International Core Journal of Engineering Volume 11 Issue 6, 2025
ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

An Edge-Intelligent Flotation Monitoring and Control System based on 8051 MCU and Offline EfficientNet-B5 Inference on Raspberry Pi

Boyu Yang[1], Fangyuan Yao[2], Wenqi Bai[2], Yuxin Zhang[2], Hang Zhu[2], Zhaoning Yin[2]

[1] North China University of Science and Technology, Tangshan, 063000, China

[2] Zhangjiakou University, Zhangjiakou, 075000, China

Abstract

To address the limitations of traditional flotation processes such as delayed feedback and low detection precision, this paper proposes an edge-intelligent flotation monitoring and control system. The system integrates a low-power 8051 microcontroller unit (MCU) for real-time multimodal sensor data acquisition with a Raspberry Pi for local image-based state recognition. An EfficientNet-B5 model, pre-trained and quantized via TensorFlow Lite, is deployed offline on the Raspberry Pi to enable high-accuracy flotation froth classification with an average inference latency of 250–350 ms. Through attention-based feature refinement and float32-quantized model optimization, the proposed system achieves a 67.5% reduction in memory usage and an 18.4% decrease in average inference time compared to the original model. In addition, the mean and peak power consumption are reduced by 18.3% and 9.5%, respectively, ensuring stable performance in low-power embedded environments. A Tkinter-based GUI further supports both manual and automatic control modes, enabling intelligent adjustments in flotation operations. The proposed architecture realizes a closed-loop control chain from sensing to decision-making and actuation, offering a scalable and energy-efficient solution for intelligent mineral processing at the edge.

Keywords

Edge Computing; 8051 MCU; Raspberry Pi; EfficientNet-B5; Flotation Monitoring; Intelligent Control; Embedded System; Offline Inference.

1. Introduction

With the accelerating demand for strategic minerals, the efficient utilization of lead and zinc ores has become a national priority in many resource-driven economies. In particular, China-accounting for over 40% of global zinc and lead production-has witnessed a continuous increase in output over recent years. As shown in Figure 1, the country's annual lead and zinc outputs rose from 5.11 and 5.88 million tons in 2018 to 7.36 and 6.56 million tons in 2021, respectively, highlighting the expanding scale of mineral beneficiation operations [1]. This surge in production, however, has placed increasing pressure on the efficiency, sustainability, and intelligence of flotation processes, which remain the primary method for selective separation of sulfide minerals. Conventional flotation control systems typically rely on manual observation and rule-based adjustment, which struggle to cope with real-time variations in ore grade, bubble morphology, and froth stability, especially under high-throughput conditions [2]. Moreover, excessive dosing of collectors and frothers-due to the absence of accurate state feedback-not only increases operating costs but also contributes to wastewater contamination and downstream treatment burdens [3]. These limitations underscore the urgent need

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

for intelligent, low-cost, and adaptive flotation control systems that can support green and automated mineral processing.

In response to the limitations of manual flotation monitoring, recent research has increasingly turned to computer vision and deep learning for froth image classification. Convolutional neural networks (CNNs) have demonstrated promising accuracy in identifying flotation states from visual features such as bubble size, texture, and color distribution [4]. For example, Shi et al. proposed a CNN-based froth recognition system that achieved over 92% classification accuracy under laboratory conditions [5]. However, the majority of these models require GPU acceleration and cloud-based inference to support real-time processing, which poses significant deployment barriers in remote mining sites [6]. Moreover, fluctuating lighting, low signal-to-noise ratios, and lack of online sensor fusion often degrade model robustness in actual industrial environments [7]. As such, a key challenge remains in designing lightweight, high-accuracy inference systems that can operate reliably on resource-constrained embedded platforms without sacrificing detection performance [8].

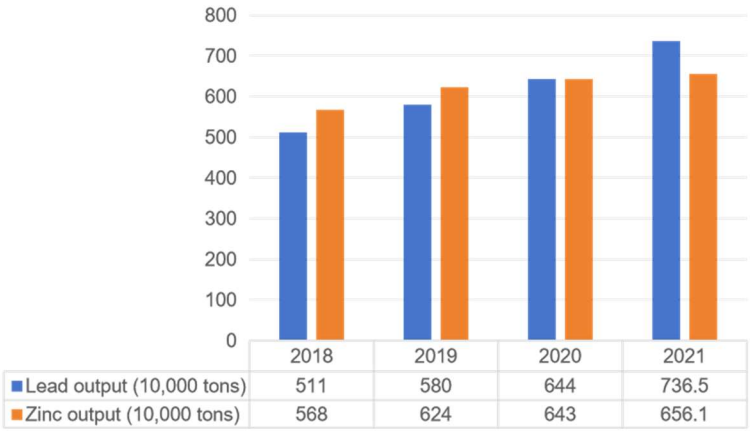


Figure 1. Annual lead and zinc production in China from 2018 to 2021 (10,000 tons)

To enable real-time deployment in field environments, single-board computers like the Raspberry Pi have become attractive edge computing platforms for industrial vision tasks. Their compact size, low power demand, and open-source ecosystem make them suitable for scenarios with energy and connectivity constraints [9]. Kumar et al. demonstrated a Raspberry Pi-based fault detection system with inference delays under 400 ms using lightweight models [10]. However, limited CPU and memory resources restrict deployment of conventional deep networks with high parameter counts and floating-point operations [11]. To address this, hardware-aware optimizations—such as quantization, pruning, and float32-to-int8 conversion—are essential for maintaining accuracy while ensuring efficient edge inference [12].

Among CNN architectures, EfficientNet-B5 is notable for its compound scaling strategy, which jointly optimizes network depth, width, and resolution. Compared to traditional models like ResNet and VGG, it offers higher ImageNet accuracy with significantly fewer parameters and FLOPs [13]. Tan and Le reported 83.6% top-1 accuracy using only 1/30 of ResNet152's parameters [14]. Furthermore, EfficientNet-B5 is highly compatible with TensorFlow Lite (TFLite), enabling offline inference through quantization and optimization without GPU support [15]. Several studies have deployed EfficientNet variants on embedded platforms using float32 quantization, achieving reduced memory use and latency while retaining accuracy [16]. These features make EfficientNet-B5 well suited for edge-based real-time flotation froth classification.

Inspired by the aforementioned advances in edge computing, lightweight neural network deployment, and embedded sensor integration, this paper proposes an edge-intelligent flotation monitoring and control system tailored for mineral processing environments. The system integrates a low-power 8051

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

microcontroller unit (MCU) for real-time multimodal sensor acquisition, including temperature, illumination, and liquid level data, with a Raspberry Pi serving as the local inference hub. A pre-trained EfficientNet-B5 model is optimized via TensorFlow Lite and deployed offline to classify flotation froth states with high accuracy and low latency. The model is further enhanced by attention-based feature refinement and float32 quantization, enabling stable inference within 250–350 ms while reducing power consumption and memory usage. Experimental evaluations demonstrate the system's effectiveness in supporting real-time flotation control under resource-constrained conditions. In addition, a Tkinter-based GUI allows for intuitive manual and automatic control, completing a closed-loop architecture from perception to actuation.

The main contributions of this work are summarized as follows:(1) A hybrid embedded control architecture is proposed, combining a low-power 8051 MCU for real-time sensor acquisition with a Raspberry Pi for edge inference, enabling closed-loop flotation control without cloud dependency.(2) An EfficientNet-B5 model is optimized and deployed via TensorFlow Lite for offline froth image classification, achieving 99.26% recognition accuracy with 18.4% reduction in average inference latency and 67.5% lower memory footprint.(3) A full-stack flotation monitoring interface based on Tkinter is developed to support both manual and automated control strategies, allowing dynamic adjustment of dosing and aeration operations based on inference and sensor fusion feedback.

2. System Architecture

2.1 System Overview

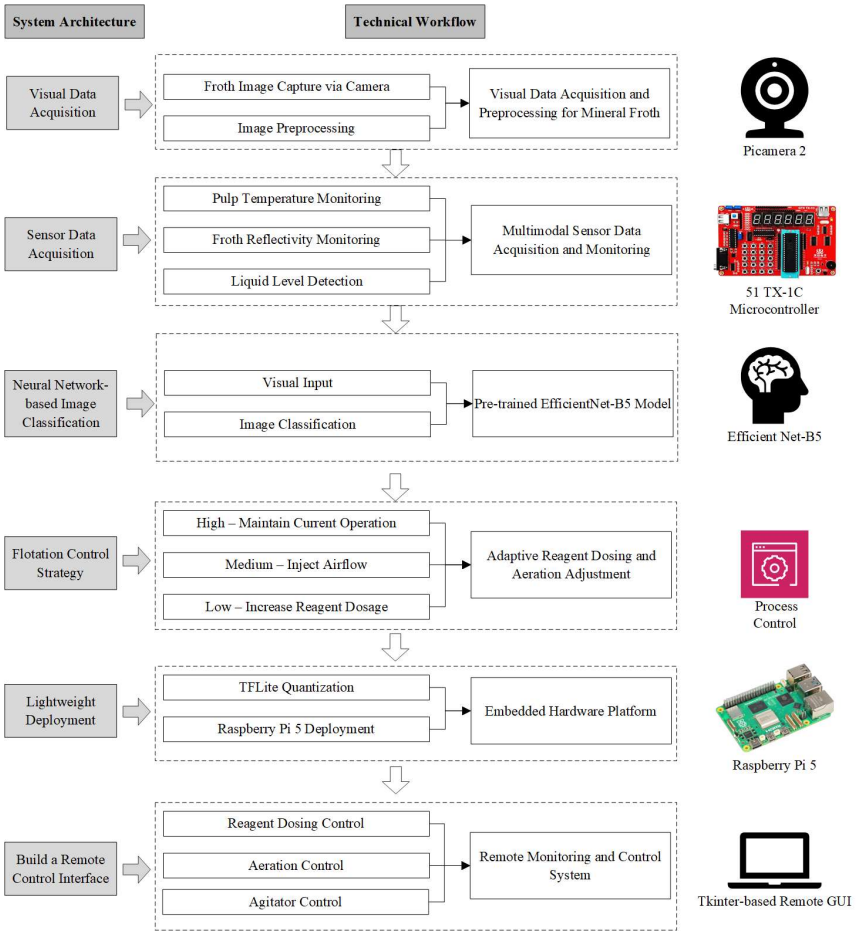


Figure 2. Architecture of the proposed edge-intelligent flotation monitoring system

As illustrated in Figure 2, the proposed flotation monitoring and control system adopts a multi-stage embedded framework that integrates sensor acquisition, visual classification, edge inference, and

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

real-time control. At the front end, froth images are captured in real time by a Picamera2 module installed above the flotation tank. Simultaneously, a low-power 51 TX-1C microcontroller acquires pulp temperature, froth reflectivity, and liquid level information using thermistor, photodiode, and infrared reflective sensors, respectively. These multimodal signals are transmitted via a USB-TTL interface to a Raspberry Pi, which functions as the edge control hub. Captured images are preprocessed and fed into a pre-trained EfficientNet-B5 model optimized with TensorFlow Lite for lightweight deployment. The model classifies froth states into high, medium, or low categories based on visual texture and brightness features.

The classification results are fused with real-time sensor inputs to determine corresponding flotation control strategies. Specifically, when a “high” froth state is detected, the system maintains current operation; for “medium,” it triggers airflow injection; and for “low,” it increases reagent dosage. These decisions are executed locally without reliance on cloud connectivity. The entire model is deployed on a Raspberry Pi 5 with float32 quantization, ensuring inference latency between 250–350 ms and stable performance under constrained hardware conditions. Additionally, a GUI developed using the Tkinter library allows users to monitor real-time sensor values, view classification results, and control dosing, aeration, and agitation functions in both manual and automatic modes. This architecture forms a closed-loop flotation control solution that is lightweight, scalable, and well-suited for deployment in remote or bandwidth-limited mining environments.

2.2 Sensor and Visual Data Acquisition

To enable accurate flotation state recognition, a dedicated visual acquisition pipeline is developed for capturing and preprocessing mineral froth images. As shown in Figure 3, the system utilizes a Picamera2 module to capture images at a resolution of 640×480 pixels. An HDMI real-time display interface is incorporated to facilitate on-site visual supervision during flotation operation. The system is configured to automatically capture an image every 10 seconds, which is stored locally for subsequent inference and model training.

Figure 3. Real-time camera preview during froth image acquisition using Picamera2

Due to challenging imaging conditions-such as uneven lighting, reflections, and complex backgrounds-raw images are preprocessed to enhance feature clarity. First, grayscale conversion is applied to retain luminance while removing color interference. Then, histogram equalization is used to improve local contrast, making the edges of bubbles more distinguishable. After these steps, the dynamic brightness range is stretched to 0–255, ensuring better differentiation of froth structures. Sampled images exhibit high-contrast and well-defined textures, significantly aiding downstream deep feature extraction. For training label calibration, we adopt a publicly available flotation froth dataset released by Yanten et al. [17], which includes annotated images collected from laboratory-scale Magotteaux flotation cells under controlled conditions.The labeled dataset is randomly divided into training, validation, and test sets in a 5:3:2 ratio to ensure balanced learning and evaluation across different flotation froth categories, as illustrated in Figures 4–6.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

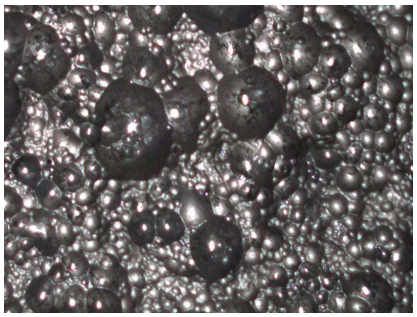


Figure 4. Low froth category

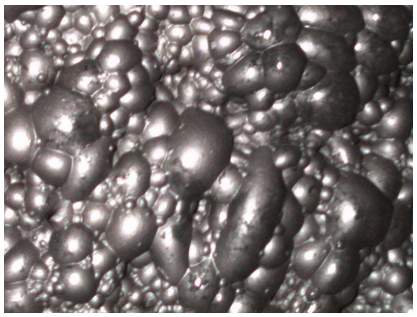


Figure 5. Medium froth category

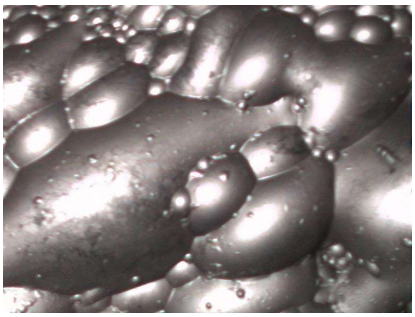


Figure 6. High froth category

2.3 Real-Time Sensor Monitoring Via 8051 MCU

Real-time monitoring of environmental variables is essential for stable flotation control. This system uses a low-power 8051 microcontroller to continuously acquire three physical parameters: pulp temperature, froth reflectivity, and liquid level. The sensors are interfaced with the MCU via GPIO or ADC channels and transmit data through a USB-TTL interface to the Raspberry Pi. The Raspberry Pi parses and visualizes the incoming data using a Tkinter-based GUI, enabling operators to observe the sensor readings and adjust dosing, pumping, and stirring strategies manually or automatically.

Specifically, froth reflectivity is captured using a light-dependent resistor module (Figure 7), which reflects brightness intensity variations across the flotation surface. Pulp temperature is monitored via an NTC thermistor (Figure 8) with nonlinearity compensated using the Beta parameter method. Liquid level is detected using an infrared reflective sensor (Figure 9) that outputs a digital HIGH or LOW signal based on surface proximity. An automatic port recognition mechanism ensures robust USB-TTL communication under varying hardware configurations, ensuring sensor reliability in complex industrial environments.

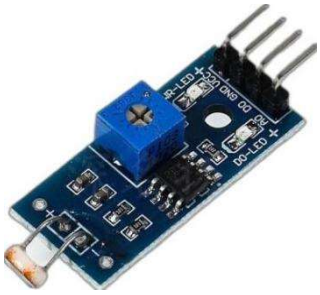


Figure 7. Light sensor module

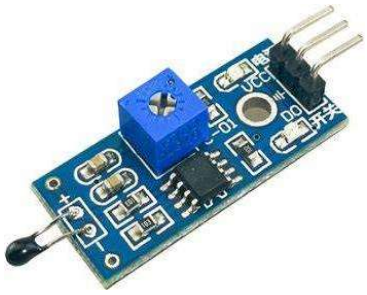


Figure 8. Temperature sensor module

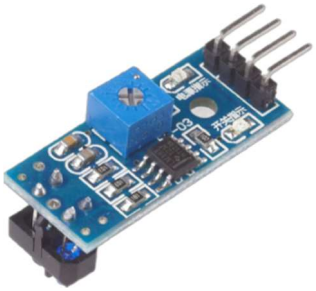


Figure 9. Infrared level sensor

2.4 EfficientNet-B5 Architecture and Feature Computation

EfficientNet-B5 is a convolutional neural network designed with a compound scaling strategy, which simultaneously adjusts network depth, width, and input resolution using a unified coefficient ϕ. The network begins with an input RGB image $I \in R^{224\times224\times3}$, which is normalized and passed through a 3×3 convolution layer to produce the initial feature map $X_0 \in R^{112\times112\times32}$.

$$X_0 = Conv_{3\times3}(I), X_0 \in R^{112\times112\times32} \quad (1)$$

where I is the input RGB image of shape 224×224×3, and X_0 is the output feature map after the initial convolution.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

The network is organized into 9 structural stages, each consisting of one or more MBConv blocks, where spatial resolution is progressively reduced while channel capacity increases. The detailed configuration of each stage-including module type, resolution, number of channels, and block repetitions-is summarized in Table 1, providing a layer-wise blueprint of EfficientNet-B5.

Table 1. EfficientNet-B5 Layer Configuration Table

Numble	Module	Resolution	Number of channels	Number of floors
1	Conv3×3	224×224	32	1
2	MBConv1, k3×3	112×112	24	3
3	MBConv6, k3×3	56×56	40	5
4	MBConv6, k5×5	28×28	64	5
5	MBConv6, k3×3	14×14	128	7
6	MBConv6, k5×5	14×14	176	7
7	MBConv6, k5×5	7×7	304	9
8	MBConv6, k3×3	7×7	512	3
9	Conv1×1 & Pooling & FC	7×7	1280	1

Each MBConv block contains a four-step structure. First, the input tensor $X \in R^{H \times W \times C}$ is passed through a 1×1 expansion convolution that increases the number of channels from C to t ·C, where t is the expansion factor (typically 6):

$$X_e = ReLU6(BN(Conv_{1\times1}(X))) \tag{2}$$

where X is the input tensor, t is the expansion ratio, and $X_e \in R^{H \times W \times tC}$ is the expanded feature.This is followed by a depthwise separable convolution applied channel-wise:

$$X_d = ReLU6(BN(DWConv_{k\times k}(X_e))) \tag{3}$$

where $DWConv_{k\times k}$ denotes a depthwise convolution with kernel size $k \in \{3,5\}$, and $X_d \in R^{H \times W \times tC}$ is the depthwise convolution output.

Next, a squeeze-and-excitation (SE) module captures channel-wise dependencies by first applying global average pooling:

$$S = \sigma(W_2 \cdot \delta(W_1 \cdot GAP(X_d))), X_s = X_d \cdot S \tag{4}$$

where GAP is global average pooling, W_1 and W_2 are weights of the SE module's two FC layers, δ is the ReLU activation, σ is the sigmoid function, $S \in R^{1\times1\times tC}$ is the channel-wise attention map, and X_s

is the recalibrated feature.

Finally, a 1×1 projection layer compresses the channel size back to the target output C′ , optionally with a residual connection if dimensions match:

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

$$X_p = BN(Conv_{1\times1}(X_s)) \tag{5}$$

where $X_p \in R^{H' \times W' \times C'}$ is the projected output.

$$Y = X + X_p \tag{6}$$

where residual addition is applied only if H=H′ ,W=W′ ,C=C′ .Taking Stage 2 as an example, five repeated MBConv6 blocks operate on inputs of shape 56×56×40, expanding to 240 channels and then projecting back to 40. This stage is responsible for extracting medium-scale structural textures such as foam contour boundaries and reflectivity gradients. After all MBConv stages, the final tensor $X_{final} \in R^{7\times7\times1280}$ is reduced via global average pooling:

$$Z = GAP(X_{final}) \in R^{1\times1\times1280} \tag{7}$$

where X_{final} is the last-stage output tensor, and Z is the pooled feature vector.

The flattened vector is then classified through a fully connected layer and softmax function:

$$\hat{y} = Softmax(W_{fc} \cdot Z + b) \tag{8}$$

where $W_{fc} \in R^{3\times1280}$ and $b \in R^3$ are the classifier weights and bias, and $\hat{y} \in R^3$ is the final probability output for the three flotation froth categories.

As illustrated in Figure 10, shallow MBConv blocks (1–3) focus on local texture features and bubble edges, while deeper blocks (6–8) capture global semantic cues and macro foam structure, enabling robust multi-scale feature integration throughout the network.

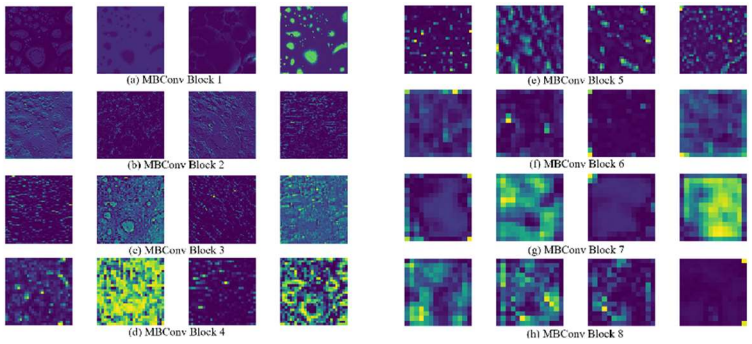


Figure 10. Architecture of the proposed edge-intelligent flotation monitoring system

2.5 Remote Control and Visualization Interface Design

To improve system operability and user interaction, a graphical user interface (GUI) was developed using the Tkinter framework. As shown in Figure 11, the interface supports real-time monitoring, sensor data visualization, and both manual and automatic control of flotation equipment. It is designed to serve as a lightweight and intuitive bridge between the data acquisition layer, AI inference engine, and actuation module.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

The GUI is logically divided into two sections: the left panel displays the camera-captured froth image and parsed sensor data, including pulp temperature, ambient light intensity, and liquid level status; the right panel contains a set of control buttons for manual operations. These include toggle commands for starting/stopping reagent dosing, pumping, and stirring, along with an emergency stop-all button.

The interface supports dual control modes: manual override and automatic execution. In manual mode, the operator can directly initiate or halt specific actions. In automatic mode, the GUI listens to the output of the EfficientNet-B5 model and executes control logic based on classification results (described in detail in Section 3.3). This modular design enables seamless coordination between perception, decision-making, and control layers within an embedded system.

Figure 11. Initial interface layout for real-time monitoring and manual control

3. Experimental Results and System Evaluation

3.1 Model Training and Validation Performance

The training and validation performance of the EfficientNet-B5 model is shown in Figure 17, which includes multiple evaluation metrics tracked across 50 training epochs. The plotted indicators include training and validation precision, recall, accuracy, and AUC, along with the dynamic learning rate schedule used during training (dashed line).

In the initial training stage (Epochs 1–20), the learning rate is relatively high (up to 5×10^{-5}), which results in significant oscillation of validation metrics-particularly val_recall, val_precision, and val_auc-due to aggressive weight updates. During this phase, the model experiences unstable generalization behavior on the validation set.

As training progresses, the learning rate is gradually decayed, and after Epoch 30, it drops below 2×10^{-5} . This lower rate leads to more stable convergence. Both validation recall and precision begin to stabilize, and accuracy curves for training and validation reach near-perfect levels. Notably, val_accuracy and val_auc consistently approach 1.0 after Epoch 35, indicating highly stable model predictions.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

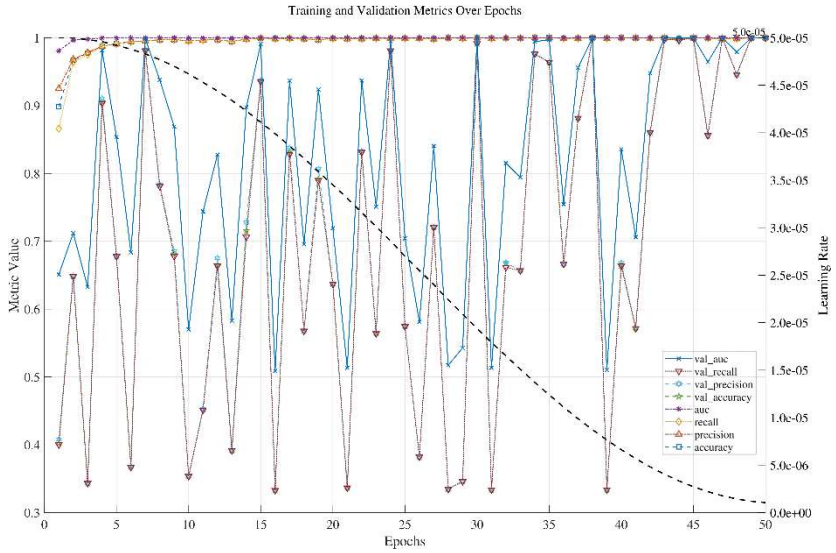


Figure 12. Training and validation metrics over epochs

The model achieves a final validation accuracy of 0.996, confirming its ability to generalize well across unseen flotation froth samples. The AUC and precision–recall metrics further validate its suitability for deployment in real-world flotation classification tasks, where high sensitivity and reliability are required.

3.2 Lightweight Inference with TensorFlow Lite

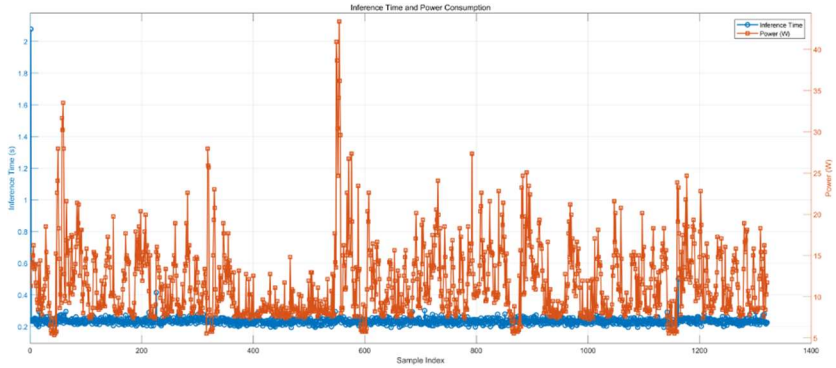


Figure 13. Inference latency and power consumption before TFLite conversion

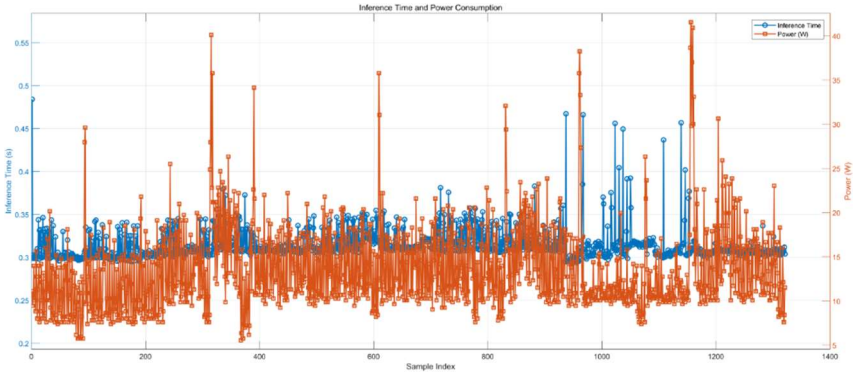


Figure 14. Inference latency and power consumption after TFLite conversion

To evaluate the real-time deployment feasibility of the proposed model, the original EfficientNet-B5 trained in Keras was converted to a TensorFlow Lite (TFLite) format and deployed on a Raspberry Pi 5. This conversion significantly reduced the model's memory and computational demands, with

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

the storage footprint dropping from 104,486 KB to 33,997 KB—an overall reduction of 67.5%. Float32 quantization was applied instead of post-training integer quantization to maintain output fidelity and avoid accuracy degradation.

Benchmark experiments were conducted to measure both inference latency and power consumption before and after TFLite optimization. As shown in Figure 13 and Figure 14, the optimized model exhibited clear improvements in runtime stability and energy efficiency. The average inference time was reduced from 0.38 s to 0.31 s, while the maximum delay dropped from 2.1 s to 0.56 s, representing an 18.4% decrease in average latency. Additionally, the inference time distribution became more concentrated, indicating improved runtime consistency.

In terms of power efficiency, the original model consumed an average of 12.6 W, with a peak of 42 W. After optimization, the TFLite model operated at a reduced average of 10.3 W and a peak of 38 W, achieving an 18.3% reduction in mean power consumption. These experimental results confirm that the TFLite-optimized model significantly enhances deployment feasibility on low-power embedded systems, enabling real-time intelligent control while lowering thermal and energy costs.

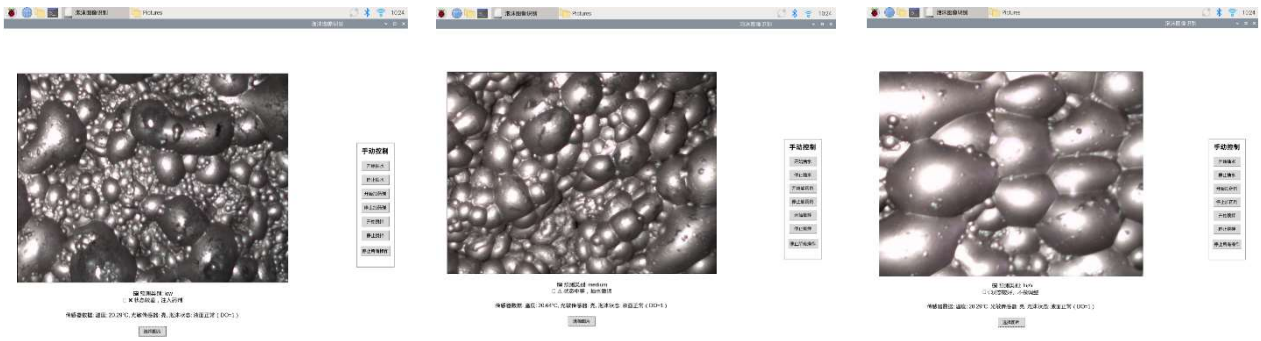
3.3 Classification-Driven Control Response Evaluation

To evaluate the integrated response mechanism of the proposed flotation monitoring system, real-time GUI behavior was tested under three distinct flotation froth states: low, medium, and high. As shown in Figure 15, the graphical interface successfully reflects the predicted category and executes the corresponding control strategy.

In the low state (Figure 15a), the froth image is dominated by sparse, small bubbles with irregular distribution, which the EfficientNet-B5 model correctly classifies as “low”. In this case, the system automatically initiates reagent dosing, as indicated by the status message and control log. The dosing action aims to enhance the hydrophobicity of mineral surfaces and improve particle-bubble attachment efficiency.

For the medium state (Figure 15b), the froth image displays a transitional texture with moderate bubble size and coverage. Upon classification, the system triggers pulp pumping to lower the slurry level and stabilize flotation dynamics. This automatic response is clearly indicated in the GUI, with updated status messages and sensor readings reflecting the real-time adjustments.

In the high state (Figure 15c), large, densely packed bubbles with stable reflectivity dominate the image, representing optimal flotation conditions. The system recognizes this as a desirable state and takes no action, allowing flotation to proceed without interference. This “pass-through logic” helps preserve reagent efficiency and avoids unnecessary actuator wear.



(a). Low state recognized (b). Medium state recognized (c). High state recognized

Figure 15. GUI responses to flotation state recognition

These experimental scenarios confirm that the GUI and control logic work cohesively with the AI inference engine to execute appropriate actions. The system demonstrates accurate classification,

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

real-time feedback, and intelligent decision-making, supporting its practical applicability in closed-loop flotation automation.

4. Conclusion

This paper presents a lightweight and edge-intelligent flotation monitoring and control system that integrates an 8051 MCU for multimodal sensor acquisition with a Raspberry Pi-based visual classification module running an EfficientNet-B5 model. The proposed system enables real-time flotation state recognition and closed-loop process control without reliance on cloud resources.

The hardware architecture was carefully optimized for embedded deployment, and the EfficientNet-B5 model was compressed using TensorFlow Lite to achieve a 67.5% reduction in model size. Experimental results on a Raspberry Pi 5 demonstrate that the optimized model achieves a validation accuracy of 0.996, an average inference latency of 310 ms, and an 18.3% reduction in power consumption, supporting efficient and stable operation under resource-constrained conditions.

A custom GUI was also developed to support real-time visualization and dual-mode control. Through experimental validation, the system accurately classified flotation froth images into low, medium, and high categories and executed appropriate dosing, pumping, or idle strategies accordingly.

In summary, the proposed system offers a scalable, energy-efficient, and field-deployable solution for intelligent flotation monitoring. It provides a feasible and practical approach for integrating deep learning-based recognition with embedded process control in mineral processing environments.

Acknowledgments

This work was supported by the Innovation and Entrepreneurship Training Program for College Students of North China University of Science and Technology under project number X2024102. The authors would like to sincerely thank Prof. Zhiqiang Wang from North China University of Science and Technology, Tangshan, China, and Prof. Yanyang Liu from Zhangjiakou University, Zhangjiakou, China, for their valuable guidance and supervision throughout this research.

References

[1] Chen L L, Han J W, Qin W Q, Liu W. Comprehensive utilization of lead-zinc smelting slag: A review[J]. Nonferrous Metals (Mineral Processing Section), 2021, (3): 49–55. DOI:10.13779/j.cnki.issn1001-0076.2021.03.007.

[2] Peng Y, Bradshaw D. Froth flotation: A century of innovation and its recent trends in machine vision and intelligent control[J]. Minerals Engineering, 2020, 146: 106139.

[3] Zhang H, Liu Q, Zhou D. Adaptive reagent control in flotation using real-time image analysis and probabilistic models[J]. Journal of Cleaner Production, 2022, 363: 132543.

[4] Yang Z, Zhang X, Wang Y, et al. Flotation froth image classification using convolutional neural networks[J]. Minerals Engineering, 2020, 157: 106530.

[5] Shi F, Xiao W, Hu Y, et al. Intelligent recognition of flotation working conditions based on froth images and deep CNNs[J]. Advanced Powder Technology, 2021, 32(3): 964–973.

[6] Liu J, Zhou L, Zhang C. Challenges in deploying deep learning for flotation control in mining environments[J]. Journal of Intelligent Manufacturing, 2022, 33(1): 157–170.

[7] Huang D, Zhao Y, Chen Y. Domain adaptation for robust froth recognition in real-world flotation scenarios[J]. Expert Systems with Applications, 2023, 213: 119034.

[8] Wang M, Sun Q, Liu Z. Lightweight CNN deployment on embedded platforms for industrial visual tasks[J]. IEEE Transactions on Industrial Informatics, 2021, 17(10): 6812–6821.

[9] Banerjee A, Panigrahi S, Kar A, et al. Edge computing in mining: Opportunities, challenges and application potential[J]. IEEE Access, 2021, 9: 104276–104288.

[10] Kumar A, Singh S, Chauhan H. Embedded deep learning for fault detection using Raspberry Pi and OpenCV[J]. Procedia Computer Science, 2020, 171: 2057–2064.

ISSN: 2414-1895 DOI: 10.6919/ICJE.202506_11(6).0046

[11] Mahmoud M, Al-Ayyoub M, Jararweh Y. Deploying CNNs on low-resource hardware: An empirical study with Raspberry Pi and Arduino platforms[J]. Future Generation Computer Systems, 2022, 128: 313–324.

[12] Tang J, Xiao T, Duan Z, et al. Towards real-time object detection on low-power devices: Quantization-aware training and model optimization[J]. IEEE Transactions on Industrial Informatics, 2021, 17(12): 8402–8413.

[13] Tan M, Le Q V. EfficientNet: Rethinking model scaling for convolutional neural networks[J]. International Conference on Machine Learning (ICML), 2020: 6105–6114.

[14] Tan M, Le Q V. EfficientNet: Improving accuracy and efficiency with a new family of models[J]. arXiv preprint, arXiv:1905.11946, 2020.

[15] Chowdhery A, Warden P, Shlens J, et al. Visual wake words dataset and efficient models for small-footprint keyword spotting[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021, 43(5): 1747–1759.

[16] Zhang Y, Wang H, Xie M. Lightweight and efficient CNN deployment on Raspberry Pi using TensorFlow Lite[J]. Embedded Systems Letters, 2022, 14(2): 56–60.

[17] Yanten C, Kracht W, Diaz G, Lois-Morales P, Egana A. Froth images from flotation laboratory test in Magotteaux cell[J]. Data, 2023, 8(4): 69. https://doi.org/10.3390/data8040069