Tiny machine learning with convolutional neural network for intelligent radiation monitoring in nuclear installation

Istofa, Gina Kusuma, Firliyani Rahmatia Ningsih, Joko Triyanto, I Putu Susila, Atang Susila
Research Center for Nuclear Beam Analysis Technology, Research Organization for Nuclear Energy,
National Research and Innovation Agency, Tangerang Selatan, Indonesia

Article Info

Article history:

Received May 5, 2025
Revised Sep 25, 2025
Accepted Nov 23, 2025

Keywords:

Convolutional neural network
Monitoring station
Radionuclide
Raspberry Pi
Tiny machine learning

ABSTRACT

This study focuses on developing an intelligent radiation monitoring system capable of operating on a low-power single-board computer (Raspberry Pi) for deployment in remote monitoring stations within nuclear facility environments. The proposed system utilizes a radionuclide identification method based on tiny machine learning (TinyML) with a convolutional neural network (CNN) architecture. The radionuclide dataset was acquired through measurements of standard radiation sources, with variations in distance, exposure time, and combinations of multiple sources-including Cs-137, Co-60, Cs-134, and Eu-152. The radiation intensity data from detector measurements were structured into a response matrix and subsequently converted into a grayscale image dataset for model training. Keras is used to design and train machine learning models, while Tensor Flow Lite is used to model size reduction. Experimental results demonstrate that the developed model achieves an accuracy of 99.338% for Keras model trained on computer and 84.568% after deployment on the Raspberry Pi. Furthermore, this study successfully designed and embedded the TinyML model into an environment radiation monitoring system at the PUSPIPTEK nuclear installation.

Corresponding Author:

Istofa
Research Center for Nuclear Beam Analysis Technology, Research Organization for Nuclear Energy,
National Research and Innovation Agency
Tangerang Selatan, Banten, Indonesia
Email: isto001@brin.go.id

1. INTRODUCTION

The utilization and development of artificial intelligence (AI), for instance machine learning (ML) and deep learning (DL), have advanced significantly in recent years, revolutionizing data analysis and computational tasks by enabling applications to operate intelligently [1]. Real-time monitoring of environmental radiation is an important requirement in nuclear safety and risk mitigation systems [2]. However, a frequent challenge is the efficiency of data transmission from the field to the control center, especially when the system monitors the radiation spectrum continuously [3]. Tiny machine learning (TinyML) provides an innovative approach by deploying ML models on low-power, resource-constrained edge devices, enabling real-time on-device inference [4], [5]. This system supports real-time analytics, which enhances decision-making speed and improves the overall responsiveness [6]. Such capabilities are particularly critical for time-sensitive applications, including autonomous vehicles [7], healthcare monitoring [8], [9], and early warning systems [10], where processing delays could lead to severe consequences. Therefore, in the context of environmental radiation monitoring, the implementation of TinyML can serve as both an early detection system and a decision-support tool, effectively mitigating the potential impacts of radiation incidents [11].

Numerous prior studies have utilized simulated data generated through Monte Carlo [12]–[14] and Geant4 applications [15]–[17]. As demonstrated in the study by Altayeb *et al.* the majority of existing research leveraging gamma-ray spectrum for radionuclide identification and utilizing silicon photomultiplier (SiPM) [5] for the sensor or scintillation detector [2]. Additionally, various machine learning (ML) techniques [18], [19], including artificial neural network (ANN) [20]–[22] and the convolutional neural network (CNN) [23]–[25], have been explored for developing automated models. However, limited research has focused on implementing the trained models within TinyML systems for radionuclide classification.

The use of TinyML aims to overcome problems related to high data rate transmission and limited resources at monitoring stations. With local data processing capabilities, TinyML can reduce the need for data transmission, and efficiently, quickly, and portably identify radionuclides [26]. In this study, an intelligent system-based radiation monitoring station was developed to identify radionuclides directly in the field. With this identification capability, the system can selectively transmit data only when spectrum irregularities or abnormalities are detected, thereby reducing transmission load and improving data communication efficiency. To support intelligence on edge devices, a TinyML approach was implemented to enable localized spectrum analysis with minimal resource consumption. The first contribution of this paper is building a dataset based on real experiment including the background environment in nuclear installation which comes from the gamma spectrum energy converted to a grayscale image. The second is designing a model with high accuracy and embedding it in a low power consumption device to apply TinyML to recognize the types of radionuclides released in the environment. A performance evaluation was tested to compare the efficiency of the system before and after conversion to TinyML, including aspects of model size reduction, inference speed, and impact on identification quality. Discussions also focused on the causes of data load reduction and its implications for system reliability. As a continuation, long-term integration of TinyML into the monitoring station will be carried out to test the stability and adaptivity of the system in more complex field conditions.

2. METHOD

The methodology implemented in this study commenced with the acquisition of gamma-ray spectral data from a radiation detection system, shown in Figure 1. These spectral datasets were subsequently transformed into grayscale image representations, serving as input features for training a CNN. This approach lies in the CNN's proven capability to extract spatial patterns and features from two-dimensional image inputs, making it suitable for recognizing spectral signatures associated with different radionuclides. The training process was initially conducted on a personal computer (PC) to optimize the model parameters and evaluate its learning performance. Once a satisfactory level of classification accuracy and model generalization was achieved, the trained CNN model was converted and deployed onto a Raspberry Pi, a low-power edge computing device which represent the hardware configuration of an intelligent radiation monitoring station. The embedded model was then subjected to a series of field trials designed to simulate realistic environmental conditions. These field evaluations aimed to verify the inference accuracy and robustness of the CNN when operating in situ, as well as to assess the feasibility of real-time spectral classification on resource-constrained hardware. Validation is crucial in ensuring the reliability and responsiveness of intelligent monitoring systems in practical applications.

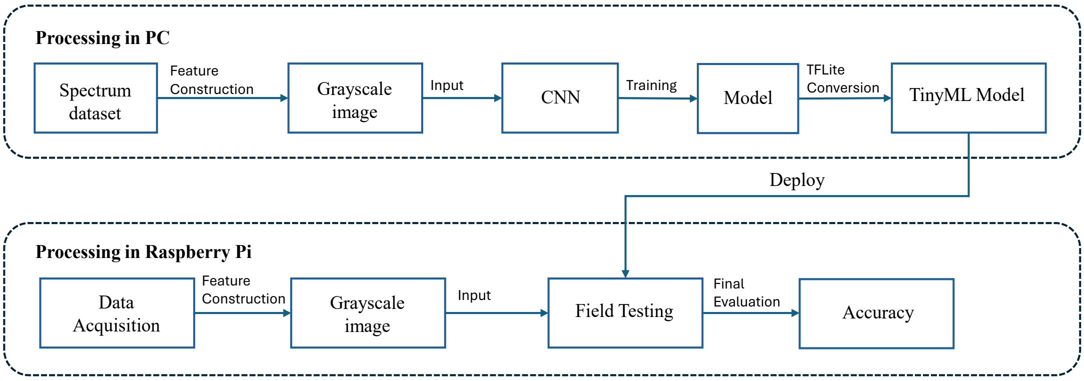


Figure 1. Workflow diagram of PC and Raspberry Pi process

2.1. Grayscale image conversion and dataset construction

The experiments were conducted by varying several parameters, including the type of radionuclide, measurement duration, and the distance between the source and the detector. These variations were intended to generate diverse background data that closely reflect actual environmental conditions. Radiation signals were acquired using a scintillation detector, producing intensity data that were plotted into a response matrix. Based on the characteristic energy peaks of each radionuclide, labelling was applied to the resulting spectral data. Subsequently, the labelled spectra were converted and mapped into grayscale images (feature transfer), then arranged into a dataset that suitable for application supervised learning algorithms in computer. Figure 2 explains the conversion of gamma spectrum to grayscale image using the z curve method [11].

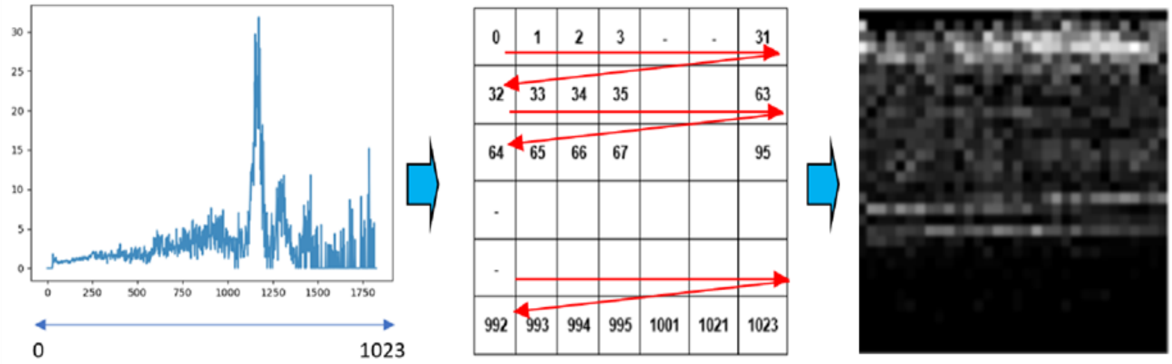


Figure 2. Grayscale image conversion [13]

2.2. Design and deploy the TinyML model

Figure 3 shows the model architecture starting with an input layer that receives a 32×32×1 grayscale image. First, it applies a Conv2D layer with 32 filters and rectified linear unit (ReLU) activation to extract basic features like edges, followed by a MaxPooling2D layer to down sample the feature maps. Next, a Conv2D layer with 128 filters and ReLU activation captures more complex patterns, again followed by a MaxPooling2D layer to reduce the spatial size. The third step involves a flatten layer to convert the feature map that received from the max-pooling layer into a format that the dense layers can understand. Finally, the model ends with a dense layer with a few neurons equal to the number of classes, using SoftMax activation to output class probabilities, in this case there are four classes, Cs-137, Co-60, Cs-134, and Eu-152.

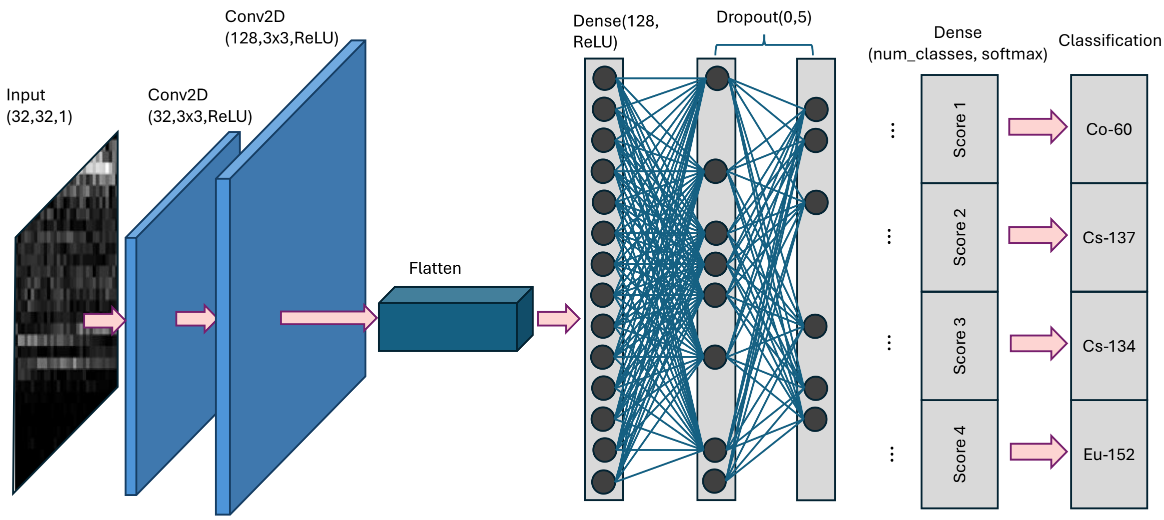


Figure 3. Model architecture to identify radionuclide

After the model is designed, tested and validated, it is then embedded in the hardware system with the help of TensorFlow Lite (TFLite). TFLite is used to convert ML models that initially have large memory and are heavy to run, into smaller, faster, and more efficient versions so that they can be run on devices that have limited resources. The model is converted into TFLite format for optimization, so the model size can be smaller to be deployed to Raspberry Pi. After the model is embedded in the hardware, testing is done again using radionuclides as shown in Figure 4. Implementation of model testing using a NaI(Tl) detector and a standard radiation source. Data collection for 30 seconds with a distance of 50 cm from the source to the detector. Each test was carried out 20 times. The position of the detector and standard source was placed parallel as when the dataset was taken and was also kept away from walls or objects that could cause backscattering that interfered with the identification results.

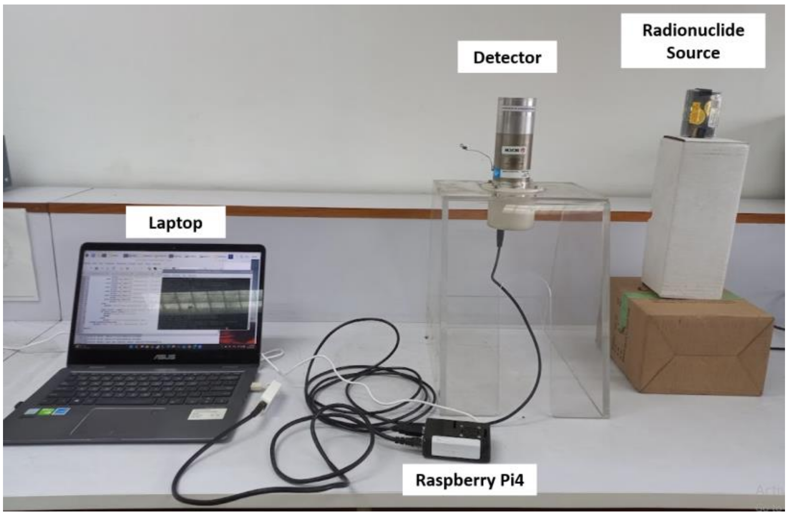


Figure 4. Testing of TinyML models embedded in system hardware using radionuclide standard

3. RESULTS AND DISCUSSION

3.1. Feature extraction and transfer to image

The transformation of radionuclide spectra into an image dataset is illustrated in Figure 5. This figure presents examples of the transformation results for Cs-137, Co-60, Cs-134, and Eu-152, obtained with a source-to-detector distance of 20 cm and a measurement time of 60 seconds. Figure 5(a) shows the transformation of the gamma spectrum of Co-60 into a grayscale image, displaying two distinct energy peaks. Figure 5(b) depicts the transformation for Cs-137, which exhibits a single peak. Figure 5(c) corresponds to Cs-134, and Figure 5(d) to Eu-152, both of which contain multiple peaks. The gamma spectrum images represent the original spectra of each radionuclide, including their characteristic energy peaks. These original spectra were then converted into normalized one-dimensional spectra after background correction, before being transformed into grayscale images. Using grayscale images (1-channel) instead of RGB (3-channel) for training classification models reduces computational complexity and memory usage, making them more efficient for deployment. Since grayscale images have only one intensity channel, they require less storage, faster processing, and smaller model sizes, which is beneficial for TinyML applications.

These grayscale images are both the dataset and the input for the CNN model identification process. The color gradation for each pixel of the image carries information about the value of radiation intensity at certain channel positions that represent the characteristics of certain types of radionuclides. These grayscale images were trained using Keras in computer, tested and validated with a data ratio of 70:20:10. The radionuclide identification model obtained from the initial training could not be directly embedded into a TinyML environment. In TinyML, Keras is used to design and train machine learning models on powerful systems, while TFLite optimizes these models for deployment on resource-constrained edge devices. Converting a Keras model to TFLite reduces memory usage and speeds up inference, making it feasible for limited random-access memory (RAM). Despite the model may slightly reduce accuracy, it is necessary to fit models into tiny devices, ensuring efficient, low-power execution without requiring cloud dependency. Thus, TFLite support TinyML to be applicable on environment monitoring device in remote area.

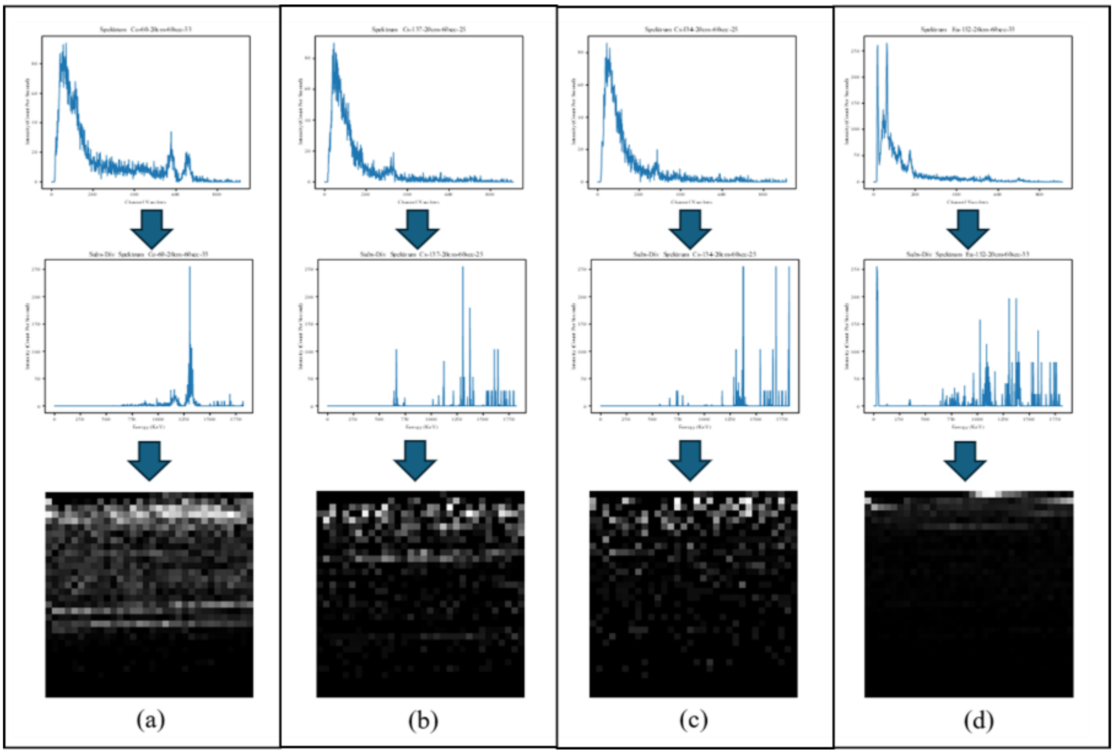


Figure 5. Feature transfer from spectrum vector to image (a) Co-60, (b) Cs-137, (c) Cs-134, and (d) Eu-152

3.2. System prediction accuracy

In the context of TinyML, machine learning models which deployed on Raspberry must combine the function of Keras and TensorFlow Lite (TFLite) to provide an efficient framework. Keras excels in model construction and training, while TFLite provides the necessary optimizations for real-time, on-device inference. Keras is widely used for model design and training, while TFLite is a lightweight inference engine which enables model execution on edge devices. It allows models trained with Keras to be converted into an efficient TFLite format, significantly reducing file size and computational requirements through optimizations such as quantization and operator fusion.

At the conversion from Keras to TFLite format, default setting is implemented. This means that optimization and quantization process is not utilized to obtain efficient and smaller file size model. However, even though the conversion process does not conduct optimization and quantization, the TFLite model which resulted is still smaller than Keras unto its half-size. Half-size result is achieved from conversion process because at this process, data pruning and implicit compression are utilized. When converted into TFLite, the unemployed parts at inference are discarded, including node Graf training. Through implicit compression, the flat buffer structure stores information into very compact form without Python framework overhead. Other parameters such as metadata, checkpoint, graph ops, signature and training information in Keras model, are also abandoned during the conversion process. The remaining information is then become structured and serialized. Therefore, the half-size compression model is not processed by quantization from float32 to others format such as int8, int16, but rather through data pruning and compression processes.

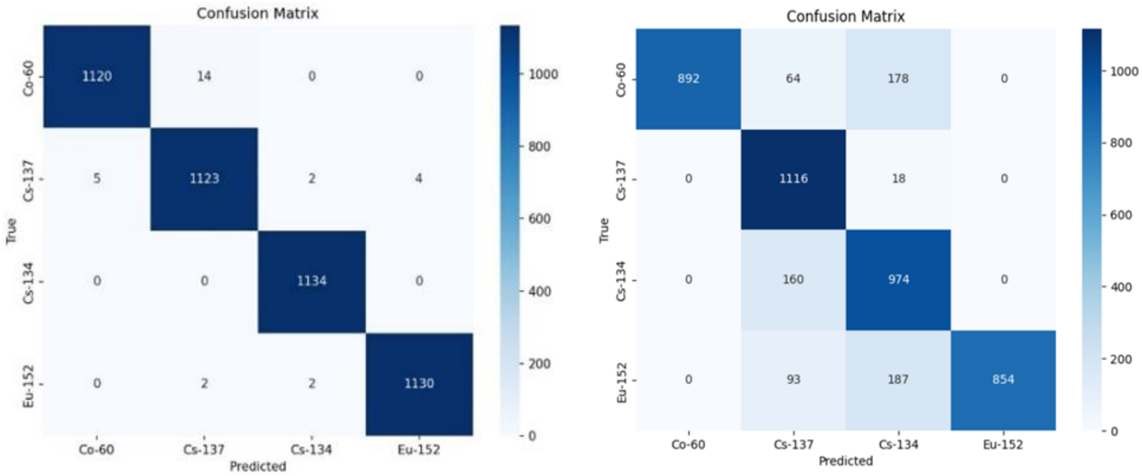
The confusion matrix results explained in Figure 6 shows that the model performs very well, since most predictions fall along the diagonal (correct classifications). From the confusion matrix results, we can calculate the accuracy, precision, recall and f1 score for each class, as presented in Table 1. The results show in a very good value, especially in the Cs-134 class which gets the highest accuracy result of 99.890%, seen in Figure 6(a). Overall accuracy for the four classes was 99.338%. For information, the size of the storage memory of the model that has been trained using Keras is 8 MB, while the size of the memory after being transformed in TFLite is 2MB, reduced by approximately one quarter time.

Figure 6(b) presents the confusion matrix results of the TFLite model with a lower memory size tested on a Raspberry device. The results indicate a lower accuracy for each class compared to previous tests. The highest accuracy is achieved for the Co-60 class at 94.665%, while the Cs-134 class records the lowest accuracy at 88.029%, as shown in Table 1. The overall model accuracy is 84.568%. This decrease in performance may be attributed, in part, to the Raspberry's relatively limited processing capabilities compared to a standard computer. Nevertheless, for environmental monitoring station applications, this level of accuracy remains acceptable, although further improvements are planned to enhance the model's performance.

3.3. Evaluation of the TinyML model embedded in hardware

The most accurate architectural model was integrated into the Raspberry Pi hardware system and evaluated through direct testing with radionuclides, as illustrated in Figure 4. The experimental procedure involved partitioning the data according to radionuclide type. Table 2 displays the test results, which reveal variations in accuracy across different classes. The accuracy results of the CNN model test in TinyML using direct measurement data from the detector are lower compared to the training dataset. This difference is caused by several factors related to data characteristics, the real environment, and the limitations of the TinyML platform. Direct measurement data usually has higher variability due to noise from the sensor, environmental fluctuations, or hardware inconsistencies. If the training dataset does not include these variations, the model has difficulty recognizing patterns in real data.

In addition, training datasets are often taken from controlled conditions, and therefore less representative of the actual conditions under which direct measurements are made. For example, variations in radionuclide activity, environmental interference, or different detector characteristics are not always represented in the training dataset. If the dataset lacks these variations, the CNN model tends to simply memorize patterns from the training data without being able to generalize to new data. This can also be exacerbated if the model suffers from overfitting, where the model focuses too much on specific patterns in the training dataset and is unable to handle variations in the direct measurement data.



(a) (b)

Figure 6. Confusion matrix (a) using "Keras" model and (b) using "TFLite" model

Table 1. Model performance based on Keras training vs TFLite training

Matrix evaluation	Model trained using Keras				Model trained using TFLite			
	Co-60	Cs-137	Cs-134	Eu-152	Co-60	Cs-137	Cs-134	Eu-152
Accuracy	99.559	99.405	99.890	99.824	94.665	92.615	88.029	93.827
Precision	99.467	98.595	99.648	99.647	100.000	77.879	71.776	100.000
Recall	98.765	99.030	99.912	99.647	78.660	98.413	85.891	75.309
F1 score	99.115	98.812	99.780	99.647	88.055	86.950	78.202	85.915
Macro-precision	99.339				87.414			
Macro-recall	99.338				84.568			
Macro-F1	99.337				84.781			
Overall Accuracy	99.338				84.568			

In addition, training datasets are often taken from controlled conditions, and therefore less representative of the actual conditions under which direct measurements are made. For example, variations in radionuclide activity, environmental interference, or different detector characteristics are not always represented in the training dataset. If the dataset lacks these variations, the CNN model tends to simply memorize patterns from the training data without being able to generalize to new data. This can also be exacerbated if the model suffers from overfitting, where the model focuses too much on specific patterns in the training dataset and is unable to handle variations in the direct measurement data.

Table 2. Results of testing model-4 on Raspberry with detector measurement data

Dataset	Number of Tests	True Prediction	False Prediction	Accuracy (%)
Co-60	20	14	6	70
Cs-137	20	17	3	85
Cs-134	20	16	4	80
Eu-152	20	17	3	85

According to Table 2, it can be analyzed that the high accuracy of Cs-134 is most likely due to its spectral characteristics that resemble the background, making it easier for the model to recognize the pattern compared to other more complex radionuclides. In addition, bias in the training dataset, such as the dominance of data that resembles the background or simple patterns, can improve the prediction of Cs-134. When Cs-134 appears in combination with other radionuclides, its stable contribution helps the model recognize the overall pattern better. However, differences in accuracy between radionuclides can also be influenced by noise, data imbalance, or lack of variation in the training dataset, which require improvements to improve model generalization.

Environmental factors also play a significant role in reducing accuracy. Real-world direct measurements are often affected by conditions such as temperature, humidity, or electromagnetic interference. These conditions are not always captured in the training dataset, so the model cannot adapt well. The signal variations produced by the radiation detector in direct measurements can also differ significantly from the training data, making the model's predictions less accurate.

The tested TinyML system, implemented on a Raspberry Pi, has been integrated with additional microcontroller components within an environmental monitoring station, as illustrated in Figure 7. This station encompasses not only a model for monitoring radionuclide releases but also sensors for collecting meteorological and humidity data. To support energy autonomy in remote or off-grid locations, the system is equipped with solar panels. In the future, this station is intended to be deployed in isolated areas to monitor radionuclide dispersion carried by wind from various sources.

Figure 7. Raspberry pi hardware integrated into environmental monitoring station

4. CONCLUSION

This study demonstrates the successful implementation of a TinyML model for real-time classification of radionuclides in an embedded environmental monitoring system. Keras and TensorFlow Lite serve complementary roles in the development and deployment of TinyML models on devices like the Raspberry Pi. The optimized model based on result achieved high accuracy of 99.338% trained using Keras, and 84.568% trained using TFLite. For the real measurement using the hardware, the highest accuracy obtained 85% for Eu-152 class. This integrated system not only monitors radioactive releases but also tracks weather and humidity data, enhancing environmental surveillance capabilities. With solar-powered energy autonomy, the solution is suitable for remote deployments, enabling early detection of radionuclide dispersion via wind currents. Future work will focus on scaling the system for wider geographic coverage, improving model robustness with additional data, and integrating wireless sensor networks for real-time data transmission.

FUNDING INFORMATION

This research was funded by National Research and Innovation Agency of Indonesia.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Istofa	✓	✓		✓	✓	✓		✓	✓				✓	✓
Gina Kusuma		✓	✓			✓		✓	✓		✓			
Firliyani Rahmatia Ningsih			✓	✓			✓		✓		✓		✓	
Joko Triyanto			✓					✓		✓				
I Putu Susila	✓	✓		✓	✓					✓		✓		
Atang Susila		✓		✓			✓			✓				

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

The authors state no conflict of interest.

DATA AVAILABILITY

The Authors state for data availability that supports the findings of this study are available on request from the corresponding author.

REFERENCES

[1] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN Computer Science*, vol. 2, no. 3, May 2021, doi: 10.1007/s42979-021-00592-x.

[2] S. M. Galib, P. K. Bhowmik, A. V. Avachat, and H. K. Lee, "A comparative study of machine learning methods for automated identification of radioisotopes using NaI gamma-ray spectra," *Nuclear Engineering and Technology*, vol. 53, no. 12, pp. 4072–4079, 2021, doi: 10.1016/j.net.2021.06.020.

[3] M. Soori, B. Arezoo, and R. Dastres, "Artificial intelligence, machine learning and deep learning in advanced robotics, a review," *Cognitive Robotics*, vol. 3, no. March, pp. 54–70, 2023, doi: 10.1016/j.cogr.2023.04.001.

[4] P. P. Ray, "A review on TinyML: State-of-the-art and prospects," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 4, pp. 1595–1623, Apr. 2022, doi: 10.1016/j.jksuci.2021.11.019.

[5] M. Altayeb, M. Zennaro, and E. Pietrosemoli, "TinyML Gamma radiation classifier," *Nuclear Engineering and Technology*, vol. 55, no. 2, pp. 443–451, 2023, doi: 10.1016/j.net.2022.09.032.

[6] A. Elhanashi, P. Dini, S. Saponara, and Q. Zheng, "Advancements in TinyML: applications, limitations, and impact on IoT devices," *Electronics (Switzerland)*, vol. 13, no. 17, 2024, doi: 10.3390/electronics13173562.

[7] M. de Prado *et al.*, "Robustifying the deployment of tinyML models for autonomous mini-vehicles," *Proceedings - IEEE International Symposium on Circuits and Systems*, vol. 2021-May, pp. 1–16, 2021, doi: 10.3390/s21041339.

[8] M. Bhamare, P. V. Kulkarni, R. Rane, S. Bobde, and R. Patankar, "TinyML applications and use cases for healthcare," *TinyML for Edge Intelligence in IoT and LPWAN Networks*, vol. 15, no. 1, pp. 331–353, 2024, doi: 10.1016/B978-0-44-322202-3.00019-1.

[9] W. Ding, J. Yu, C. Zheng, P. Fu, and Q. Huang, "Machine learning-based noninvasive quantification of single-imaging session dynamic PET-CT in oncology," *IEEE Transactions on Medical Imaging*, vol. 41, no. 2, pp. 347–359, 2022, doi: 10.1109/TMI.2021.3112783.

[10] D. Sansone and A. Zhu, "Using machine learning to create an early warning system for welfare recipients," *Oxford Bulletin of Economics and Statistics*, vol. 85, no. 5, pp. 959–992, 2023, doi: 10.1111/obes.12550.

[11] D. Liang *et al.*, "Rapid nuclide identification algorithm based on convolutional neural network," *Annals of Nuclear Energy*, vol. 133. Elsevier Ltd, pp. 483–490, Nov. 2019, doi: 10.1016/j.anucene.2019.05.051.

[12] Y. Wang *et al.*, "Explainable radionuclide identification algorithm based on the convolutional neural network and class activation mapping," *Nuclear Engineering and Technology*, vol. 54, no. 12, pp. 4684–4692, 2022, doi: 10.1016/j.net.2022.08.011.

[13] S. Qi *et al.*, "Radionuclide identification method for NaI low-count gamma-ray spectra using artificial neural network," *Nuclear Engineering and Technology*, vol. 54, no. 1, pp. 269–274, 2022, doi: 10.1016/j.net.2021.07.025.

[14] G. Daniel, F. Ceraudo, O. Limousin, D. Maier, and A. Meuris, "Automatic and real-time identification of radionuclides in Gamma-Ray Spectra: a new method based on convolutional neural network trained with synthetic data set," *IEEE Transactions on Nuclear Science*, vol. 67, no. 4, pp. 644–653, Apr. 2020, doi: 10.1109/TNS.2020.2969703.

[15] H. L. Liu, H. B. Ji, J. M. Zhang, C. L. Zhang, J. Lu, and X. H. Feng, "A novel approach for feature extraction from a gamma-ray

energy spectrum based on image descriptor transferring for radionuclide identification," *Nuclear Science and Techniques*, vol. 33, no. 12, Dec. 2022, doi: 10.1007/s41365-022-01150-7.
[16] S. X. Zeng *et al.*, "Identification of low-count, low-resolution gamma spectral radionuclide using 2DCNN-BiLSTM neural network," *Nuclear Engineering and Technology*, vol. 57, no. 12, Dec. 2025, doi: 10.1016/j.net.2025.103854.
[17] C. Li *et al.*, "A new radionuclide identification method for low-count energy spectra with multiple radionuclides," *Applied Radiation and Isotopes*, vol. 185, Jul. 2022, doi: 10.1016/j.apradiso.2022.110219.
[18] G. Kusuma, R. M. Saryadi, S. K. Wijaya, S. Soekirno, P. Prajitno, and I. P. Susila, "Radionuclide identification analysis using machine learning and GEANT4 simulation," *AIP Conference Proceedings*, vol. 2381, no. November, 2021, doi: 10.1063/5.0067593.
[19] H. C. Lee *et al.*, "Radionuclide identification based on energy-weighted algorithm and machine learning applied to a multi-array plastic scintillator," *Nuclear Engineering and Technology*, no. May, 2023, doi: 10.1016/j.net.2023.07.005.
[20] C. Bobin, O. Bichler, V. Lourenço, C. Thiam, and M. Thévenin, "Real-time radionuclide identification in γ-emitter mixtures based on spiking neural network," *Applied Radiation and Isotopes*, vol. 109, pp. 405–409, Mar. 2016, doi: 10.1016/j.apradiso.2015.12.029.
[21] M. Kamuda and C. J. Sullivan, "An automated isotope identification and quantification algorithm for isotope mixtures in low-resolution gamma-ray spectra," *Radiation Physics and Chemistry*, vol. 155, pp. 281–286, Feb. 2019, doi: 10.1016/j.radphyschem.2018.06.017.
[22] M. Kamuda, J. Stinnett, and C. J. Sullivan, "Automated isotope identification algorithm using artificial neural networks," *IEEE Transactions on Nuclear Science*, vol. 64, no. 7, pp. 1858–1864, Jul. 2017, doi: 10.1109/TNS.2017.2693152.
[23] A. Sharma, E. Vans, D. Shigemizu, K. A. Boroevich, and T. Tsunoda, "DeepInsight: A methodology to transform a non-image data to an image for convolution neural network architecture," *Scientific Reports*, vol. 9, no. 1, pp. 1–7, 2019, doi: 10.1038/s41598-019-47765-6.
[24] R. S. Saeed and B. K. O. Chabor Alwawi, "A binary classification model of COVID-19 based on convolution neural network," *Bulletin of Electrical Engineering and Informatics*, vol. 12, no. 3, pp. 1413–1417, 2023, doi: 10.11591/eei.v12i3.4832.
[25] F. Tempola, R. Wardoyo, A. Musdholifah, R. Rosihan, and L. Sumaryanti, "Classification of clove types using convolution neural network algorithm with optimizing hyperparamters," *Bulletin of Electrical Engineering and Informatics*, vol. 13, no. 1, pp. 444–452, 2024, doi: 10.11591/eei.v13i1.5533.
[26] D. L. Dutta and S. Bharali, "TinyML meets IoT: a comprehensive survey," *Internet of Things*, vol. 16, Dec. 2021, doi: 10.1016/j.iot.2021.100461.

BIOGRAPHIES OF AUTHORS

Istofa is a researcher at the Research Center for Nuclear Beam Analysis Technology, Research Organization for Nuclear Energy, National Research and Innovation Agency, Indonesia. He obtained his bachelor's degree in nuclear engineering from Gadjah Mada University, Indonesia, in 2002. After that, he continued his study in the magister science program at the Physics Department, Universitas Indonesia, graduating in 2024. His research interests include electronics, nuclear instrumentation, embedded systems, and artificial intelligence. He can be contacted at email: isto001@brin.go.id.

Gina Kusuma is a researcher at the Research Center for Nuclear Beam Analysis Technology, Research Organization for Nuclear Energy, National Research and Innovation Agency, Indonesia. He was born in Bogor, West Java. He obtained his bachelor's degree in nuclear technophysics from the Polytechnic Institute of Nuclear Technology – BATAN Yogyakarta in 2013 and then continued his study in the magister science program at the Physics Department, Universitas Indonesia, graduating in 2021. His research fields are nuclear instrumentation, electronics, embedded systems, desktop programming, and artificial intelligence. He can be contacted at email: gina004@brin.go.id.

Firliyani Rahmatia Ningsih is a researcher at the Research Center for Nuclear Beam Analysis Technology. She was born in Surabaya, East Java. She graduated from the bachelor's program in nuclear engineering at Universitas Gadjah Mada in 2015. After that, she continued her study in the magister science program at the Physics Department, Universitas Indonesia, graduating in 2023. She is currently active as an assistant researcher at the National Research and Innovation Agency of the Republic of Indonesia. Her research interests include photon beam technology, radiation measurement, artificial intelligence, and radiation imaging techniques. She can be contacted at email: firl001@brin.go.id.

Joko Triyanto is a researcher at the Research Center for Nuclear Beam Analysis Technology. He was born in Wonogiri, Central Java. He graduated from the undergraduate program in nuclear engineering at Universitas Gadjah Mada in 1995. After that, he continued his studies in the master of engineering program at the Department of Electrical Engineering, University of Indonesia, graduating in 2012. He is currently active as a researcher at the National Research and Innovation Agency of the Republic of Indonesia. His research interests include nuclear instrumentation and control systems, radiation measurement, and embedded systems. He can be contacted at email: joko019@brin.go.id.

I Putu Susila is a researcher at the Research Center for Nuclear Beam Analysis Technology, Research Organization for Nuclear Energy, National Research and Innovation Agency, Indonesia. He received his doctoral degree from Tohoku University, Japan, in 2009. His research interests include software development, image processing, and embedded systems in biomedical and health research. He is particularly interested in developing healthcare equipment, including hardware, software, and open-source applications. He can be contacted at email: iput001@brin.go.id.

Atang Susila is a researcher at the Research Center for Nuclear Beam Analysis Technology, Research Organization for Nuclear Energy, National Research and Innovation Agency, Indonesia. He obtained his bachelor's degree in nuclear engineering from Gadjah Mada University, Indonesia, in 1988. After that, he continued his study in the master's program in sciences and engineering at the University of Tsukuba, Japan, graduating in 2001. His research interests include electronics, nuclear instrumentation, embedded systems, and artificial intelligence. He can be contacted at email: atan001@brin.go.id.