Research Article
Digital System and Computing

Development of A Smart Piggy Bank System Based on IoT with Computer Vision Technology for Money Nominal Detection

Adelia Clarissa, **Anugrah Adiwilaga**, **Deden Pradeka**

[The author informations are in the declarations section. This article is published by ETFLIN in Digital System and Computing, Volume 2, Issue 1, 2026, Page 27-34. DOI: 10.58920/dsc0201566]

Received: 23 January 2026
Revised: 30 March 2026
Accepted: 29 June 2026
Published: 30 June 2026

Editor: Ika Novita Dewi

Keywords: Smart piggy bank, Internet of Things, Computer vision, Currency detection, YOLOv8 model, Financial literacy education.

Abstract: Saving behavior plays an important role in supporting early financial literacy and responsible financial management. However, conventional piggy banks still have limitations in transaction transparency, security, and real-time monitoring capabilities. This study aimed to develop and conduct prototype-level validation of an Internet of Things (IoT)-based smart piggy bank integrated with computer vision technology for automatic currency denomination detection. The study employed a Design and Development (D& D) approach combined with the Artificial Intelligence Life Cycle (AILC) for object detection model development and the Software Development Life Cycle (SDLC) for web-based monitoring system development. The dataset consisted of 9, 000 annotated images of Indonesian Rupiah banknotes and coins categorized into nine denomination classes under varying lighting and object orientation conditions. During training and validation, the YOLOv8 nano model achieved an mAP50 value of 0.995 under controlled conditions. However, real-world prototype testing produced an average operational accuracy of 62.2%. This performance degradation is primarily attributed to specular reflections on coin surfaces and edge-hardware-induced latency which compromised real-world inference stability. The main technical contribution of this work lies in quantifying this model-to-edge hardware performance gap, providing an empirical baseline for lightweight computer vision deployment on low-resource microcomputing nodes. These findings indicate a performance gap between controlled and real-world conditions. Overall, the findings demonstrate the feasibility of integrating lightweight computer vision and IoT technologies into an embedded smart saving prototype, although further optimization is required to improve operational robustness and detection stability in real-world environments.

Introduction

Saving behavior is an essential component of financial management that contributes to long-term financial stability and responsible economic decision-making. Early financial literacy has become increasingly important because it influences an individual's understanding of money management, budgeting, and saving habits in adulthood. In Indonesia, improving financial literacy among children remains a significant challenge, requiring more interactive and technology-adaptive educational media (1, 2).

Conventional piggy banks have long been used as simple tools to introduce saving habits and financial responsibility. Although they can support discipline and encourage saving behavior, conventional piggy banks still present several limitations, including the absence of real-time balance monitoring, automatic transaction recording, and adequate security mechanisms (3, 4).

The advancement of Internet of Things (IoT) technology has enabled the integration of communication, automation, and monitoring capabilities into various educational devices. Several previous studies have developed IoT-based smart piggy banks using RFID authentication, weight sensors, and digital monitoring systems. However, most existing systems still rely on basic sensing mechanisms and are unable to automatically identify currency denominations through computer vision approaches (5–9).

In parallel, recent developments in computer vision and deep learning have demonstrated strong performance in object detection tasks, including currency recognition applications. YOLO-based object detection models have shown considerable capability for real-time detection while maintaining computational efficiency suitable for embedded systems and edge AI deployment. Recent studies have also reported that lightweight object detection architectures can be effectively implemented on low-resource devices such as Raspberry Pi (10 – 11).

While classical architectures such as SSD-MobileNetV2 offer reduced computational footprints, they often suffer from significant accuracy loss when handling small, overlapping objects like stacked currency. Similarly, EfficientDet-Lite provides decent structural scalability but displays higher latency overhead when deployed on non-accelerated edge devices like the Raspberry Pi 4. By leveraging an anchor-free detection mechanism and an optimized feature pyramid network, YOLOv8 nano presents a compelling alternative, though its real-world edge hardware constraints demand further empirical scrutiny.

Despite these advancements, most previous studies primarily focused on software-level validation under controlled conditions and provided limited discussion regarding real-world embedded deployment performance. Furthermore, the integration of embedded AI, IoT communication, real-time monitoring, and automatic currency denomination detection within a single system remains relatively underexplored. Therefore, a research gap exists in the implementation and operational evaluation of lightweight object detection models for embedded IoT-based real-time currency recognition systems. Previous studies have rarely examined the discrepancy between model-level validation performance and real-world operational performance after deployment on embedded hardware. To address this gap, this study developed an IoT-based smart piggy bank that integrates YOLOv8 nano-based computer vision, Raspberry Pi embedded deployment, RFID authentication, web-based monitoring, and Telegram notification services within a single system architecture. The main contribution of this study lies in the integration of lightweight object detection and embedded IoT technologies into a real-time smart saving prototype that was evaluated under both controlled validation settings and real-world operational conditions.

Methodology

Study Design and Rationale

This study employed a Design and Development (D&D) research approach to design, implement, and evaluate an IoT-based smart piggy bank system integrated with computer vision technology for automatic banknote and coin denomination detection. The D&D approach was selected because it supports systematic product-oriented research focusing on the development, testing, and refinement of functional systems. In this study, the research followed the Product and Tool Research cluster, which emphasizes the creation and validation of technological artifacts.

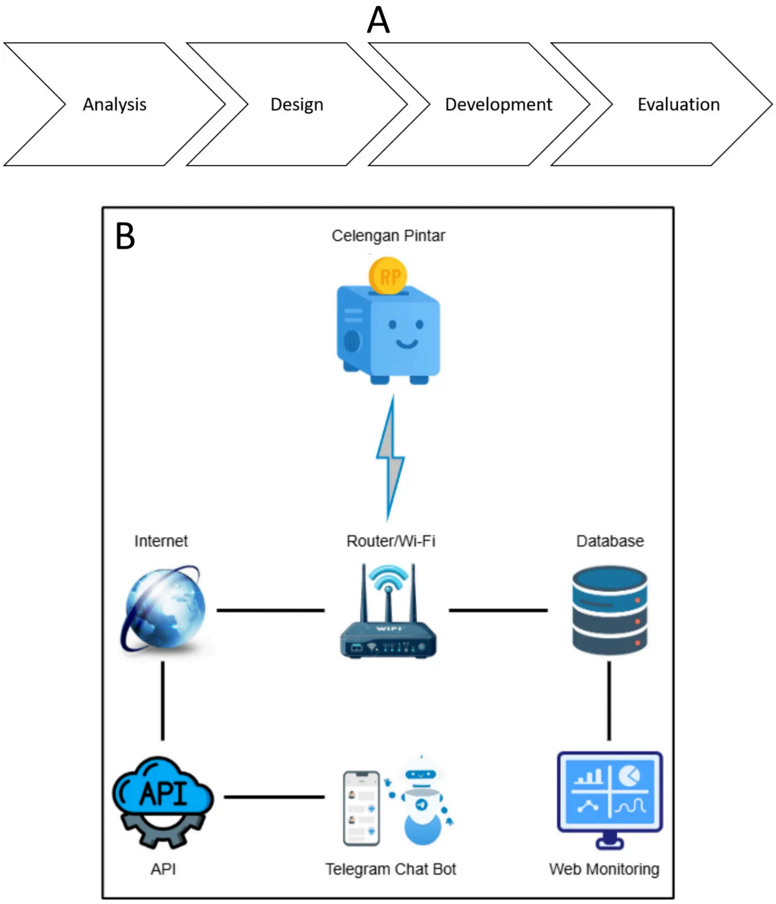


Figure 1. Development and design (D&D) framework for a Smart Piggy Bank System based on IoT: (A) phases flow diagram depicting the iterative development stages and (B) system architecture diagram illustrating the integration of hardware components and cloud-based data management.

The development process was structured into four operational phases: Analysis, Design, Development, and Evaluation. An overview of the research workflow and phase sequence is illustrated in **Figure 1A.** This phased approach was considered appropriate due to the integration of hardware, artificial intelligence models, and web-based software components within a single system. The primary objective was to ensure that the resulting smart piggy bank not only functioned correctly but also provided reliable monitoring, accurate money detection, and practical usability for children and parents. The Artificial Intelligence Life Cycle (AILC) framework was used for computer vision model development, while the Software Development Life Cycle (SDLC) framework was applied to the development of the web-based monitoring system. These frameworks were selected because they support systematic and iterative development processes for embedded artificial intelligence and IoT systems (11, 12).

Tools and Techniques

The system architecture consisted of several integrated components, including a Raspberry Pi as the embedded processing unit, ESP32 for IoT communication, a camera module for image acquisition, an RFID module for user authentication, an LCD display, a web-based monitoring platform, and Telegram notification services. The Raspberry Pi performed image processing and real-time object detection inference, while ESP32 handled communication between hardware components and the monitoring system. The overall system architecture and interaction between components are illustrated in **Figure 1B.**

Sample and Selection Criteria

The dataset used in this study consisted of 9,000 images of Indonesian Rupiah banknotes and coins categorized into nine denomination classes, including Rp500 coins, Rp1,000 coins, Rp1,000 banknotes, Rp2,000, Rp5,000, Rp10,000, Rp20,000, Rp50,000, and Rp100,000 banknotes (see **Figure 2**). Image acquisition was conducted using smartphone cameras and Raspberry Pi camera modules under various environmental conditions to improve dataset diversity and model robustness. Data collection variations included differences in lighting intensity, camera distance, object orientation, viewing angle, and background conditions. All images were annotated using Roboflow and manually verified to maintain annotation consistency and minimize labeling errors.

The dataset was divided into training, validation, and

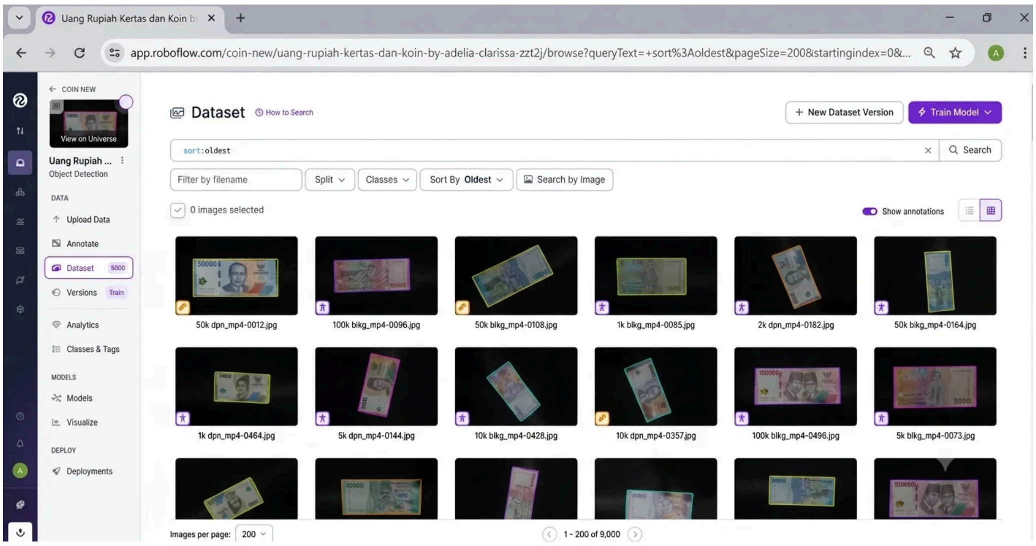
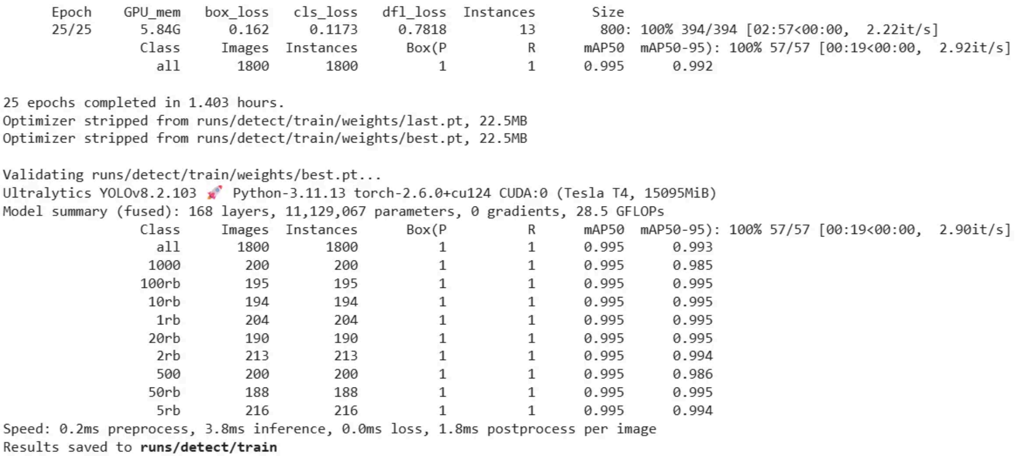


Figure 2. Dataset preparation and annotation process using Roboflow Universe.



```
      Epoch    GPU_mem   box_loss   cls_loss   dfl_loss  Instances       Size
      25/25      5.84G      0.162     0.1173     0.7818         13        800: 100% 394/394 [02:57<00:00,  2.22it/s]
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 57/57 [00:19<00:00,  2.92it/s]
                   all       1800       1800          1          1      0.995      0.992

25 epochs completed in 1.403 hours.
Optimizer stripped from runs/detect/train/weights/last.pt, 22.5MB
Optimizer stripped from runs/detect/train/weights/best.pt, 22.5MB

Validating runs/detect/train/weights/best.pt...
Ultralytics YOLOv8.2.103 🚀 Python-3.11.13 torch-2.6.0+cu124 CUDA:0 (Tesla T4, 15095MiB)
Model summary (fused): 168 layers, 11,129,067 parameters, 0 gradients, 28.5 GFLOPs
                 Class     Images  Instances      Box(P          R      mAP50  mAP50-95): 100% 57/57 [00:19<00:00,  2.90it/s]
                   all       1800       1800          1          1      0.995      0.993
                  1000        200        200          1          1      0.995      0.985
                 100rb        195        195          1          1      0.995      0.995
                  10rb        194        194          1          1      0.995      0.995
                   1rb        204        204          1          1      0.995      0.995
                  20rb        190        190          1          1      0.995      0.995
                   2rb        213        213          1          1      0.995      0.994
                   500        200        200          1          1      0.995      0.986
                  50rb        188        188          1          1      0.995      0.995
                   5rb        216        216          1          1      0.995      0.994
Speed: 0.2ms preprocess, 3.8ms inference, 0.0ms loss, 1.8ms postprocess per image
Results saved to runs/detect/train
```

Figure 3. YOLOv8 nano training workflow using Google Colaboratory.

testing subsets using a 70:20:10 ratio. The currency detection model was developed using YOLOv8 nano because of its lightweight architecture, lower computational requirements, faster inference capability, and suitability for deployment on resource-constrained embedded devices such as Raspberry Pi. Compared with larger object detection architectures, YOLOv8 nano provides a better balance between computational efficiency and detection performance for edge AI applications (13 – 18).

Model training was conducted using Google Colaboratory with GPU acceleration support. The training configuration included 100 epochs, a batch size of 16, and a learning rate of 0.001. Data augmentation techniques, including image rotation, brightness adjustment, scaling, and horizontal flipping, were applied to improve model generalization under varying operational conditions. The overall training workflow, including data loading, model configuration, and training execution on Google Colab, is illustrated in **Figure 3**.

Several deployment constraints were identified during embedded implementation. Raspberry Pi hardware limitations affected computational performance, memory utilization, and inference speed during continuous operation. In addition, unstable lighting conditions, reflective coin surfaces, and tilted object orientations contributed to reduced detection consistency during real-world testing.

The evaluation process was divided into two stages to distinguish model validation performance under controlled conditions from operational system performance in real-world environments. Controlled model evaluation used Precision, Recall, F1-score, mAP50, and mAP50-95 metrics. Meanwhile, operational prototype evaluation included detection accuracy, transaction consistency, response time, and communication stability under varying environmental conditions. This evaluation separation was necessary to analyze the performance gap between controlled validation environments and real-world operational deployment.

System testing was performed under varying operational conditions, including differences in lighting intensity, camera angle, object position, and object orientation. These scenarios were designed to evaluate the robustness and feasibility of the developed embedded AI system under real operational environments.

To evaluate operational accuracy, structured testing was executed across 100 discrete deposition trials for each of the 9 denomination classes (N = 900 total trials). Hardware performance metrics were continuously profiled using the Linux psutil library. The embedded system's resource consumption was benchmarked under continuous inference loads, measuring average inference latency (ms), frame rates (FPS), CPU utilization (%), and memory allocation (MB) to gauge edge efficiency.

Results and Discussion

The training and validation results showed that the YOLOv8 nano model achieved high detection performance under controlled dataset conditions. Precision, recall, and mAP values indicated that the model was capable of identifying most currency denominations with relatively low classification errors during internal evaluation. The overall detection performance of the YOLOv8-based model, including accuracy and robustness across different denominations, is visualized in **Table 2.**

The average mAP50 value reached 0.995 during the validation stage, indicating strong model performance within controlled evaluation environments. However, operational testing on the embedded prototype produced an average system accuracy of 62.2%, revealing a substantial difference between model validation performance and real-world operational performance (see **Figure 4**). These findings indicate a performance gap between controlled and real-world conditions. During validation, the model was evaluated using relatively consistent datasets in terms of lighting, image quality, and

Table 1. YOLOv8 nano training configuration.

Parameter	Value
Model	YOLOv8 nano
Epoch	100
Batch size	16
Learning rate	0.001
Image size	640 × 640
Optimizer	SGD
Dataset split ratio	70:20:10

Table 2. Performance evaluation results of the YOLOv8-based detection model.

Evaluation	Class	100rb	50rb	20rb	10rb	5rb	2rb	1rb	1000	500
Precision	1	1	1	1	1	1	1	1	1	1
Recall	1	1	1	1	1	1	1	1	1	1
mAP50	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995	0.995
mAP50-95	0.993	0.995	0.995	0.995	0.995	0.994	0.994	0.995	0.984	0.986

object orientation. In contrast, real-world operational testing introduced environmental variations that directly affected detection stability.

Operational testing demonstrated that environmental conditions significantly influenced detection accuracy. The best detection stability was achieved under adequate indoor lighting with frontal object positioning and relatively consistent camera distances. Under these conditions, banknotes such as Rp5, 000 and Rp10, 000 showed more stable detection performance because of their clearer visual characteristics.

In contrast, the largest performance degradation occurred under low-light conditions, tilted object positions, overlapping objects, and reflective coin surfaces. Rp500 and Rp1, 000 coins exhibited higher misclassification rates because of visual similarities and light reflection effects on metallic surfaces. These findings indicate that environmental variability and embedded hardware limitations significantly affected inference stability during real-world deployment. The results also demonstrate that model evaluation metrics such as mAP and precision do not always fully represent the operational performance of embedded AI systems.

To further analyze the classification behavior of the model, a confusion matrix was generated to visualize the distribution of correct and incorrect predictions across all denominations. The confusion matrix of the currency detection results can be seen in **Figure 5**.

In addition to environmental factors, Raspberry Pi computational limitations also influenced inference consistency during repeated testing. Continuous inference cycles under constrained hardware environments compounded processing latency. Specifically, during low-light scenarios, the frame rate dropped from 7.5 FPS to 4.2

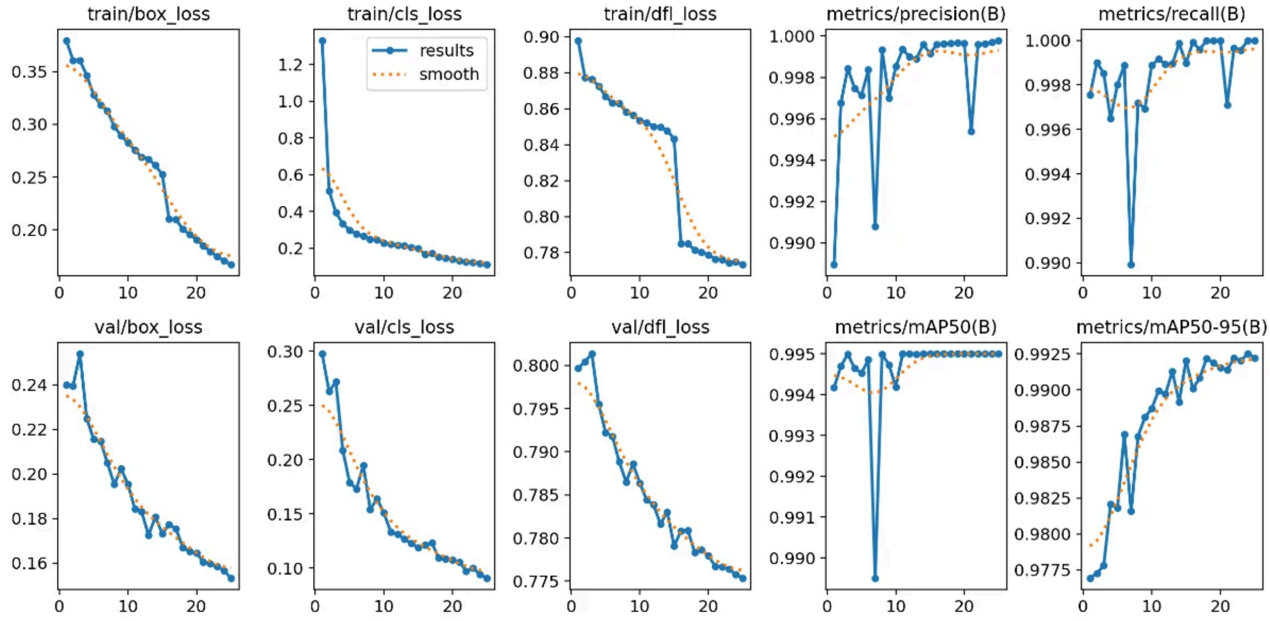


Figure 4. Training and validation performance graphs.

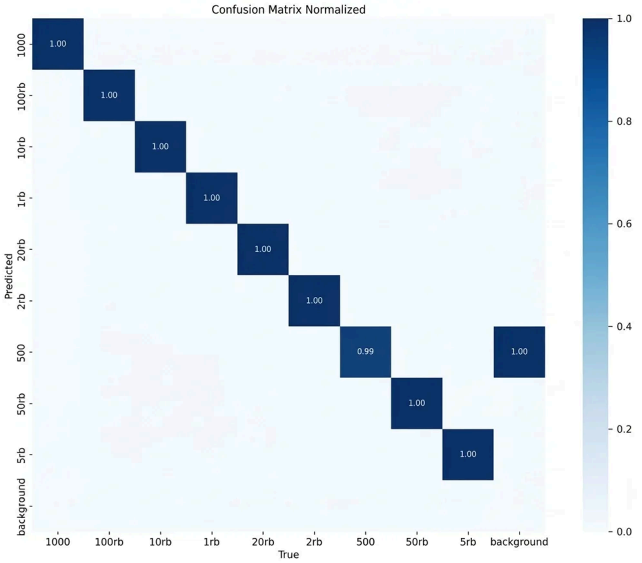


Figure 5. Confusion matrix of currency denomination detection results.

FPS due to the increased computational overhead of handling high-specularity and poorly contrasted frames.

The embedded system successfully integrated RFID authentication, currency denomination detection, transaction recording, balance monitoring, and Telegram notifications. Transaction data were successfully transmitted to the web-based monitoring platform in real time without significant communication interruptions. However, the testing results showed that communication reliability did not always correspond directly to currency detection stability. In several operational scenarios, the system was still able to transmit transaction data despite misclassification of currency denominations. These findings indicate that IoT communication reliability and computer vision detection accuracy represent two distinct evaluation aspects that should be analyzed separately.

The physical realization of the smart piggy bank prototype and component integration can be seen in **Figure 6A**. Integration was performed incrementally to ensure stable communication between hardware, AI processing, and the web monitoring system. The website was implemented and connected to the IoT system to display savings data, transaction history, and notifications in real time. The monitoring interface used for system validation is shown in **Figure 6B** and the example of notification on Telegram can be seen in **Figure 6C**. The evaluation phase involved functional testing, accuracy testing of money detection, system response time measurements, and end-to-end validation of data transmission from the smart piggy bank to the monitoring interface.

Previous studies on IoT-based smart piggy banks mainly focused on balance monitoring, RFID authentication, or sensor-based mechanisms without integrating computer vision for automatic currency denomination recognition. In contrast, this study integrated lightweight YOLOv8 nano object detection within an embedded operational environment (8, 9). Several previous embedded object detection studies also reported that lightweight AI models experience performance degradation when deployed on edge devices with limited computational resources and varying environmental conditions. Similar trends were observed in this study, particularly under unstable lighting conditions and reflective object surfaces (15–19).

Therefore, the findings of this study not only demonstrate system integration feasibility but also highlight operational challenges associated with embedded AI deployment in real-world environments. These results emphasize the importance of evaluating prototype systems under operational conditions rather than relying solely on dataset validation metrics.

Limitations

This study has several limitations that may affect operational system performance. First, the evaluation was limited to prototype-level validation and did not include long-term deployment or large-scale multi-user testing. Second, environmental variations such as lighting intensity, object orientation, and reflective surfaces significantly influenced detection consistency.

Third, deployment using Raspberry Pi introduced computational constraints that affected inference speed and operational stability compared with desktop-based environments. In addition, this study did not conduct quantitative comparisons with alternative object detection architectures such as SSD, Faster R-CNN, or transformer-based models because of hardware and development time limitations.

These limitations indicate that the findings remain at the prototype validation stage and cannot yet be generalized to broader operational deployment scenarios. Future studies are recommended to expand dataset diversity, optimize embedded deployment performance, compare multiple object detection architectures, and evaluate usability under larger-scale operational environments.

Conclusion

This study successfully developed and conducted prototype-level validation of an Internet of Things (IoT)-based smart piggy bank integrated with computer vision technology for automatic currency denomination detection. The developed prototype integrated YOLOv8 nano object detection, Raspberry Pi embedded deployment, RFID authentication, web-based monitoring, and Telegram notification services within a single operational system.

The evaluation results showed that the YOLOv8 nano model achieved high performance during training and validation stages, with an mAP50 value of 0.995 under controlled conditions. However, real-world operational testing produced an average system accuracy of 62.2% due to lighting variability, object positioning, reflective

Daftar Tabungan

RFID	Nama	Kelas	Tanggal	Jam	Setoran	Saldo

❌ Akses Ditolak: Kartu Tidak Terdaftar 15:48

❌ Akses Ditolak: Kartu Tidak Terdaftar 15:48

💰 Setoran Uang:
🆔 ID: 2CAB1305
💵 Nominal: Rp0
📅 Waktu: 2025-08-27T15:49:17
📦 Total: Rp0 15:49

💰 Setoran Uang:
🆔 ID: 2CAB1305
💵 Nominal: Rp20000
📅 Waktu: 2025-08-27T15:49:32
📦 Total: Rp20000 15:49

✅ Akses Diterima: Doorlock Terbuka 15:49

August 28

⚠️ Admin Mereset Saldo Tabungan 21:53

Figure 6. System implementation and user interface: (A) hardware prototype of the smart piggy bank, (B) website monitoring interface for real-time data visualization, and (C) Telegram notification output for automated saving transaction updates.

surfaces, and embedded hardware limitations. These findings indicate a performance gap between controlled and real-world conditions, demonstrating that model validation metrics alone are insufficient to fully represent operational embedded system performance.

The findings of this study primarily support prototype-level validation rather than generalized real-world applicability. Although the developed system was capable of performing real-time currency detection, transaction recording, and IoT-based monitoring, the operational performance remained influenced by environmental variability and embedded hardware constraints. Therefore, the conclusions of this study are limited to demonstrating the technical feasibility of integrating lightweight computer vision and IoT technologies within an embedded smart saving prototype.

In addition, the study does not provide empirical evidence regarding long-term educational impact or large-scale deployment effectiveness. Consequently, claims related to educational outcomes or broader implementation benefits should be interpreted as future potential rather than experimentally validated outcomes. Further optimization, scalability testing, and broader operational evaluation are required to improve detection robustness, inference efficiency, and real-world deployment reliability.

Declaration

Author Information

Adelia Clarissa
*Corresponding author
Department of Computer Engineering, Universitas Pendidikan Indonesia, Bandung, Indonesia.
Contribution: Data Curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Writing - Original Draft, Writing - Review & Editing.

Anugrah Adiwilaga
Department of Computer Engineering, Universitas Pendidikan Indonesia, Bandung, Indonesia.
Contribution: Conceptualization, Methodology, Supervision.

Deden Pradeka
Department of Computer Engineering, Universitas Pendidikan Indonesia, Bandung, Indonesia.
Contribution: Software, Supervision, Visualization.

Conflict of Interest

The authors declare no conflicting interest.

Data Availability

The data generated during and/or analyzed during the current study are available from the corresponding author on reasonable request.

Ethics Statement

Ethical approval was not required for this study.

Funding Information

The authors declare that no financial support was received for the research, authorship, and/or publication of this article.

References

1. Iradianty A, Azizah PZ. Analisis Pengaruh Literasi Keuangan, Sosialisasi Keuangan Keluarga, dan Uang Saku Terhadap Minat Menabung pada Siswa Usia Remaja Kota Yogyakarta. Jobs. 2023;4(1):13-22. doi: https://doi.org/10.33476/jobs.v4i1.3600

2. Kusumahadi TA, Utami N. Teknologi Finansial, Literasi Keuangan, dan Keputusan Investasi Produk Reksa Dana di Indonesia. Jmm. 2022;21(2):177-186. doi: https://doi.org/10.28932/jmm.v21i2.4631

3. Munir S. Meningkatkan budaya menabung remaja usia sekolah melalui literasi keuangan dan aplikasi menabung. IComSE. 2024;5(1):526-532. doi: https://doi.org/10.34010/icomse.v5i1.14215

4. Khorida E. Implementasi Pembelajaran Sosial Pembiasaan Diri Bershodaqoh di Sekolah dalam Membentuk Kepribadian Anak (Studi Kasus di MI Perguruan Muâ€™allimat Cukir Jombang). Jurnal Modeling. 2016;3(2):187-204. doi: https://doi.org/10.69896/modeling.v3i2.81

5. Hadiana H, Syurmita S, Rahman ZD. Peningkatan Literasi Keuangan dan Perbankan Pada Generasi Alpha. psn. 2024;3(1):100. doi: https://doi.org/10.36722/psn.v3i1.2502

6. Ahmad Wahyudin, Faisol Faisol, Zulviar Anas. Literasi Keuangan Untuk Generasi Muda Milenial Bagi Siswa SMKN 2 Sampang. Karya. 2024;1(3):165-172. doi: https://doi.org/10.62951/karya.v1i3.637

7. Haniko P, Sappaile BI, Gani IP, Sitopu JW, Junaidi A, Sofyan, et al. Menjembatani Kesenjangan Digital: Memberikan Akses ke Teknologi, Pelatihan, Dukungan, dan Peluang untuk Inklusi Digital. Jpws. 2023;2(05):306-315. doi: https://doi.org/10.58812/jpws.v2i5.371

8. Ulum MB, Ayuni SD, Uddin J, Falah AH. Bangun sistem celengan pintar pengendali jaringan listrik berbasis mikrokontroler. Jitet. 2025;13(1). doi: https://doi.org/10.23960/jitet.v13i1.5611

9. Komarudin RM. Rancang bangun celengan pintar berbasis IoT menggunakan RFID dan notifikasi Telegram [skripsi]. Bandung: Universitas Pendidikan Indonesia; 2024.

10. Nur Hidayat AM, Antamil A, Zakiyah M I. Identifikasi Nominal Mata Uang Rupiah Bagi Penyandang Tunanetra Dengan Algoritma Convolutional Neural Network Berbasis Android. shift. 2023;3(2):60-65. doi: https://doi.org/10.24252/shift.v3i2.102

11. De Silva D, Alahakoon D. An artificial intelligence life cycle: From conception to production. Patterns. 2022;3(6):100489. doi: https://doi.org/10.1016/j.patter.2022.100489

12. Chua C, Storey V. A Tutorial on Prototyping Internet of

Things Devices and Systems: A Gentle Introduction to Technology that Shapes Our Lives. Cais. 2022;51(1):327-368. doi: https://doi.org/10.17705/1cais.05115

13. Wang CY, Bochkovskiy A, Liao HYM. YOLOv7: Trainable Bag-of-Freebies Sets New State-of-the-Art for Real-Time Object Detectors. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 2023:7464-7475. doi: https://doi.org/10.1109/cvpr52729.2023.00721

14. Jocher G, Chaurasia A, Qiu J. YOLO by Ultralytics [Internet]. GitHub; 2023 [cited 2026 May 13]. Available from: https://github.com/ultralytics/ultralytics

15. Meribout M, Baobaid A, Khaoua MO, Tiwari VK, Pena JP. State of Art IoT and Edge Embedded Systems for Real-Time Machine Vision Applications. IEEE Access. 2022;10:58287-58301. doi: https://doi.org/10.1109/access.2022.3175496

16. Nguyen TK, Pham TK, Doan ST. Real-Time Flying Object Detection and Counting Using Deep Learning. 2024 RIVF International Conference on Computing and Communication Technologies (RIVF). 2024:382-387. doi: https://doi.org/10.1109/rivf64335.2024.11009026

17. Kim J, Lee S, Park H. Embedded deep learning-based currency recognition system for assistive financial applications. Expert Syst Appl. 2022;198:116763.

18. Goyal V, Singh R, Kumar A, Dhawley M, Sharma S. Aerial Object Detection Using Different Models of YOLO Architecture: A Comparative Study. Singapore: Springer Nature Singapore; 2022. doi: https://doi.org/10.1007/978-981-19-2126-1_27

19. Khelifi C. Design and Deployment of Lightweight Edge AI Models for IoT Network Intrusion Detection. Techrxiv. 2025. doi: https://doi.org/10.36227/techrxiv.176184835.56908681/v1

20. Padilla R, Netto SL, da Silva EAB. A Survey on Performance Metrics for Object-Detection Algorithms. 2020 International Conference on Systems, Signals and Image Processing (IWSSIP). 2020:237-242. doi: https://doi.org/10.1109/iwssip48289.2020.9145130

21. Parab C. Lightweight AI Models for Real-Time Object Detection on Edge Devices. SSRN. 2025. doi: https://doi.org/10.2139/ssrn.5424294

22. Li L, Qin S, Yang N, Hong L, Dai Y, Wang Z. LVNet: A lightweight volumetric convolutional neural network for real-time and high-performance recognition of 3D objects. Multimed Tools Appl. 2024;83(21):61047-61063. doi: https://doi.org/10.1007/s11042-023-17816-2

23. Han T, Dong Q, Sun L. SenseLite: A YOLO-Based Lightweight Model for Small Object Detection in Aerial Imagery. Sensors. 2023;23(19):8118. doi: https://doi.org/10.3390/s23198118

24. Şimşek F. Real-Time Object Detection with QuantizedYOLOv11 and YOLOv8 on Raspberry Pi 5 for Low-Speed ADAS. Preprint. 2026. doi: https://doi.org/10.21203/rs.3.rs-8584571/v1

25. Nurul Achmadiah M, Ahamad A, Sun CC, Kuo WK. Energy-Efficient Fast Object Detection on Edge Devices for IoT Systems. IEEE Internet Things J. 2025;12(11):16681-16694. doi: https://doi.org/10.1109/jiot.2025.3536526

Additional Information

How to Cite

APA 7th Edition: Clarissa, A., Adiwilaga, A. & Pradeka, D. (2026). Development of A Smart Piggy Bank System Based on IoT with Computer Vision Technology for Money Nominal Detection. Digital System and Computing, 2(1), 27-34. https://doi.org/10.58920/dsc0201566

Vancouver: Clarissa A, Adiwilaga A, Pradeka D. Development of A Smart Piggy Bank System Based on IoT with Computer Vision Technology for Money Nominal Detection. Digital System and Computing. 2026;2(1):27-34. https://doi.org/10.58920/dsc0201566

Harvard: Clarissa, A., Adiwilaga, A. & Pradeka, D. (2026) 'Development of A Smart Piggy Bank System Based on IoT with Computer Vision Technology for Money Nominal Detection', Digital System and Computing, 2(1), pp. 27-34. doi: 10.58920/dsc0201566

Publisher Note

Open Access