INFOTECH
Journal of Applied Sciences and Knowledge Society

Volume 2, No. 1, January/December 2021

Smart Buildings: Water Leakage Detection Using TinyML

Amelia Jackson[1]*

[1] Lakeside University – Canada, ameliajackson2015@hotmail.com, https://orcid.org/0009-0000-3207-7238

ABSTRACT

Rising global water demand and urban water shortages underscore the need for effective water management. This study explores leveraging TinyML in smart buildings to enhance water management by integrating sensors and embedded Machine Learning models. TinyML enables real-time data collection, analysis, and precise decision-making for optimal water utilization, reducing reliance on centralized entities. The proposed solution adapts to real-world scenarios, detecting leaks with minimal human intervention. Following a machine learning lifecycle, the study utilizes an acoustic dataset, applying transfer learning to five Convolutional Neural Network (CNN) variants. The EfficientNet model achieves notable results, with a maximum testing accuracy, recall, precision, and F1 score of 97.45%, 98.57%, 96.70%, and 97.63%, respectively. For deployment on the Arduino Nano 33 BLE edge device, the EfficientNet model undergoes quantization, ensuring low inference time (1932 ms), peak RAM usage (255.3 KB), and flash usage (48.7 KB).

Keywords: *EfficientNet; TinyML; accelerometer; acoustic data; scalogram*

Copyright: © 2021 by the authors. This article is an open access article distributed under the terms and conditions of the Creative Commons

Received:
17 May, 2021

Accepted for publication:
30 June, 2021

Edificios Inteligentes: Detección de Fugas de Agua Utilizando TinyML

RESUMEN

El creciente aumento en la demanda global de agua y las escaseces urbanas subrayan la necesidad de una gestión efectiva del agua. Este estudio explora la aplicación de TinyML en edificios inteligentes para mejorar la gestión del agua mediante la integración de sensores y modelos de Machine Learning integrados. TinyML permite la recolección de datos en tiempo real, análisis y toma de decisiones precisas para una utilización óptima del agua, reduciendo la dependencia de entidades centralizadas. La solución propuesta se adapta a escenarios del mundo real, detectando fugas con una intervención humana mínima. Siguiendo un ciclo de vida de aprendizaje automático, el estudio utiliza un conjunto de datos acústicos, aplicando transfer learning a cinco variantes de Convolutional Neural Network (CNN). El modelo EfficientNet logra resultados destacados, con una precisión máxima en las pruebas, recall, precisión y puntuación F1 de 97.45%, 98.57%, 96.70% y 97.63%, respectivamente. Para la implementación en el dispositivo de borde Arduino Nano 33 BLE, el modelo EfficientNet se somete a cuantificación, asegurando un bajo tiempo de inferencia (1932 ms), uso máximo de RAM (255.3 KB) y uso de flash (48.7 KB).

Palabras clave: *EfficientNet; TinyML; acelerómetro; datos acústicos; escalograma*

INTRODUCTION

The presence of water is the critical factor enabling Earth to support life. Recent research indicates a projected 55% surge in global water usage, with nearly a quarter of major cities already facing varying degrees of water strain [1]. Scarce safe drinking water is a consequence, affecting approximately 2.2 billion people globally [1]. Various initiatives, including water restrictions, non-potable water reuse, and awareness campaigns, have been proposed to address the global water crisis. However, a significant challenge for water management institutions worldwide is the substantial waste of potable water due to leaks in distribution networks [2]. Water wastage results from both human behavior-related factors and the condition of distribution network pipes. Pipe defects can lead to substantial losses of quality water and energy wasted in purification processes [3].

Morocco, exemplifying the struggles of Global South countries with water scarcity, experiences substantial water wastage primarily due to leaks. According to a 2019 World Bank report, Morocco loses approximately 40% of its water supply due to leaks, inefficient use, and aging infrastructure [4]. This significant water wastage emphasizes the urgency of adopting more efficient water management practices.

Water leakages in building pipelines are a widespread and costly problem with diverse causes, including age, wear and tear, corrosion, external damage (temperature and humidity), poor installation, or pipeline quality. Leaks can occur throughout the pipeline system, leading to considerable water waste, higher water bills, structural damage, mold growth, and related issues. Early detection and repair of leaks are crucial to minimizing damage and preventing expensive repairs.

One approach to alleviate water stress involves enhancing water efficiency by minimizing non-revenue water, particularly through the detection of leaks in building pipelines. Various techniques, from visual inspection to advanced technologies like cameras, acoustic, pressure, acceleration, and flow sensors, have been developed for this purpose. Identifying the most effective methods for leak detection is crucial in smart building environments, where technical intelligence plays a key role. The goal of intelligent buildings is autonomous operation, capable of learning, forecasting, and adapting without user intervention. Sensors and monitors enable automatic adjustments in parameters such as room temperatures, lighting, shading, and energy and water consumption.

Information and communication technology, specifically sensors and Machine Learning (ML) techniques, contribute to enhancing the quality of life in smart buildings, especially in urban settings. Smart cities require buildings equipped with smart water management technology for efficient and sustainable water resource monitoring, usage, and reuse. Smart water management systems typically incorporate technologies such as live data-collecting sensors, ML models for data analysis, and intelligent controllers for local decision-making.

While traditional IoT systems and various ML techniques have shown improvements, they face challenges such as high communication loads, processing delays, and data privacy issues when relying on cloud-based processing. TinyML, a subset of edge computing, has emerged as a solution by enhancing intelligence on resource-constrained edge devices like sensors and IoT objects, effectively addressing these challenges.

This enables devices to perform AI tasks locally, often with very low power consumption, without relying on cloud-based processing. As a result, the central entity's work is drastically decreased or in certain circumstances abolished completely, with just periodic updates of metadata information required for supervision purposes. This work delves deeper into smart water leak detection using TinyML, which would permit the refinement of water utilities by detecting leaks and tracking water distribution within a building. Sensors 2023, 23, 9210 3 of 17 Detecting water leaks with ML requires the usage of various sorts of data in order to effectively detect and locate probable leaks. Several data sources and approaches may be used in this process to detect and prevent leaks effectively [10]. Among these, sensors' data play a pivotal role. Water distribution systems are equipped with numerous sensors that capture real-time data such as water flow rates, pressure levels, and vibrations. ML systems can discover abnormalities and trends that indicate anomalies by monitoring these metrics. A quick decrease in pressure or an unusual rise in flow rate, for instance, might indicate a possible leak. Acoustic data is another useful data source [10]. Water leaks frequently make distinct noises that may be detected by sophisticated acoustic sensors or even by analyzing audio recordings. ML algorithms that have been taught to recognize certain leak-related sound patterns can be used to detect leaks automatically. By leveraging these diverse data sources and employing ML algorithms, water utilities can improve their ability to detect leaks promptly. The aim of our work is to develop TinyML models that are capable of detecting water leakage in pipes using acoustic data. In this study, we focus on deep learning models, especially Convolutional Neural Networks (CNNs). CNNs are employed to detect water leakages by utilizing the sensory input of a water distribution network.

CNNs examine similarities in sensor inputs within specific regions, identifying leaks by detecting variations in sensor readings due to changes in flow, pressure, or vibration. These models effectively capture these distinctions, enabling the precise identification of leaks within the water distribution network. The subsequent sections of this paper are organized as follows: Section 2 explores related work, presenting prior research addressing similar issues. Section 3 outlines the materials and methods employed in our experiments. Section 4 details the empirical results and discusses the findings obtained from the experiments. Finally, Section 5 summarizes conclusions and suggests directions for future work.

Related Works

Numerous research endeavors have addressed the application of machine learning or deep learning, specifically CNN architectures, to identify patterns and detect anomalies within water and wastewater pipeline systems. For instance, Fang et al. [11] introduced a CNN-based method designed to identify multiple leakage spots. This model extracts crucial features from historical leakage data and applies them to real-time data to ascertain the presence of a leak. To validate their approach, they constructed an experimental platform simulating a water distribution system within the Anhui Province Key Laboratory of Intelligent Building and Building Energy Saving. This platform covers an area of 200 m2, incorporating pipe sections measuring 400 m in length with pipe diameters ranging from 30 cm to 50 cm. Strategically placing 21 water pressure sensors throughout the platform, they collected a comprehensive dataset encompassing non-leakage data and four types of pipe network pressure data under different conditions, including single-point leakages, two-point leakages, and three-point leakages. The experimental results demonstrated significant detection accuracies, reaching 99.63%, 98.58%, and 95.25% accuracy for one, two, and three leakage spots, respectively, utilizing 21 sensors. A reduction in the number of sensors to eight resulted in slightly diminished accuracies for one, two, and three leakage points, reaching 96.43%, 94.88%, and 91.56%, respectively.

In Kang et al. [12], the researchers employed non-invasive measurements of leakage signals using piezoelectric accelerometers (PCB-393B31).

These accelerometers possess the capability to objectively measure vibrations and convert them into levels of acceleration. Additionally, the authors introduced a local search technique based on graphs to pinpoint leaks. However, their approach was not entirely optimized for categorizing 1-D signals, necessitating feature extraction from the recorded signal data before applying the classification layers. Moreover, the detection range relied on the clarity and correlation of acoustic signals, and the challenge of errors caused by signals with low correlation coefficients remained unresolved.

In a distinct investigation by Cody et al. [13], a water system experimental test bed was designed, comprising components such as a full-scale hydrant and PVC pipes with bends. These pipes, constructed from grayscale schedule 80 PVC with a 152.4 mm inner diameter, are extensively used in water distribution networks in Canada and the United States. The authors employed a CNN architecture, with its output fed into a variational autoencoder (VAE) aiming to recover the original spectrogram image. The mean squared error (MSE) between the original image and its reconstructed counterpart was used to compute the loss function. This proposed method achieved an accuracy of 97.2% for detecting a 0.25 L/s leak.

Shukla et al. [14] conducted a study in which a CNN model was constructed using modified layers of the pre-trained AlexNet network. The objective of the model was to classify images based on various scenarios, utilizing a dataset of 9000 scalogram images. This involved considering 25 scenarios, with 12 accelerometers per scenario, 10 samples per scenario, and three images per sample. The model demonstrated proficiency in categorizing images based on their corresponding leakage scenarios, accurately identifying both healthy configurations and various leaky situations with a 95% accuracy rate. Additionally, the average recall, precision, and F1 score for both validation and testing data were

94% and 95%, respectively. The authors' findings suggest that the CNN model effectively detects true positive labels with a high degree of accuracy.

Coelho et al. [16] introduced an IoT system with the capability to monitor water distribution systems and precisely detect and locate water leaks.

The proposed solution incorporates flow sensors and budget-friendly microcontrollers (ESP32) for the acquisition and real-time processing of data. Five diverse classification algorithms, including random forests, decision trees, neural networks, support vector machines, and XGBoost, were evaluated through 12 tests each to determine the most accurate algorithm for system implementation. The random forest algorithm consistently demonstrated the highest accuracy across various scenarios, reaching almost 85%, making it the preferred choice.

Loukatos et al. [17] introduced a system addressing the challenges of traditional IoT systems. By employing embedded machine learning on a Raspberry Pi Pico microcontroller board, the authors trained a neural network to identify three distinct water utilization profiles: Normal Use (NU), Water Leak (WL), and Water Waste (WW). The neural network structure consisted of an input layer with 200 features (window size), two hidden layers—with the first containing 20 neurons and the second 10 neurons—and an output layer with three classes. Upon evaluating the testing data, the system achieved 77.8% accuracy for the NU category, correctly identifying instances of Normal Use. Additionally, it achieved a 100% success rate for both WW and WL categories, accurately identifying scenarios of Water Waste and Water Leak. These results led to an anticipated accuracy of 98.5% for the final model when tested using the quantized (int8) version of the dataset.

The investigation into domestic water leak detection involved the analysis of flow data [18]. This research focused on training both a random forest and a CNN-based model in the cloud to address the classification problem of distinguishing leak events from non-leak events and determining the magnitude of leaks categorized as small (≤1 L/h), medium-sized (1 to 10 L/h), or large (≥10 L/h). The CNN model outperformed, exhibiting an accuracy, precision, and recall ranging from 92% to 96%. Additionally, the area under the Precision-Recall (PR) and Receiver Operating Characteristic (ROC)

INFOTECH
Journal of Applied Sciences and Knowledge Society
Volume 2, No. 1, January/December 2021

curves consistently achieved high values, ranging between 97% and 99%. Table 1 provides a summary of the key features from the literature review.

Table 1. Summary of Relevant Studies

Research Paper	**Data/Sensors**	**Application Domain**	**Model Deployment**	**Classification Problem**	**Model(s)**	**Results**
[11]	Pressure data, JOHNSON CONTROLS (P499ABS-401) sensor	Water distribution systems	No specific target	13 categories depending on leakage point and position	CNN model	Accuracy 97.33%
[12]	Accelerometer data, piezoelectric accelerometers (PCB-393B31)	Water distribution systems	Backend server	Leak, non-leak conditions	Ensemble 1D-CNN-SVM model, graph-based localization algorithm	Accuracy 99.3%, Sensitivity 98.2%, Specificity 99.8%, AUC 99.9%
[13]	Acoustic signals, hydrophone sensor	Water distribution networks	No specific target	Leak, non-leak conditions	CNN with VAE (Variational Autoencoders)	Accuracy 97.2%, Precision 92%, Recall 96%
[14]	Accelerometer data (available upon request from authors)	Infrastructure monitoring	Offline sensitivity analysis–not applicable	25 scenarios depending on leakage flow	CNN adapted from AlexNet	Average Accuracy 94–98% depending on locations
[16]	Water flow data	Agriculture IoT, cloud	No leaks, micro, minor, major leaks	Random Forest	Accuracy 85%	
[17]	Water flow data	Smart Agriculture Edge	Normal Use, Water Leak, Water Waste	Multi-layer Perceptron Neural Network	Accuracy 98.5%	
[18]	Water flow data	Domestic water	Cloud	No leaks, small, medium, and large leaks	CNN	Accuracy 92–96%, Precision and Recall varied between 92% and 96%

The existing body of research on water leakage detection systems highlights a significant gap in the incorporation of TinyML methodologies. While previous endeavors have focused on addressing the challenge of water leakage, a considerable portion of these solutions has not fully leveraged the potential advantages offered by TinyML approaches. Furthermore, those few initiatives that do incorporate

TinyML techniques often demonstrate limitations, particularly in terms of the scope and the considered CNN models.

Our contribution lies in conducting experiments with cutting-edge CNN models, including ResNets [19], MobileNet [20], and EfficientNet [21] architectures, to achieve highly accurate and efficient water leakage detection. What sets our work apart is the special attention we give to integrating these models into small-scale devices using TinyML. This dual approach not only widens the range of considered ML models but also facilitates the seamless integration of state-of-the-art technology into an effective real-time operational framework for water leakage detection.

MATERIALS AND METHODS

In this section, we elaborate on the methodologies employed for acquiring, preprocessing, and analyzing the data central to our study. Figure 1 provides a comprehensive visual representation of the approach we adopted for this study. The process involves several crucial stages, similar to traditional machine learning projects, with additional considerations for the constrained real-time processing requirements of the embedded device. We preprocess the acoustic leakage data and generate scalogram images. The various CNN models undergo training and optimization, with hyperparameters tuned on the validation set. Evaluation is performed using a separate testing set, and the chosen model with the best testing metrics is optimized for on-device resource efficiency. In our research, we compressed the model using quantization and evaluated its performance, considering the real-time hardware constraints of the device. Finally, we generate the C++ packages for deployment on the device. The development lifecycle was executed on Edge Impulse (https://edgeimpulse.com/ (accessed on 15 March 2023)), a platform facilitating model training, hyperparameter tuning, and model optimization for deployment on any edge device.

Figure 1 illustrates the systematic approach employed in this study

Data Acquisition and Preprocessing

For this investigation, we utilized an existing dataset compiled by Shukla et al. [14], derived from a small-scale experimental setup established at Clemson University's campus in the United States. The dataset captures pipeline vibrations, utilizing accelerometers strategically placed at various positions along the pipe's length. The experimental design incorporated elements simulating real-world complexity, including pipes with different diameters, rounded bends, T-joints, pipes at various heights, and varied burial conditions.

Acceleration data were collected using Bruel and Kjaer (BK) 4507-B-006 accelerometers situated at 12 locations dispersed throughout the pipeline system. The setup featured two PVC pipes, one with a diameter of 3 inches (76 mm) and the other with a diameter of 4 inches (102 mm). The latter was partially covered, while the remainder was left exposed. Wooden boxes were erected on the subterranean to facilitate accelerometer placement. Accelerometers were positioned at an average distance of 30 inches (762 mm) for the unburied pipeline and 24 inches (610 mm) for the buried pipeline.

The experimental setup included two leak simulators, one on a 3-inch (76 mm) PVC pipe and the other on a 4-inch (102 mm) pipe. These simulators, along with the test configuration details, are further described in [22,23]. Accelerometers 1-6 detected leaks in the unburied section, while accelerometers 7-12 detected leaks in the buried section.

Table 2 outlines the recorded flow information for all 25 scenarios obtained from the flowmeter. Each scenario's acceleration signal data was recorded under stable flow conditions, ensuring repeatability. Ten samples of 15-second data were taken for each scenario to ensure accuracy and eliminate biases. The dataset was diversified by varying the size of leaks and water flow in each scenario [14].

Our research focused on detecting pipeline leaks, categorizing the data into two classes: 'Leak' and 'NonLeak.' To employ CNN models in this context, acoustic data needed to be transformed into scalogram images. Scalograms represent the absolute value of the continuous wavelet transform (CWT), illustrating how a signal varies with frequency and time. Unlike spectrograms obtained from the fast Fourier transform (FFT), which splits the signal into smaller parts, scalograms break down the signal into wavelets. Wavelets, characterized by parameters, are effective in detecting abrupt shifts in the signal by localizing it in both frequency and time. Scalograms are particularly suitable for studying low-frequency acceleration signals [24].

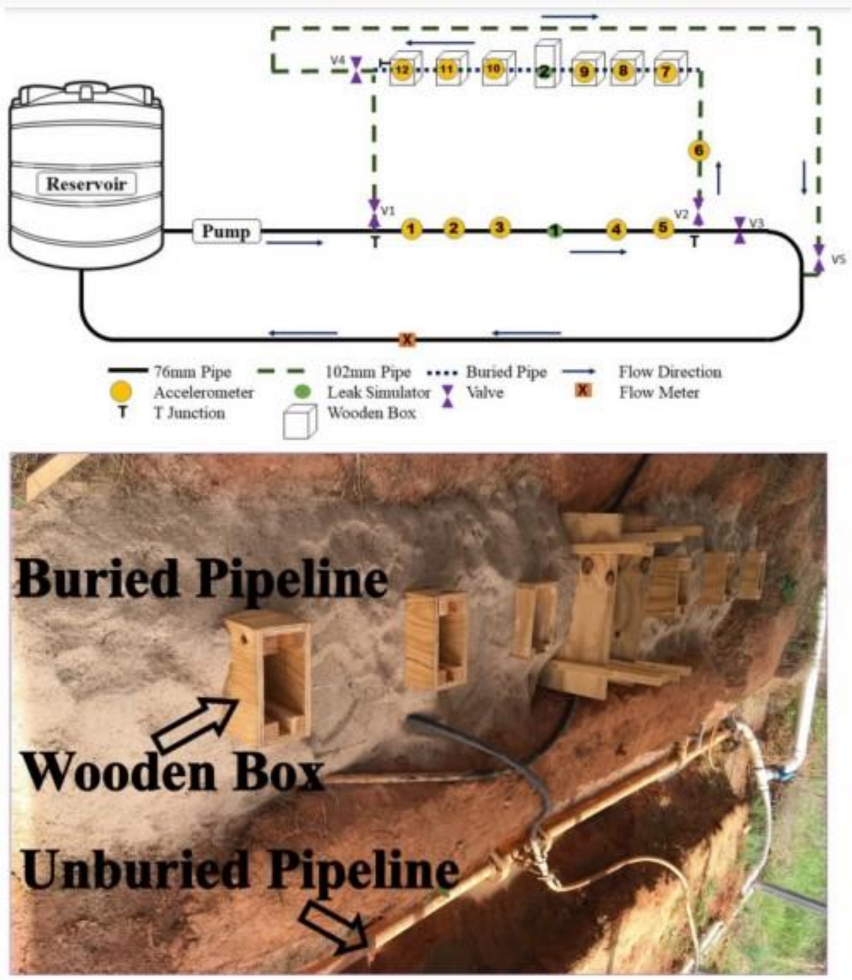

Figure 2 depicts the schematic arrangement of the experimental configuration for the pipeline system
We generated a total of 3,120 scalograms, comprising 1,440 for 'NonLeak' and 1,680 for 'Leak.'

Achieving a balanced dataset was crucial, and we employed data augmentation techniques, specifically random cropping and noise injection. Random cropping involved randomly cropping sections of scalogram images during training to enhance the model's robustness to variations. Noise injection simulated real-world conditions where data is often corrupted, contributing to increased model resilience. This transformative process not only addressed class imbalance but also improved the generalization of our machine learning models, reducing the risk of overfitting.

Figures 3–14 showcase scalogram images representing different scenarios for Leak, NonLeak, buried, and unburied sections of the pipeline setup. Upon comparison, it becomes apparent that scalogram images from the buried section display less noise and clearer features (indicated by prominent green spots) in contrast to those from the unburied section. However, distinguishing between scalogram images corresponding to non-leaky (Figures 3–8) and leaky (Figures 9–14) scenarios presents a challenge.

Table 2 outlines the scenarios related to water leakage data

Scenarios	**Leak 1 (GPM)**	**Leak 2 (GPM)**	**Flow Rate (GPM)**
SC1	0	0	163
SC2	1	0	160.6
SC3	1	1	159.2
SC4	5.28	1	157.7
SC5	14.73	1	156.8
SC6	22.63	1	152.8
SC7	0	1	163.5
SC8	0	5.19	160
SC9	1.04	5.19	159
SC10	1.04	14.08	156
SC11	1.04	23.46	154
SC12	0	23.46	153
SC13	4.73	23.46	152
SC14	13.77	23.46	148
SC15	22.63	23.46	143
SC16	22.63	4.53	150.9
SC17	22.63	13.77	146
SC18	0	13.77	156
SC19	14.4	13.77	149
SC20	14.4	5.66	152
SC21	4.73	5.66	156
SC22	4.73	13.77	152
SC23	4.73	0	158.8
SC24	13.77	0	157
SC25	21.12	0	155

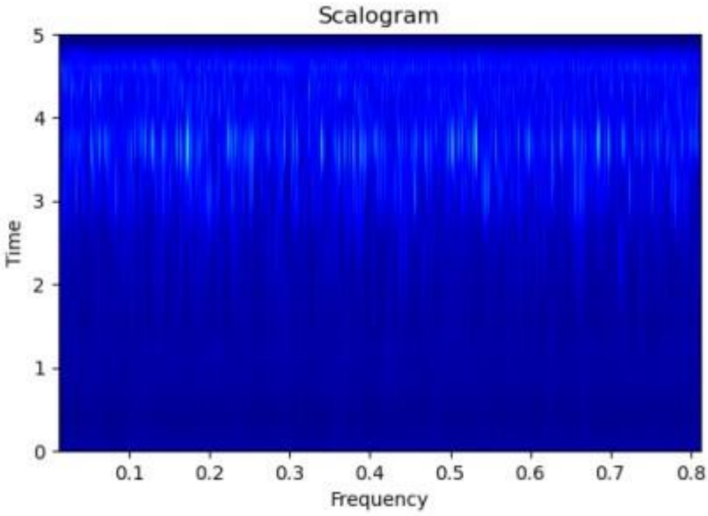

Figure 3. Scalogram 1 of Unburied NonLeak Scenario

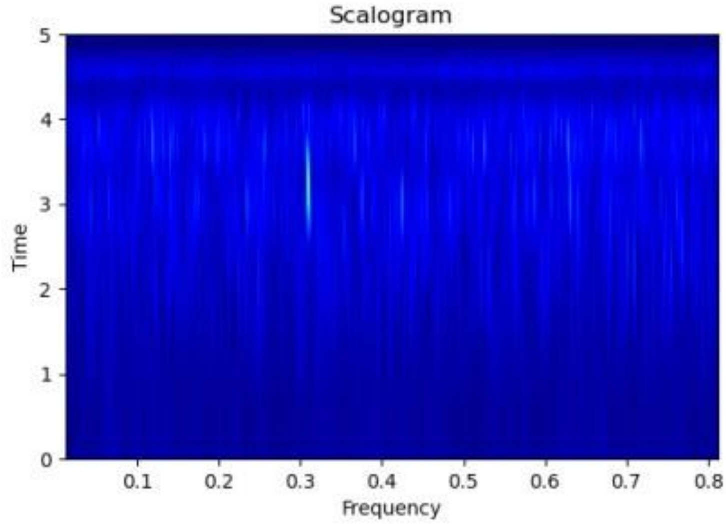

Figure 4. Scalogram 7 of Unburied NonLeak Scenario

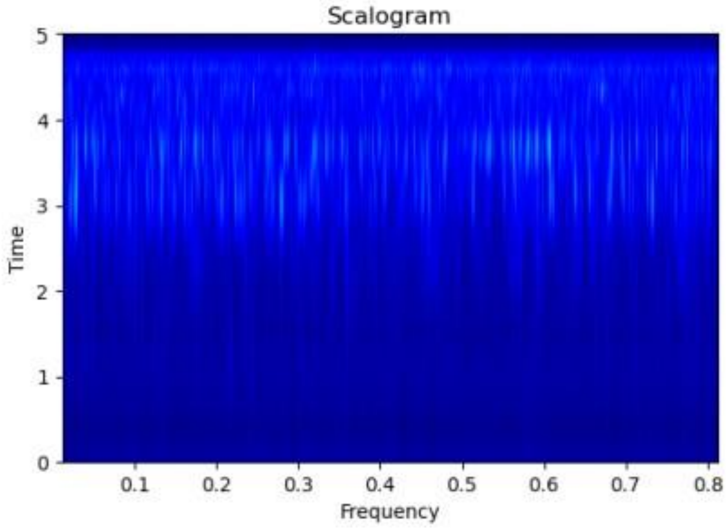

Figure 5. Scalogram 18 of Unburied NonLeak Scenario

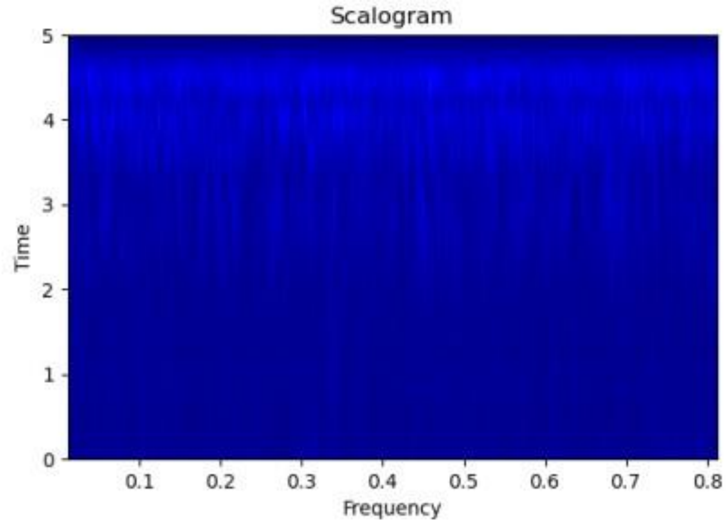

Figure 6. Scalogram 1 of Buried NonLeak Scenario

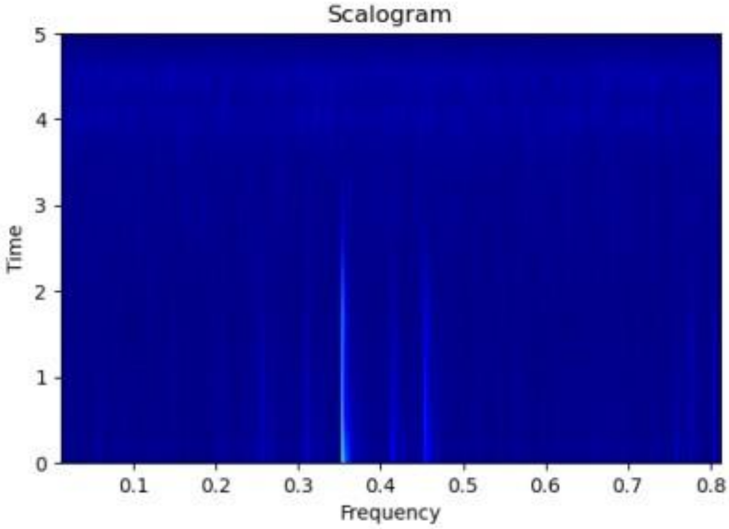

Figure 7. Scalogram 23 of Buried NonLeak Scenario

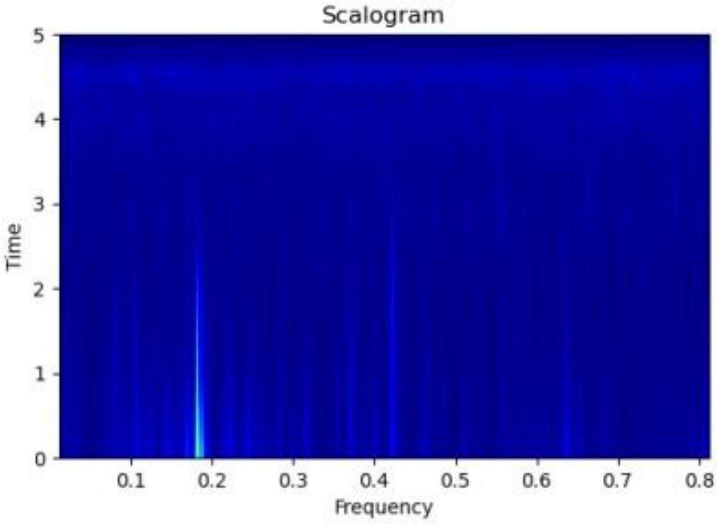

Figure 8. Scalogram 25 of Buried NonLeak Scenario

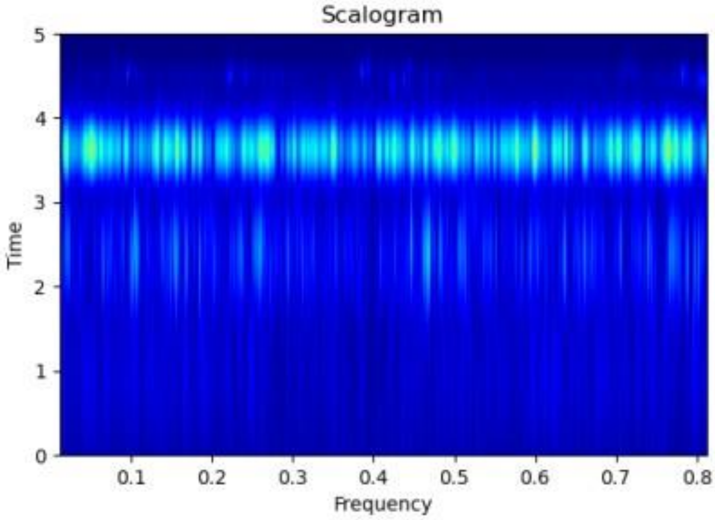

Figure 9. Scalogram 5 of Unburied Leak Scenario

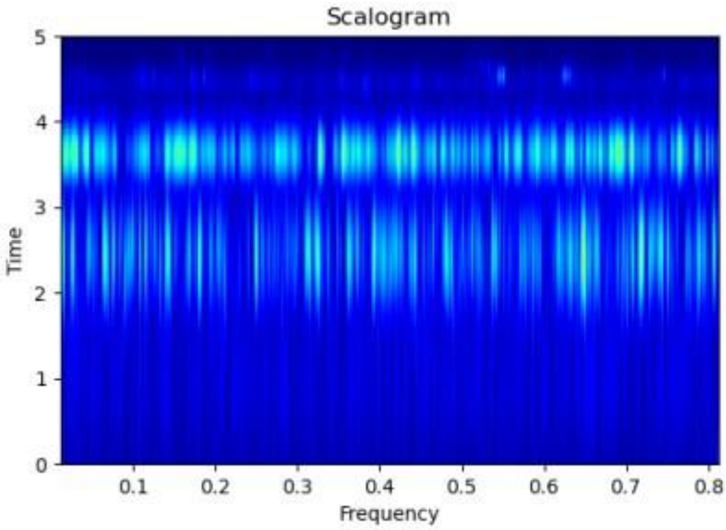

Figure 10. Scalogram 17 of Unburied Leak Scenario

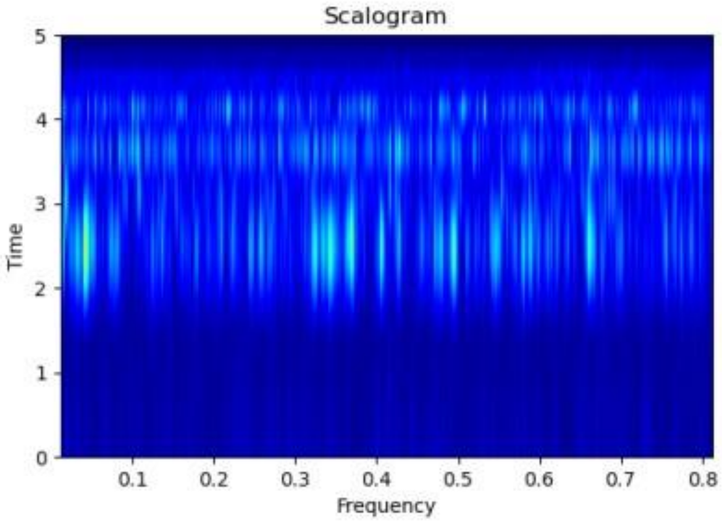

Figure 11. Scalogram 23 of Unburied Leak Scenario

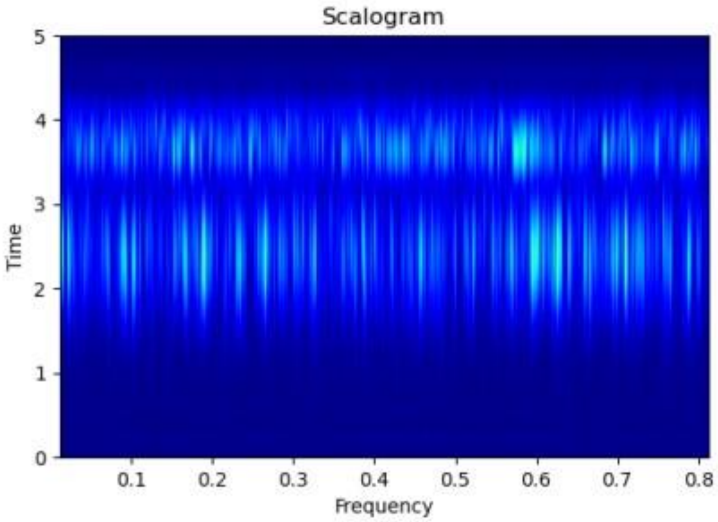

Figure 12. Scalogram 6 of Buried Leak Scenario

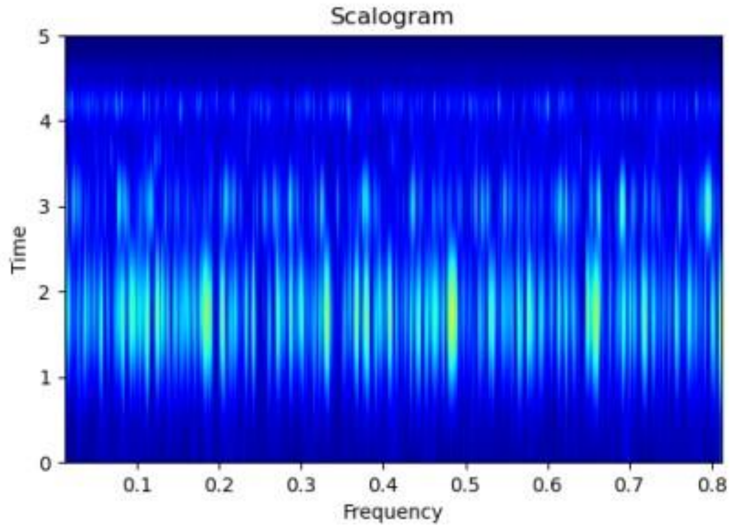

Figure 13. Scalogram 12 of Buried Leak Scenario

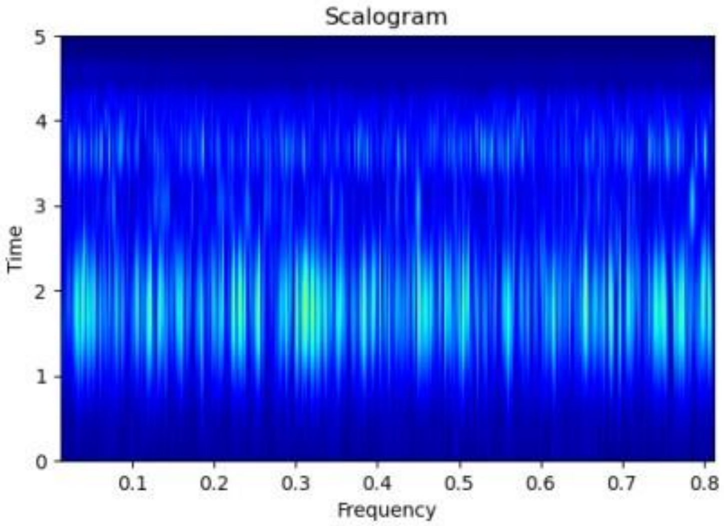

Figure 14. Scalogram 19 of Buried Leak Scenario

CNN Models for Detecting Water Leaks

In our study, we implemented transfer learning to train several CNN models, including AlexNet, ResNet, EfficientNet, MobileNet V1, and MobileNet V2. Each of these models exhibits unique architectural features and computational efficiencies, rendering them well-suited for our investigation.

AlexNet, introduced by Krizhevsky et al. in 2012 [15], stands as a deep CNN model that played a crucial role in transforming image classification tasks. It comprises multiple convolutional layers followed by fully connected layers, utilizing techniques like ReLU activation, dropout regularization, and overlapping pooling. Despite its relatively small size (61 million parameters), AlexNet delivered outstanding performance and surpassed other existing models upon its introduction.

ResNet, or Residual Network, presented by He et al. in 2015 [19], addressed the challenge of training extremely deep neural networks. Introducing the concept of residual connections, it allowed the network to learn residual mappings, effectively dealing with the vanishing gradient problem. By incorporating skip connections, ResNet facilitated the training of deeper networks with improved accuracy and ease of optimization.

EfficientNet [21] aims to optimize both model accuracy and computational efficiency. Its architecture was derived using neural architecture search to strike a balance between model size and performance, making it highly efficient for resource-constrained environments. EfficientNet employs a compound scaling method to systematically balance the depth, width, and resolution of the network. These features enable EfficientNet to transfer well and achieve state-of-the-art accuracy on various benchmark datasets, leading to its increasing popularity for impressive results within limited computational budgets.

MobileNets, introduced by Howard et al. in 2017 [20], focus on efficient mobile applications and low-power devices. They utilize depthwise separable convolutions to reduce computational costs while maintaining reasonable accuracy. MobileNet architectures are well-suited for resource-constrained environments, allowing real-time image classification on devices with limited processing capabilities. In our research, we utilized two versions from this family of models: MobileNet V1 [20] and MobileNet V2 [25].

All five models underwent pre-training on the ImageNet dataset, a large-scale collection of approximately 1.2 million labeled images spanning 1000 different categories. This pre-training process ensures a strong foundation in visual recognition and feature extraction, providing the models with a comprehensive understanding of various objects, scenes, and concepts. This extensive pre-training equips the models to be valuable tools for a range of computer vision tasks.

The selection of these models reflects a balance between model complexity, resource efficiency, and historical significance, specifically tailored to the requirements and constraints of the water leak detection project. Furthermore, these models have proven effective not only in traditional image classification but also in the context of analyzing scalogram images. Additional details about the architectures of the utilized models are available in Table 3.

Table 3. Architectural Overview of Models

Model	**Architecture**
AlexNet [15]	Convolutional layers featuring kernel sizes (11 × 11, 5 × 5, 3 × 3), ReLU activation, and overlapping max pooling. Fully connected layers follow with dropout and softmax activation for classification.
ResNet [19]	Residual blocks incorporating shortcut connections. Each block includes multiple convolutional layers with batch normalization and ReLU activation. Global average pooling is applied, followed by fully connected layers with softmax activation for classification.
EfficientNet [21]	Utilizes a compound scaling technique, adjusting the depth, width, and resolution of the network. Consists of convolutional layers with efficient bottleneck structures, swish activation, and batch normalization. Global average pooling is applied, followed by fully connected layers with softmax activation for classification.
MobileNet V1 [20]	Employs depthwise separable convolutional layers, dividing the standard convolution into separate depthwise and pointwise convolutions. Features ReLU activation, batch normalization, depthwise, and pointwise convolutional layers. Max pooling is applied, followed by fully connected layers with softmax activation for classification.
MobileNet V2 [25]	Inverted residual blocks with linear bottlenecks. Utilizes depthwise separable convolutions, skip connections, and expansion layers for enhanced efficiency. The architecture includes ReLU6 activation, batch normalization, and global average pooling, with fully connected layers using softmax activation for classification.

We employed transfer learning methodologies to harness existing knowledge from diverse domains and apply it to our water leak detection problem. Transfer learning proves advantageous by capturing intricate patterns and representations present in the water leak detection dataset, thereby reducing the time and computational resources required for model development. Given the small size of our dataset, we adopted a transfer learning strategy wherein we froze the pre-existing parameters and exclusively trained the parameters of the output layer. This strategic freezing of the early layers during neural network training ensures that the weights and parameters in those layers remain unchanged, allowing the model to retain the knowledge acquired during its initial training phases.

The optimization algorithm used was ADAM, a technique seamlessly combining two essential optimization strategies [26]: Nesterov momentum, enhancing convergence, and adaptive learning rates, facilitating the adaptation of the learning rate for each parameter. Additionally, we employed a mini-batch gradient descent approach to provide a reliable estimation of the error and gradients. Our training strategy incorporated a specific number of epochs, approximately 50, determined by the tuner based on optimal validation results. To prevent overfitting, the dropout technique was applied to the different models undergoing transfer learning. Furthermore, we implemented data augmentation, a practice known for enhancing generalization.

For hyperparameter tuning, we tested a total of 15 combinations involving the number of epochs, learning rate, and mini-batch size across all five models. The objective was to identify optimal hyperparameters that could enhance training efficiency while improving classification accuracy. The details of the various hyperparameter combinations are outlined in Table 4.

Table 4. Hyperparameter Configurations

Cases	**Epochs**	**Learning Rate**	**Mini-Batch Size**
Configuration 1	10	5 × 10^(-3)	8
Configuration 2	25	5 × 10^(-3)	8
Configuration 3	50	5 × 10^(-3)	8
Configuration 4	10	5 × 10^(-4)	8
Configuration 5	25	5 × 10^(-4)	8
Configuration 6	50	5 × 10^(-4)	8
Configuration 7	10	5 × 10^(-4)	16
Configuration 8	25	5 × 10^(-4)	16
Configuration 9	50	5 × 10^(-4)	16
Configuration 10	10	5 × 10^(-4)	32
Configuration 11	25	5 × 10^(-4)	32
Configuration 12 *	50	5 × 10^(-4)	32
Configuration 13	10	5 × 10^(-5)	32
Configuration 14	25	5 × 10^(-5)	32
Configuration 15	50	5 × 10^(-5)	32

*Case demonstrating the highest levels of accuracy

We assessed the performance of the five models and conducted a comparative analysis of the results. The evaluation metrics encompassed accuracy, recall, precision, and F1 score [27,28]. These metrics collectively provide a comprehensive evaluation of model performance, addressing various aspects of prediction quality. Accuracy serves as an initial overview, offering a holistic perspective on correct classifications. However, real-world scenarios, characterized by imbalanced datasets or specific business requirements, often necessitate a more in-depth analysis. Recall is particularly important for identifying the proportion of true leak detection alerts among actual positives, emphasizing the model's capacity to detect relevant events. Precision complements recall by measuring the proportion of detected leaks among predicted positives, focusing on minimizing false positives. The F1 score strikes a balance between precision and recall, making it a reliable metric for achieving a harmonious trade-off between minimizing false positives and false negatives. Collectively, these metrics ensure a nuanced evaluation and guide the refinement of models tailored to specific project objectives and constraints.

Target Deployment

The designated device for implementation is the Arduino Nano 33 BLE Sense (Cortex-M4F 64 MHz, Arduino, Somerville, MA, USA), chosen for its compact dimensions and versatility across various applications. With a maximum RAM capacity of 256 kB and ROM of 1024 kB, the device is constrained in terms of computational resources. Notably, the Arduino Nano 33 BLE features five sensors, with particular relevance to our investigation lying in the acoustic data sensor. The MEMS microphone on board facilitates the capture of acoustic signals, supporting tasks related to sound analysis and recognition. Additionally, the integrated digital barometric sensor aids in recording atmospheric pressure variations, contributing to the understanding of fluid dynamics and airflow patterns.

Cutting-edge CNN models, while potent, often prove too extensive for deployment on compact devices like the Arduino Nano 33 BLE Sense due to their resource-intensive nature. To tackle this issue, we adopt post-training quantization, a functionality provided by the Edge Impulse platform, which optimizes models based on implementations from the Tensorflow Lite Micro library. This method strategically reduces the precision of a model's internal representations by converting 32-bit floating-point parameters into lower precision int8. Consequently, this reduction in precision significantly curtails the ROM requirements of the model, rendering it more practical for deployment on a diminutive device. Furthermore, the diminished memory usage translates into faster computation, facilitating expedited predictions and responses. Essentially, post-training quantization stands as a vital optimization step, preserving the core functionality and accuracy of the models while aligning them with the constraints of the device's limited resources.

Given that the water leak detection models are slated for deployment on the Arduino Nano 33 BLE Sense, additional metrics were taken into account, including inference time (the speed of model predictions), the model's memory usage, and storage footprint. These considerations are pivotal to ensuring the efficiency and seamless performance of the models on the Arduino Nano 33 BLE. By scrutinizing classification metrics, computational efficiency, and suitability for the compact device, informed decisions regarding model selection were made.

To deploy the ML model on the Arduino Nano 33 BLE, our application is converted into fully optimized C++ source code, ready for integration as an application on the device. The customizable library packages encompass both the preprocessing block, transforming acoustic data into a scalogram, and the machine learning block for inference. All necessary external libraries are consolidated into a single package that includes all available source code.

RESULTS AND DISCUSSION

For the training and validation of the CNN models, a split of 72%-10%-18% was employed for training, validation, and testing, respectively. A 5-fold cross-validation strategy was implemented, where distinct sets of training and validation data were randomly chosen for each fold. This approach ensured a harmonious balance, offering the model sufficient and diverse training data while facilitating a robust evaluation of its generalization performance throughout multiple folds. The outcomes of our experiments are summarized in Table 5.

EfficientNet emerged as the model exhibiting the most favorable results, as illustrated in Table 5. Several factors contributed to this notable performance. Firstly, EfficientNet's design incorporates a groundbreaking compound scaling technique, which uniformly scales the depth, width, and resolution dimensions, enabling it to surpass other models. This unique characteristic allowed EfficientNet to strike a balance between model size and accuracy, resulting in superior performance. Additionally, its extensive pre-training on extensive datasets equipped EfficientNet with the ability to learn rich and generalized representations. The effective network block designs and utilization of depth-wise separable convolutions contributed to reducing computational costs while preserving expressive capability. Consequently, the innovative architectural design and efficiency-enhancing features of EfficientNet played a pivotal role in achieving optimal performance in our research.

Table 5 outlines the evaluation of models on both validation and testing sets, with the best results highlighted for clarity

Model	Accuracy	Precision	Recall	F1 Score
AlexNet				
Testing	98.00%	99.10%	96.97%	98.02%
Validation	96.24%	96.70%	95.46%	96.08%
ResNet				
Validation	99.20%	100.0%	98.43%	99.21%
Testing	96.76%	96.70%	96.89%	96.80%
EfficientNet				
Validation	99.55%	99.10%	100.0%	99.55%
Testing	**97.45%**	96.70%	**98.57%**	**97.63%**
MobileNet V1	98.65%	99.00%	98.51%	98.75%
Validation	96.24%	96.70%	95.46%	96.08%
Testing				
MobileNet V2	99.25%	100.0%	98.52%	99.26%
Validation	96.70%	96.89%	96.80%	
Testing	96.76%			

ResNet secures the second position in our evaluation for several reasons. Its deep residual connections effectively address the vanishing gradients issue, enabling more profound training. The skip connections in ResNet aid information flow, allowing the model to capture intricate patterns. Despite higher computational complexity compared to EfficientNet and MobileNet, ResNet's architectural design and capacity for deeper networks contribute to its notable performance, placing it second in accuracy. However, the emphasis on depth comes with increased computational complexity, limiting practicality in resource-constrained scenarios.

Considering the deployment device constraints, we compressed all models using quantization. Two models, EfficientNet and MobileNet V1, fit within the device characteristics range, as outlined in Table 6, detailing inference time, RAM, and flash usage.

Among the models explored, EfficientNet emerged as the optimal choice, achieving a balance between performance and compatibility with our resource-limited device. This decision takes into account the stringent resource limitations and underscores EfficientNet's ability to meet these requirements effectively.

Table 6 displays the inference time, peak RAM usage, and flash usage for the various models

Model	Inference_Time (MS)	Peak Ram Usage (KB)	Flash Usage (KB)
AlexNet	20,843	132.4	20.7×10^3
ResNet	1958	333.8	640.7
EfficientNet	1932	255.3	48.7
MobileNet V1	3156	253.5	310.8
MobileNet V2	5200	720.8	580.2

This study underscores the critical need for advancements in leak detection techniques, especially within water distribution networks and building pipelines. The incorporation of deep learning algorithms, as exemplified by our proposed method utilizing scalogram images of vibration signals, demonstrates potential for effectively identifying and pinpointing leaks. Leveraging TinyML, a convergence of sensor technology and machine learning at the network edges, facilitates real-time data collection, analysis, and localized decision-making. This not only enhances the accuracy and efficiency of leak detection but also diminishes reliance on centralized entities.

Our experimental results highlight the dedication to achieving substantial performance and seamless integration into the designated compact deployment device. EfficientNet, with its capacity for impressive performance and resource-efficient design, emerges as the optimal choice for our deployment objectives. This reiterates our commitment to leveraging state-of-the-art technology for practical applications on hardware with inherent limitations.

Moreover, the model's notable recall ensures a reduction in false alarms. The edge solution's capability to process data locally enables the filtration of noise or false alarms before triggering alerts, thus minimizing unnecessary responses.

This research culminates in the development of an embedded water leak detection model. Looking ahead, our focus will shift towards practically implementing a water leak detection solution in smart buildings. This involves translating our discoveries to seamlessly integrate the embedded ML system into a real-world environment with hardware and environmental constraints. The deployment process will involve thorough testing, optimization, and fine-tuning to guarantee reliability and effectiveness.

BIBLIOGRAPHICAL REFERENCES

Salehi, M. Global water shortage and potable water safety; Today's concern and tomorrow's crisis. Environ. Int. 2022, 158, 106936.

Hope, W.; Bircumshaw, J. The Waste of Water in Public Supplies, and Its Prevention. In Proceedings of the Minutes of the Proceedings

of the Institution of Civil Engineers; Thomas Telford-ICE Virtual Library; Emerald Publishing Limited: Leeds, UK 1996; Volume 115,

pp. 68–75.

Fahmy, M.; Moselhi, O. Automated detection and location of leaks in water mains using infrared photography. J. Perform. Constr.

Facil. 2010, 24, 242–248.

Kadi, M.A. From water scarcity to water security in the Maghreb Region: The Moroccan Case. In Proceedings of the

Environmental Challenges in the Mediterranean 2000–2050: Proceedings of the NATO Advanced Research Workshop on

Environmental Challenges in the Mediterranean 2000–2050. Madrid, Spain, 2–5 October 2002; Springer Science & Business Media:

Berlin/Heidelberg, Germany, 2012; Volume 37, p. 175.

Shah, J.; Mishra, B. Customized IoT enabled wireless sensing and monitoring platform for smart buildings. Procedia Technol. 2016,

23, 256–263.

Ray, P.P. A review on TinyML: State-of-the-art and prospects. J. King Saud Univ.-Comput. Inf. Sci. 2022, 34, 1595–1623.

Sensors 2023, 23, 9210 17 of 17

Abadade, Y.; Temouden, A.; Bamoumen, H.; Benamar, N.; Chtouki, Y.; Hafid, A.S. A Comprehensive Survey on TinyML. IEEE

Access 2023, 11, 96892–96922.

Merenda, M.; Porcaro, C.; Iero, D. Edge machine learning for ai-enabled iot devices: A review. Sensors 2020, 20, 2533.

Garcia Lopez, P.; Montresor, A.; Epema, D.; Datta, A.; Higashino, T.; Iamnitchi, A.; Barcellos, M.; Felber, P.; Riviere, E. Edge-centric

computing: Vision and challenges. ACM SIGCOMM Comput. Commun. Rev. 2015, 45, 37–42.

Yussof, N.A.M.; Ho, H.W. Review of Water Leak Detection Methods in Smart Building Applications. Buildings 2022, 12, 1535.

Fang, Q.; Zhang, J.; Xie, C.; Yang, Y. Detection of multiple leakage points in water distribution networks based on convolutional

neural networks. Water Supply 2019, 19, 2231–2239.

Kang, J.; Park, Y.J.; Lee, J.; Wang, S.H.; Eom, D.S. Novel leakage detection by ensemble CNN-SVM and graph-based localization

in water distribution systems. IEEE Trans. Ind. Electron. 2017, 65, 4279–4289.

Cody, R.A.; Tolson, B.A.; Orchard, J. Detecting leaks in water distribution pipes using a deep autoencoder and hydroacoustic

spectrograms. J. Comput. Civ. Eng. 2020, 34, 04020001.

Shukla, H.; Piratla, K. Leakage detection in water pipelines using supervised classification of acceleration signals. Autom. Constr.

2020, 117, 103256.

Krizhevsky, A.; Sutskever, I.; Hinton, G. Imagenet classification with deep convolutional neural networks. In Proceedings of the

Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 201,

Lake Tahoe, NV, USA, 3–6 December 2012.

Alves Coelho, J.; Glória, A.; Sebastião, P. Precise water leak detection using machine learning and real-time sensor data. IoT 2020,

1, 474–493.

Loukatos, D.; Lygkoura, K.A.; Maraveas, C.; Arvanitis, K.G. Enriching IOT modules with edge AI functionality to detect water
misuse events in a decentralized manner. Sensors 2022, 22, 4874.

Zese, R.; Bellodi, E.; Luciani, C.; Alvisi, S. Neural Network Techniques for Detecting Intra-Domestic Water Leaks of Different
Magnitude. IEEE Access 2021, 9, 126135–126147.

He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference On
Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.

Howard, A.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Wey, ; T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional
neural networks for mobile vision applications. arXiv 2017, arXiv:1704.04861.

Tan, M.; Le Q. Efficientnetv2: Smaller models and faster training. In Proceedings of the International Conference on Machine
Learning, Virtual, 18–24 July 2021; pp. 10096–10106.

Yazdekhasti, S.; Piratla, K.R.; Atamturktur, S.; Khan, A.A. Novel vibration-based technique for detecting water pipeline leakage.
Struct. Infrastruct. Eng. 2017, 13, 731–742.

Yazdekhasti, S.; Piratla, K.R.; Atamturktur, S.; Khan, A. Experimental evaluation of a vibration-based leak detection technique for
water pipelines. Struct. Infrastruct. Eng. 2018, 14, 46–55.

Gurley, K.; Kareem, A. Applications of wavelet transforms in earthquake, wind and ocean engineering. Eng. Struct. 1999,
21, 149–167.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.

Kingma, D.P.; Jimmy, B. Adam: A method for stochastic optimization. arXiv 2014, arXiv:1412.6980.

Son, H.; Choi, H.; Seong, H.; Kim, C. Detection of construction workers under varying poses and changing background in image

sequences via very deep residual networks. Autom. Constr. 2019, 99, 27–38.

Dung, C.V.; Sekiya, H.; Hirano, S.; Okatani, T.; Miki, C. A vision-based method for crack detection in gusset plate welded joints of

steel bridges using deep convolutional neural networks. Autom. Constr. 2019, 102, 217–229.