cross-Attention Network	4.86% Improvement	Requires dual cameras; reliance on RGB may impact low-light performance.
Shafin et al. (2023) [19]	Custom CNN and BiLSTM	-	Compact model suitable for IoT deployment without pre-trained weights.
Attallah and Elhelw (2023) [16]	Multiple CNNs + Thermal Imaging	98%	High computational demand; challenging for low-power devices.
Sharma et al. (2024) [17]	Multimodal Federated Learning	96%	Maintains data privacy; depends on network connectivity.
Tsoukas et al. (2023) [18]	TinyML-based Monitoring Device	F1 score: 0.77 (smoke), 0.70 (ammonia)	Offline setup limits model updates; focused on specific gases.

3. Materials and Methods

The following section outlines the materials and methods employed in the study, focusing on data acquisition, preprocessing, and the design of our gas leak detection CNN model using the Edge Impulse platform. Our approach leverages advanced machine learning techniques and state-of-the-art hardware to develop an efficient, accurate, and deployable gas leak detection system.

3.1. Data Acquisition and Preprocessing

In this work, we used an existing dataset curated by Narkhede et al. [14] known as the “MultimodalGasData” dataset. This dataset is a comprehensive collection designed specifically for gas detection and classification tasks. It consists of both thermal image data, which capture temperature variations, and numerical values from gas sensors. This multimodal approach, combining thermal images with sensor measurements, enhanced the dataset’s richness and utility, offering a more comprehensive view of the gas leak scenarios.

The experimental setup for data collection was meticulously designed to ensure precise and consistent data acquisition. The setup involved positioning seven metal-oxide gas sensors (MQ2, MQ3, MQ5, MQ6, MQ7, MQ8, and MQ135) alongside a Seek Compact Thermal Imaging Camera (UW-AAA), as shown in Figure 1. These sensors were selected due to their sensitivity to a range of gases, as shown in Table 2. The sensors were arranged with a 1 mm separation between each to ensure uniform exposure to the gas sources, thereby maintaining consistency across the dataset.

Table 2. Gas sensors and sensitive gases.

Sensor	Sensitive Gas
MQ2	Methane, Butane, LPG, Smoke
MQ3	Alcohol, Ethanol, Smoke
MQ5	Natural Gas, LPG
MQ6	LPG, Butane Gas
MQ7	Carbon Monoxide
MQ8	Hydrogen Gas
MQ135	Air Quality (Benzene, Smoke)

To simulate realistic gas leak scenarios, perfume spray and incense sticks were chosen as gas sources. The perfume spray, containing 95% alcohol, and incense sticks, emitting a mixture of carbon monoxide, carbon dioxide, nitrogen dioxide, and sulfur dioxide, served as proxies for common indoor pollutants. The controlled release of these gases provided a dynamic environment, emulating potential leak situations found in industrial and residential settings.

The data-logging process was structured to capture comprehensive and high-resolution data. Measurements from the gas sensors and thermal images were recorded at 2 s intervals over a 90-minute period. This frequent logging ensured that transient changes in the gas concentration and temperature were accurately captured, providing detailed temporal data essential for training robust machine learning models.

The release of gases was carefully regulated to ensure consistency. During the first 30 min, gases were introduced every 15 s, then changed to 30 s for the subsequent 30 min, and then the interval was increased to 45 s in the last 30 min. This systematic approach ensured the dataset encompassed a wide range of gas concentrations and thermal signatures, simulating both rapid and gradual environmental changes.

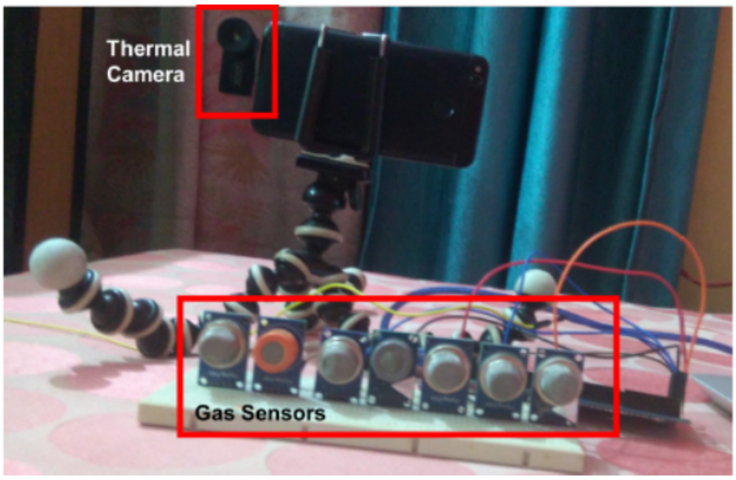
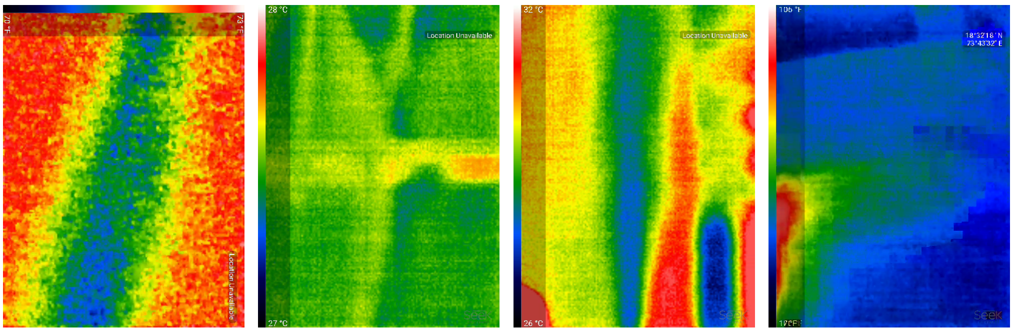


Figure 1. Experimental setup for data collection (reprinted, with permission, from [20] @2022 MDPI).

The Seek Compact Thermal Imaging Camera was integral to the data acquisition process, capturing thermal images that reflect temperature variations caused by gas leaks. The camera, with a resolution of 206 × 156 pixels and a temperature measurement range from −40 to 330 °C, is capable of detecting thermal anomalies associated with gas emissions. These images provided crucial visual data that could be processed to identify gas leaks in visually obscured or inaccessible environments.

The resulting dataset, "MultimodalGasData", comprises 6400 samples divided equally into four classes: smoke, perfume, a mixture of smoke and perfume, and a neutral environment, as shown in Figure 2. Each sample includes both thermal images and corresponding gas sensor measurements, ensuring a robust and diverse dataset for training and validating gas leak detection models.



(**a**) (**b**) (**c**) (**d**)

Figure 2. The four dataset categories: (**a**) Mixture; (**b**); No Gas, (**c**); Perfume; (**d**) Smoke.

For the purpose of this article, we only considered the thermal image data taken from the "MultimodalGasData" dataset. This choice was prompted by our desire to avoid the requirement for sensors in conjunction with the thermal camera. Thermal imaging has a number of benefits, including the capacity to identify gases in visually obscured or inaccessible areas, which is critical for early warning systems in residential and industrial applications. To optimize for edge device deployment, each image was downscaled to 96 × 96 pixels, preserving key thermal details while reducing computational load. By focusing our efforts on thermal imaging data, we simplified the complexity of the sensor array and lowered the cost associated with the deployment of multiple sensor systems, making the technology available to a larger spectrum of users and applications.

3.2. Gas Leak Detection CNN Model

In this work, we implemented the convolutional neural network (CNN) model for gas leak detection using the Edge Impulse platform (https://edgeimpulse.com/, accessed on 28 November 2024), a machine learning model creation, training, and deployment platform for edge devices. Edge Impulse was selected for this project due to its robustness in deploying machine learning models on edge devices. The platform offers extensive capabilities for training, optimizing, designing, and preparing data. For our gas leak detection system, its transfer learning support and user-friendly interface made it the perfect option. The flexibility and scalability of the system can be ensured by effectively preprocessing data, optimizing models, and deploying them across a range of hardware platforms by using Edge Impulse.

Given that our dataset was in image format, we chose two models—MobileNetV1 and EfficientNet-B0—both of which are deployable on TinyML devices. MobileNetV1 [21] was selected for its lightweight architecture, allowing it to efficiently process images with minimal computational power, making it ideal for constrained environments. EfficientNet-B0 [22], while slightly more resource-intensive, offers higher accuracy and remains efficient enough for TinyML deployment. By utilizing both models, we ensured that our system could deliver accurate gas leak detection on a wide range of TinyML devices, from low-power devices to more capable hardware.

To ensure the models could be deployed efficiently on edge devices, we applied post-training INT8 quantization. Post-training quantization converts the model's weights and activations from 32-bit floating-point to 8-bit integers (INT8), significantly reducing the memory footprint and computation required for inference. This step was crucial for enabling the models to run within the memory constraints of the Arduino Nano 33 BLE Sense, which has only 256 KB of RAM.

Using the Edge Impulse platform's built-in quantization tool, we optimized the MobileNetV1 and EfficientNet-B0 models for embedded deployment. The quantization process reduced the models' sizes while ensuring that their performance remained close to that of the original floating-point models. Both models maintained high accuracy after quantization, making them viable for real-time gas leak detection on resource-constrained hardware.

For MobileNetV1, we used a version of the system with an input size of 96×96 pixels and a depth multiplier of 0.25 for our gas leak detection system. The selection of a 96×96-pixel input size for our CNN model was primarily driven by the balance between computational efficiency and model accuracy. This input size is small enough to meet the memory constraints of the Arduino Nano 33 BLE Sense, yet large enough to retain essential spatial information from the thermal images for effective gas detection.

The model was configured with specific settings tailored to our application, as shown in Table 3.

Table 3. Training configuration and settings.

Parameter	**Value**
Input Axes	Image
Image Dimensions	96×96 pixels
Color Depth	RGB
Training Settings	
Number of Training Cycles	45
Learning Rate	0.0005
Training Processor	CPU
Validation Set Size	20%
Batch Size	32
INT8 Model Profiling	Enabled

The neural network architecture started with an input layer consisting of 27,648 features. Each 96×96 RGB image resulted in an input layer consisting of 27,648 features

(96 × 96 × 3 channels), providing a sufficient level of detail for the CNN to learn distinctive features associated with different gas leak scenarios. The pre-trained MobileNetV1 was fine-tuned to our specific task. The optimization algorithm used was Adaptive Moment Estimation (Adam), known for its adaptive learning rate capabilities and efficient handling of sparse gradients. This optimizer combines the advantages of two crucial techniques: adaptive gradient estimation and momentum, which together improve convergence and stability during training.

In addition to the primary settings, we employed several key techniques to enhance model performance and generalization. We used dropout regularization with a rate of 0.5 in the fully connected layers to prevent overfitting. The ReLU activation function was used to introduce non-linearity into the model, and batch normalization was applied to stabilize and accelerate the training process.

The second model, EfficientNet-B0, is a state-of-the-art model known for its balance between model size, computational cost, and performance. However, the configuration differed slightly from the first model, as we were limited to a maximum of 10 training cycles. The adopted configuration is presented in Table 4.

Table 4. Training configuration and settings.

Parameter	Value
Input Axes	Image
Image Dimensions	96 × 96 pixels
Color Depth	RGB
Training Settings	
Number of Training Cycles	10
Learning Rate	0.0005
Training Processor	CPU
Validation Set Size	20%
Batch Size	32
INT8 Model Profiling	Enabled

The neural network architecture for EfficientNet-B0 started with an input layer of 27,648 features, and we utilized the pre-trained EfficientNet-B0 without the top layers. Similar to MobileNetV1, the Adam optimizer was used, leveraging its ability to effectively handle large-scale data and complex neural networks.

EfficientNet-B0 [22] incorporates several advanced features to enhance performance. The model employs Swish activation functions, which have been shown to improve accuracy compared to ReLU. Batch normalization is applied throughout the network to improve training speed and stability. The compound scaling technique, which balances depth, width, and resolution, ensures that the model is both efficient and effective for a wide range of applications.

Both MobileNetV1 and EfficientNet-B0 models were implemented using the Edge Impulse platform, leveraging transfer learning with pre-trained weights optimized for image recognition tasks. Transfer learning enables these models to retain critical feature extraction capabilities developed from large, general-purpose datasets, while fine-tuning adapts the parameters to detect the specific thermal patterns associated with gas leaks in our 'MultimodalGasData' dataset. During the fine-tuning process, the models' final layers were retrained to capture unique thermal image characteristics linked to gas emissions, such as subtle variations in heat signatures caused by different gas types. By using pre-trained architectures with depth multipliers (MobileNetV1 with a 0.25 depth multiplier and EfficientNet-B0's compound scaling) and subsequently fine-tuning, we achieved a model configuration that optimally balances accuracy with computational efficiency.

During training, we monitored key metrics such as loss and accuracy on the validation set to ensure that the models were learning effectively. We employed early stopping to prevent overfitting, halting the training process if the validation accuracy did not improve for 10 consecutive epochs.

By leveraging the MobileNetV1 and EfficientNet-B0 models on the Edge Impulse platform, we developed a robust and efficient gas leak detection system. MobileNetV1 is ideal for deployment on devices with limited resources due to its lightweight nature, while EfficientNet-B0 provides higher accuracy suitable for more capable hardware. This approach ensures that our gas leak detection system can be effectively deployed across a range of environments, from low-power edge devices to high-performance computing platforms. The combination of these models allows for versatile and scalable deployment, ensuring comprehensive coverage and reliability in detecting gas leaks in various settings. The use of advanced techniques, such as dropout regularization, batch normalization, and data augmentation, further enhances the performance and generalization of the models, making them well suited for real-world applications.

3.3. Target Deployment

The goal of this project is to deploy the gas leak detection system on an Arduino Nano 33 BLE Sense (Cortex-M4F 64 MHz, Arduino, Somerville, MA, USA), complemented by the MLX90640 thermal camera. This combination is particularly suitable due to the Arduino's compact size, low power consumption, and sufficient processing capabilities for edge machine learning tasks, alongside the MLX90640's ability to provide detailed thermal imaging.

The Arduino Nano 33 BLE Sense features an ARM Cortex-M4F processor running at 64 MHz, providing the computational power needed for our CNN models. Key characteristics include 256 KB of RAM and 1 MB of ROM, which are adequate for storing and running lightweight models like MobileNetV1 and EfficientNet-B0. The MLX90640 thermal camera offers a 32×24-pixel resolution and can measure temperatures ranging from -40 °C to 300 °C, making it ideal for detecting temperature variations associated with gas leaks.

The deployment of the gas leak detection model to the Arduino Nano 33 BLE Sense is facilitated by the Edge Impulse platform. The deployment process involves several steps, which are described below.

Model Training and Optimization: The CNN models (MobileNetV1 and EfficientNet-B0) are trained on the Edge Impulse platform. Transfer learning and post-training quantization techniques are used to ensure that the models are lightweight and efficient for deployment on resource-constrained devices.

Model Export: After training and validation, the models are exported as a fully optimized C++ library, which can be directly integrated into an Arduino sketch using the Edge Impulse SDK. This package includes all necessary code for real-time data acquisition, preprocessing, and running inference on the Arduino Nano 33 BLE Sense.

Integration with Arduino IDE: The exported model library is integrated into an Arduino sketch using the Arduino Integrated Development Environment (IDE). This integration involves including the model's header and source files in the project. The Edge Impulse SDK is used to facilitate data acquisition, preprocessing, and inference, which are also incorporated into the sketch.

Code Implementation: The Arduino sketch initializes the onboard camera and collects real-time data from the MLX90640 thermal camera. The collected data are preprocessed to match the input format required by the CNN models. The preprocessed data are then fed into the CNN model to perform inference and detect the presence of gas leaks. Based on the model's output, the system can trigger alarms, send notifications via the BLE, or activate connected devices to mitigate the gas leak.

Optimization Techniques: Using the Edge Impulse platform, we apply post-training quantization to convert the model's 32-bit floating-point parameters into 8-bit integers (INT8). This reduces the model's memory footprint, making it feasible for deployment on the Arduino Nano 33 BLE Sense while also speeding up inference. The quantized models require less memory and computation, resulting in faster predictions and reduced power consumption, which are critical for real-time applications on a microcontroller.

Testing and Validation: The integrated system is thoroughly tested to ensure accurate and reliable performance in real-world scenarios. This involves validating the model's predictions and the system's responsiveness under various environmental conditions and gas concentrations.

Deployment: Once validated, the system is deployed in the target environment. The compact size and low power requirements of the Arduino Nano 33 BLE Sense, along with the MLX90640 thermal camera, allow for easy installation in residential, industrial, or commercial settings where gas leak detection is critical.

By leveraging the Edge Impulse platform, the Arduino Nano 33 BLE Sense, and the MLX90640 thermal camera, we developed a robust, efficient, and scalable gas leak detection system. This deployment strategy ensures that our system can be effectively utilized across various environments, providing enhanced safety and early warning capabilities through the real-time monitoring and detection of gas leaks.

4. Results

To train and validate the CNN models, we employed a data split of 70% for training, 20% for validation, and 10% for testing. This approach ensured that the models received ample and diverse training data while providing a robust assessment of their generalization performance. Both the MobileNetV1 96 $\times$ 96 0.25 and EfficientNet-B0 models were evaluated through this process, and their performance metrics are detailed below.

The MobileNetV1 model demonstrated an accuracy of 90.2% on the validation set during the training phase, along with a matching loss value of 0.27. The validation set's confusion matrix shows how well the model performed in classifying the various kinds of gases. In particular, 99.6% of the Mixture class, 83.9% of the No Gas class, 88.4% of the Perfume class, and 89.1% of the Smoke class were accurately categorized by the model. For these classes, the F1 scores were 1.00, 0.84, 0.86, and 0.91, in that order.

MobileNetV1's efficiency was demonstrated by its on-device performance measurements, which showed an inference time of 1414 ms, a peak RAM utilization of 124.8 KB, and a flash usage of 304.1 KB. These measurements are essential for ensuring that the model can function well with the Arduino Nano 33 BLE Sense's constrained computing capabilities.

During the testing stage, the accuracy of the MobileNetV1 model was 88.92%. The model successfully categorized 99.4% of the Mixture class, 81.5% of the No Gas class, 84.5% of the Perfume class, and 90.2% of the Smoke class, according to the confusion matrix shown in Figure 3. The F1 scores were 1.00, 0.86, 0.88, and 0.92, in that order, showing good performance in every class, even if the accuracy was somewhat lower than during the training phase.

With a loss value of 0.28, EfficientNet-B0 demonstrated excellent training accuracy, attaining 89.5% on the validation set. EfficientNet-B0 accurately categorized 99.6% of the Mixture class, 83.5% of the No Gas class, 86.2% of the Perfume class, and 88.7% of the Smoke class, according to the confusion matrix. For these classes, the F1 scores were 1.00, 0.83, 0.85, and 0.91, in that order.

The on-device performance metrics for EfficientNet-B0 were similar to those of MobileNetV1, with an inference time of 1414 ms, a peak RAM usage of 124.8 KB, and a flash usage of 304.1 KB. These results confirm that EfficientNet-B0 is also well suited for deployment on resource-constrained devices.

During the testing phase, the EfficientNet-B0 model achieved a higher accuracy of 91.73%. The confusion matrix for the testing set showed that the model correctly classified 99.7% of the Mixture class, 81.5% of the No Gas class, 90.6% of the Perfume class, and 95.3% of the Smoke class. The F1 scores were 1.00, 0.88, 0.91, and 0.96, respectively, as shown in Figure 4, demonstrating the model's superior performance during testing compared to MobileNetV1. The confusion matrices utilize a color coding to visually delineate different accuracy levels: green for the highest percentages of correct predictions, pink for significant accuracies, light pink for moderate accuracies, and very light pink for lower accuracies and misclassifications.

Both MobileNetV1 96 × 96 0.25 and EfficientNet-B0 exhibited strong performance in gas leak detection. However, EfficientNet-B0 marginally outperformed MobileNetV1 in the testing phase, achieving an accuracy of 91.73% compared to MobileNetV1's 88.92%. EfficientNet-B0 also demonstrated higher F1 scores for most classes during testing, indicating better precision and recall, as shown in Table 5.

The resource usage and efficiency of both models were optimized using post-training quantization, which reduced their memory footprints and improved their inference speeds, as demonstrated in Table 6. This optimization is critical for deployment on the Arduino Nano 33 BLE Sense, which has limited computational resources. Despite their different architectures, both models demonstrated similar on-device performance metrics, making them suitable for deployment on the target hardware.

	MIXTURE	NOGAS	PERFUME	SMOKE	UNCERTAIN
MIXTURE	99.40%	0%	0%	0.3%	0.3%
NOGAS	0%	81.50%	7.50%	1.90%	9.10%
PERFUME	0%	4.70%	84.50%	3.70%	7.10%
SMOKE	0%	2.80%	0.30%	90.20%	6.60%
F1 SCORE	1	0.86	0.88	0.92	

Figure 3. Confusion matrix for the MobileNetV1 model.

	MIXTURE	NOGAS	PERFUME	SMOKE	UNCERTAIN
MIXTURE	99.70%	0%	0%	0%	0.30%
NOGAS	0.30%	81.50%	8.50%	1.30%	8.50%
PERFUME	0%	2.00%	90.60%	2.00%	5.40%
SMOKE	0%	2.50%	0.30%	95.30%	1.90%
F1 SCORE	1	0.88	0.91	0.96	

Figure 4. Confusion matrix for the BO configuration.

Table 5. Performance comparison of the MobileNetV1 and EfficientNet-B0 models during the validation and testing phases.

Model	Accuracy	Precision	Recall	F1 Score
MobileNetV1				
Validation	90.2%	91.8%	89.3%	90.2%
Testing	88.92%	90.6%	87.8%	88.9%
EfficientNet-B0				
Validation	89.5%	92.1%	90.0%	90.5%
Testing	91.73%	92.6%	90.7%	91.3%

Table 6. Resource usage metrics for the MobileNetV1 and EfficientNet-B0 models on the Arduino Nano 33 BLE Sense.

Model	Inference Time (ms)	Peak RAM Usage (KB)	Flash Usage (KB)
MobileNetV1	1414	124.8	304.1
EfficientNet-B0	1414	124.8	304.1

After applying post-training quantization, the memory footprint of both MobileNetV1 and EfficientNet-B0 was significantly reduced, allowing the models to fit within the 256 KB RAM of the Arduino Nano 33 BLE Sense while still maintaining high accuracy. The quantized versions of both models continued to provide robust performance, making them suitable for real-time gas leak detection on resource-constrained devices.

The inference time was measured at 1.4 s, well within the acceptable range for real-time gas leak detection. The quantized models significantly reduced latency and power consumption, which are crucial factors when deploying machine learning models on low-power edge devices

Given the slightly higher accuracy and F1 scores of EfficientNet-B0 during testing, it may be the preferred model for deployment in environments where higher detection accuracy is critical. However, MobileNetV1 remains a viable option, particularly in scenarios where model simplicity and slightly lower computational overhead are prioritized.

5. Conclusions

This study highlights the critical importance of leveraging advanced techniques for gas leak detection in residential and industrial settings. By integrating IoT and TinyML technologies, we have developed a system that enhances the accuracy and efficiency of gas leak detection while operating on resource-constrained devices.

Deploying our gas leak detection system on the Arduino Nano 33 BLE Sense, complemented by the MLX90640 thermal camera, demonstrates the practical applicability of edge-based intelligence. The combination of the ARM Cortex-M4F processor and MLX90640 thermal imaging facilitates detailed and real-time monitoring, crucial for effective edge-based data analytics and immediate decision making.

Our experimental results show that the EfficientNet-B0 model achieved a testing accuracy of 91.73% with high recall rates, ensuring minimal false alarms and reliable detection capabilities. The resource-efficient design of both the MobileNetV1 and EfficientNet-B0 models, optimized through post-training quantization, ensures seamless integration into the Arduino Nano 33 BLE Sense, affirming our commitment to deploying cutting-edge technology in practical applications.

The ability to process data locally using TinyML enhances system responsiveness and reduces the likelihood of false alarms. This localized decision-making capability is crucial for maintaining system efficiency and reliability, especially in environments where rapid detection and response are essential.

The practical use of this gas leak detection system in actual settings will become the main emphasis going forward. This will entail thorough testing, optimization, and fine-tuning to guarantee dependability and efficacy in a range of scenarios. In future work, we will focus on building the testbed and deploying the trained models on the Arduino Nano 33 BLE Sense to validate the system's real-world performance. This deployment will allow us to assess the models' accuracy, energy efficiency, and responsiveness in real-time gas leak detection.

In summary, the combination of IoT and TinyML technologies offers a reliable, effective, and expandable method for detecting gas leaks. The system's potential for broad use is demonstrated by the successful deployment of the MLX90640 thermal camera in conjunction with the Arduino Nano 33 BLE Sense. This study highlights the revolutionary effect of edge-based intelligence on environmental monitoring systems, opening the door for future advancements in safety monitoring.

This study addressed the main research questions and demonstrated the effectiveness of TinyML for gas leak detection in various settings:

- **RQ1:** TinyML models were found to enhance real-time detection, with EfficientNet B0 achieving an accuracy of 91.73% on low-power devices.
- **RQ1:** The balance between speed, processing load, and accuracy was managed effectively through techniques like post-quantization, which reduced model size while maintaining reliable performance.

- **RQ1:** Testing confirmed the strong performance of optimized models across different conditions and hardware setups. MobileNetV1 provided an efficient, lightweight option, while EfficientNet B0 delivered higher accuracy in settings with more available resources.

This work supports the use of TinyML for practical, scalable safety monitoring. Future studies can expand upon this research by testing under more varied conditions and exploring detection of different types of gases.

Author Contributions: Conceptualization, N.B. and H.T.; Methodology, M.E.B., N.B. and H.T.; Validation, M.E.B.; Formal analysis, M.E.B.; Investigation, M.E.B.; Data curation, M.E.B.; Writing—original draft, M.E.B.; Writing—review and editing, N.B., H.T. and M.B; Supervision, N.B., H.T. and M.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The dataset described in this paper is available at https://data.mendeley.com/datasets/zkwgkjkjn9/2 (accessed on 15 May 2024).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Aman, F.; Puvanen Thiran, T.; Yusof, K.H.; Md Sapari, N. IoT Gas Leakage Detection Alert and Gas Concentration Reduction System. In Proceedings of the 2022 IEEE 12th Symposium on Computer Applications & Industrial Electronics (ISCAIE), Penang, Malaysia, 21–22 May 2022.
2. Lorthong, S.; Janjarassuk, U.; Jayranaiwachira, N. LPG Leakage Risk Predictions from an IoT-Based Detection System Using Machine Learning. In Proceedings of the 2023 9th International Conference on Engineering, Applied Sciences, and Technology (ICEAST), Vientiane, Laos, 1–4 June 2023.
3. Khan, M.M. Sensor-Based Gas Leakage Detector System. *Eng. Proc.* **2020**, *2*, 28. [CrossRef]
4. Abadade, Y.; Temouden, A.; Bamoumen, H.; Benamar, N.; Chtouki, Y.; Hafid, A.S. A Comprehensive Survey on TinyML. *IEEE Access* **2023**, *11*, 3294111. [CrossRef]
5. Lin, J.; Zhu, L.; Chen, W.-M.; Wang, W.-C.; Han, S. Tiny Machine Learning: Progress and Futures. *IEEE Circuits Syst. Mag.* **2024**, *23*, 8–34. [CrossRef]
6. Pimpalkar, A.S.; Niture, D.V. Towards Contactless Elevators with TinyML using CNN-based Person Detection and Keyword Spotting. *IEEE Access* **2024**, 1–9. [CrossRef]
7. Atanane, O.; Mourhir, A.; Benamar, N.; Zennaro, M. Smart Buildings: Water Leakage Detection Using TinyML. *Sensors* **2023**, *23*, 9210. . [CrossRef] [PubMed]
8. Abadade, Y.; Benamar, N.; Bagaa, M.; Chaoui, H. Empowering Healthcare: TinyML for Precise Lung Disease Classification. *Future Internet* **2024**, *16*, 391. [CrossRef]
9. Gkogkidis, A.; Tsoukas, V.; Papafotikas, S.; Boumpa, E.; Kakarountas, A. A TinyML-based system for gas leakage detection. In Proceedings of the 11th International Conference on Modern Circuits and Systems Technologies (MOCAST), Bremen, Germany, 8–10 June 2022.
10. Kopbayeva, A.; Khan, F.; Yang, M.; Halim, S.Z. Gas leakage detection using spatial and temporal neural network model. *Process Saf. Environ. Prot.* **2022**, *159*, 103–112. [CrossRef]
11. Miao, X.; Zhao, H.; Xiang, Z. Leakage detection in natural gas pipeline based on unsupervised learning and stress perception. *Process Saf. Environ. Prot.* **2022**, *164*, 60–70. [CrossRef]
12. Doshmanziari, R.; Khaloozadeh, H.; Nikoofard, A. Gas pipeline leakage detection based on sensor fusion under model-based fault detection framework. *J. Pet. Sci. Eng.* **2020**, *184*, 106581. [CrossRef]
13. Du, Y.; Jiang, J.; Yu, Z.; Liu, Z.; Pan, Y.; Xiong, K. A knowledge guided deep learning framework for underground natural gas micro-leaks detection from hyperspectral imagery. *Energy* **2024**, *294*, 130847. [CrossRef]
14. Narkhede, C. Multimodal AI-based sensor fusion for gas detection. *Sensors* **2021**, *21*, 342–354.
15. Wang, J.; Liu, F.; Zhang, H.; Yang, X.; Yan, J. Invisible Gas Detection: An RGB-Thermal Cross Attention Network and A New Benchmark. *arXiv* **2024**, arXiv:2403.17712. [CrossRef]
16. Attallah, O.; Elhelw, A.M. Gas Leakage Recognition Using Manifold Convolutional Neural Networks and Infrared Thermal Images. In Proceedings of the 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE), Las Vegas, NV, USA, 24–27 July 2023.
17. Sharma, A.; Khullar, V.; Kansal, I.; Chhabra, G.; Arora, P.; Popli, R.; Kumar, R. Gas Detection and Classification Using Multimodal Data Based on Federated Learning. *Sensors* **2024**, *24*, 5904. [CrossRef] [PubMed]
18. Tsoukas, V.; Gkogkidis, A.; Boumpa, E.; Papafotikas, S.; Kakarountas, A. A Gas Leakage Detection Device Based on the Technology of TinyML. *Technologies* **2023**, *11*, 45. [CrossRef]

19. Shafin, S.S.; Karmakar, G.; Mareels, I. Obfuscated Memory Malware Detection in Resource-Constrained IoT Devices for Smart City Applications. *Sensors* **2023**, *23*, 5348. [CrossRef] [PubMed]
20. Narkhede, P.; Walambe, R.; Chandel, P.; Mandaokar, S.; Kotecha, K. MultimodalGasData: Multimodal Dataset for Gas Detection and Classification. *Data* **2022**, *7*, 112. [CrossRef]
21. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv* **2017**, arXiv:1704.04861.
22. Tan, M.; Le, Q.V. EfficientNetV2: Smaller Models and Faster Training. *Proc. Mach. Learn. Res.* **2021**, *139*, 10096–10106.