POLITEKNIK NEGERI BALI

matriX

JURNAL MANAJEMEN TEKNOLOGI DAN INFORMATIKA

Vol. 16 No. 2 | Page 62-128 | July 2026 | ISSN 2580-5630

Editors

Reviewers

PREFACE

We would like to present, with great pleasure, the second issue of Matrix: Jurnal Manajemen Teknologi dan Informatika in Volume 16, 2026. This journal is under the management of Scientific Publication, Research and Community Service Center, Politeknik Negeri Bali, and is devoted to covering the field of technology and informatics management including managing the rapid changes in information technology, emerging advances in electrical and electronics and new applications, implications of digital convergence and growth of electronics technology, and project management in electrical. The scientific articles published in this edition were written by researchers from Universitas Pendidikan Ganesha, STMIK Bandung Bali, Universitas Amikom Yogyakarta, Politeknik Prestasi Prima, Universitas Ahmad Dahlan, Universitas Pamulang, and Politeknik Negeri Bali. Articles in this issue cover topics in the field of Development of an AI IoT-based smart system to support improved beach visitor safety, Video-based human activity recognition analysis using YOLOv26, Daily USD-IDR exchange rate prediction using Long Short-Term Memory (LSTM) with macroeconomic indicators and recursive multi-step prediction, Image classification of rerajahan ulap-ulap Bali using MobileNetV2 architecture and Multi-class skin disease classification using transfer learning architectures. Finally, we would like to thank the reviewers for their efforts and hard work in conducting a series of review phases thoroughly based on their expertise. We hope that the work of the authors in this issue will be a valuable resource for other researchers and will stimulate further research into the vibrant area of technology and information management in specific, and engineering in general.

Politeknik Negeri Bali, 31 July 2026

Editor-in-chief

Dewa Ayu Indah Cahya Dewi, S.TI., M.T.

ISSN: 2580-5630

Table of contents

Matrix: Jurnal Manajemen Teknologi dan Informatika
Volume 16 Issue 2 Year 2026 Pages 62-73
DOI: http://dx.doi.org/10.31940/matrix.v16i2.62-73
URL: http://ojs2.pnb.ac.id/index.php/matrix

Development of an AI IoT-based smart system to support improved beach visitor safety

Kusrini [1], Andriyan Dwi Putra [2]*, Bayu Setiaji [1], Eko Pramono [3], Elik Hari Muktafin [4]

[1] *Informatics Study Program, Universitas Amikom Yogyakarta, Indonesia*
[2] *Information Systems Study Program, Universitas Amikom Yogyakarta, Indonesia*
[3] *Computer Engineering, Universitas Amikom Yogyakarta, Indonesia*
[4] *Software Engineering Application Study Program, Politeknik Prestasi Prima, Indonesia*

*Corresponding Author: *andriyan@amikom.ac.id*

Abstract: The high rate of sea accidents at Indonesian coastal tourist destinations, such as those occurring at Parangtritis and Pangandaran beaches, underscores the limitations of conventional safety systems that rely on manual surveillance. The objective of this research is to develop the Smart Beach Visitor Safety System (Smart BEVISAT), an integrated system based on artificial intelligence (AI) and the Internet of Things (IoT) to enhance beachgoer safety through preventive and reactive functions. The system development method employs the waterfall model, which includes stages of requirements analysis, design, implementation, and testing. The system is composed of three primary components: an AI surveillance camera, an IoT safety buoy, and a ground station. The AI camera utilizes advanced algorithms such as YOLOv11 for object detection, ByteTrack for tracking, and DeepLabv3+ for waterline segmentation, enabling it to monitor visitors and detect safety zone violations in real-time. In the event of an incident, the system automatically sends the victim's coordinates to the IoT Safety Buoy, an Unmanned Surface Vehicle (USV) designed to autonomously navigate to the victim's location for rapid evacuation. The implementation results show a functional prototype with measurable technical specifications, where the buoy achieves a total gross buoyancy of 205.8 kg, a net carrying capacity of approximately 190 kg, and a measured operational speed of 1.20–1.53 m/s under varying load conditions, demonstrating effective rescue capability. The development of Smart BEVISAT is expected to provide a technological solution to minimize accident risks and strengthen the maritime tourism sector in Indonesia.

Keywords: Artificial Intelligence, Beach Safety, Internet of Things (IoT), Line Object Detection, Unmanned Surface Vehicle (USV).

History Article: Submitted 25 February 2026 | Revised 06 May 2026 | Accepted 21 May 2026

How to Cite: K. Kusrini, A. D. Putra, B. Setiaji, E. Pramono, and E. H. Muktafin, "*Development of an AI IoT-based smart system to support improved beach visitor safety,*" *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 16, no. 2, pp. 62–73, 2026, doi: 10.31940/matrix.v16i2.62-73.

Introduction

Indonesia is known as a maritime country with the longest coastline in the world, stretching approximately 108,000 km [1]. This geographical condition makes the sea and coastline an important part of the social, economic and cultural life of the community. According to data from the Ministry of Maritime Affairs and Fisheries (KKP), Indonesia's maritime territory covers an area of approximately 290,000 km2 as sovereign territory, plus 2.55 million km2 of Exclusive Economic Zone (EEZ) waters [2].

This great potential is accompanied by various challenges, one of which is the high risk of marine and coastal accidents, such as the drowning of tourists and fishermen. Studies show that the effectiveness of coastal safety warning signs installed to inform beach visitors about the potential dangers on the beach as a whole is not always effective [3]. For example, on January 28, 2025, 13 students from Mojokerto were swept away by waves at Drini Beach, Gunung Kidul. Of these, 9 were rescued, 3 died, and 1 is still missing [4]. This case shows that manual warning systems are not yet able to provide optimal protection for beach visitors.

Technological developments, particularly Artificial Intelligence (AI) and the Internet of Things (IoT), have opened up new opportunities to improve safety in maritime areas and beach

tourism. In this case, one example of its application is CCTV (Closed-Circuit Television) cameras that monitor visitor activities through surveillance cameras in real-time [5]. However, conventional CCTV systems are still highly dependent on human operators for monitoring, which can potentially cause delays in detection when incidents occur [6].

With technological advances, CCTV can be upgraded through the application of Computer Vision. With the support of AI and IoT, it provides comprehensive safety solutions with two main functions: preventive to prevent accidents and reactive to speed up evacuation [7]. Surveillance cameras not only record, but are also capable of detecting objects, tracking movements, and recognizing certain patterns automatically [8]. When combined with IoT, this system can connect to other devices, for example to activate warning alarms when visitors cross safe boundaries or send survivor location coordinates to automatic rescue devices [9].

To address these challenges, this study proposes the development of a Smart Beach Visitor Safety System (Smart BEVISAT). This is an integrated safety system based on Artificial Intelligence (AI) and the Internet of Things (IoT). Smart BEVISAT uses three main components, namely AI surveillance cameras, IoT safety buoys, and ground stations. The AI surveillance camera detects and tracks visitor movements in real-time and automatically sounds an alarm when someone crosses a predetermined virtual safety line. Meanwhile, the reactive function is implemented with IoT safety buoys. A hydrodynamic Unmanned Surface Vehicle (USV) can move autonomously (autopilot) to the coordinates of the victim detected by the AI system, thereby speeding up the rescue process. The entire system is supported by an independent ground station equipped with solar panels and wind turbines, ensuring 24-hour operation and wireless connectivity for remote monitoring.

Several studies have addressed water safety using AI and IoT technologies. Alotaibi [10] developed a ResNet50-based transfer learning system for swimming pool monitoring that achieved up to 99% detection accuracy. Domingo [11] demonstrated that deep learning models can predict beach attendance levels with a mean absolute error of approximately 0.03. Campo et al. [12] proposed a LoRa-based IoT wristband platform for beach resort monitoring that improved lifeguard situational awareness. However, these studies share critical limitations that leave an important gap unaddressed. First, none of them integrates both a preventive detection function and a reactive autonomous rescue function within a single unified system. The swimming pool system [10] and the beach attendance predictor [11] are purely preventive and provide no mechanism for active rescue deployment. The wristband platform [12] requires visitors to wear dedicated hardware and is not scalable to large open-beach environments. Second, none of these works addresses the unique challenges of open-sea coastal environments in developing countries, such as limited infrastructure, dynamic rip currents, and the absence of reliable power grids. Third, no existing study has proposed an energy-autonomous ground station capable of sustaining 24-hour surveillance and charging rescue devices simultaneously. This research directly addresses these gaps by developing Smart BEVISAT, the first integrated beach safety system that combines real-time AI surveillance with an autonomous IoT rescue buoy, all supported by a self-powered ground station.

The main contributions of this paper are as follows. (1) A unified preventive-reactive beach safety architecture: Smart BEVISAT is the first system to integrate AI-based safety-line crossing detection (preventive) with autonomous GPS-guided buoy deployment (reactive) in a single operational pipeline. (2) A real-time multi-algorithm AI camera subsystem: the system combines YOLOv11 for object detection, ByteTrack for multi-object tracking, OpenCV Perspective Transformation for coordinate mapping, and DeepLabv3+ for dynamic waterline segmentation, enabling accurate zone violation detection under varying coastal conditions. (3) An energy-autonomous ground station: the ground station integrates solar panels and wind turbines to operate continuously for 24 hours without external power, making the system deployable in remote coastal areas with no grid infrastructure. (4) A hydrodynamic catamaran rescue buoy: the IoT safety buoy is designed with a catamaran hull for stability in ocean waves, capable of carrying victims weighing up to 80 kg while navigating autonomously to GPS target coordinates.

The remainder of this paper is organized as follows. Section 2 describes the system development methodology, including the research design, component specifications, and experimental testing protocol. Section 3 presents the implementation results and performance measurements for each subsystem, along with a comparative discussion against related work. Section 4 concludes the paper and outlines directions for future research.

Methodology

This research follows the Waterfall development model to guide the system engineering process, covering the stages of requirements analysis, system design, implementation, and testing. The Waterfall model was adopted because the core hardware and software requirements namely the three-component architecture (AI surveillance camera, IoT safety buoy, and ground station) were established in advance through prior prototyping work on the electric safety buoy [13]. The Waterfall model remains well suited to projects with stable, clearly defined requirements and strong demands for documentation and traceability [14]. This approach is also consistent with recent digital engineering methodologies for safety-critical AI systems, which emphasize systematic design, verification, validation, and lifecycle traceability to ensure the reliability and safety of integrated AI-enabled cyber-physical systems [15]. It is important to note that the Waterfall model describes the system-level engineering workflow, while the AI model development itself follows a standard iterative machine learning pipeline involving dataset curation, model training, validation, and fine-tuning. The overall research design is illustrated in Figure 1.

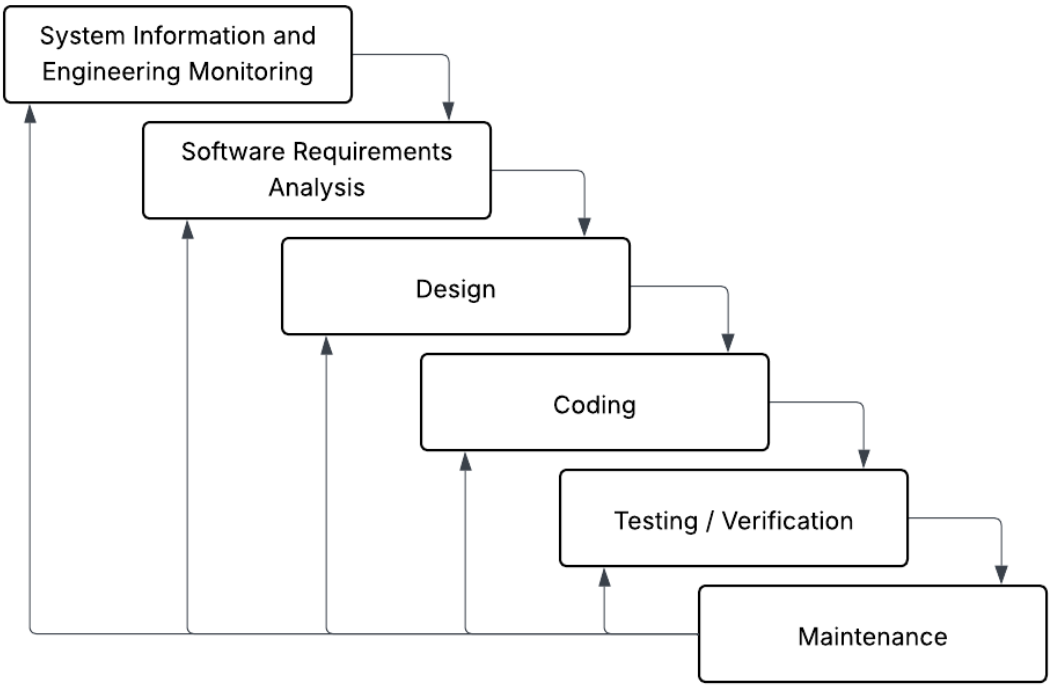


Figure 1. Waterfall method flow

Software Requirements Analysis

This stage established the functional and non-functional requirements of Smart BEVISAT. Based on accident records from Indonesia's major beaches (129 incidents at Parangtritis and 39 at Pangandaran throughout 2023), two core system functions were identified: a preventive function to detect safety-zone violations and alert lifeguards in real time, and a reactive function to deploy an autonomous rescue buoy to victim coordinates. These requirements were translated into three product components: the AI Surveillance Camera, the IoT Safety Buoy, and the Ground Station.

Design

Once all system requirements have been defined, the next step is to design the system architecture and detailed product blueprint. The design process includes:

Software Design

The AI pipeline was architected around YOLOv11 for real-time human detection, ByteTrack for multi-object tracking under occlusion, OpenCV Perspective Transformation for converting pixel coordinates into GPS coordinates, DeepLabv3+ for dynamic waterline segmentation, and Shapely buffer polygons for safe/caution/danger zone delineation. Alerts are transmitted via WebSocket to a ReactJS/VueJS monitoring dashboard streaming over RTSP/WebRTC.

Hardware Design

The IoT buoy electronics were designed around an autopilot controller integrating GPS, gyroscope, and accelerometer sensors with two brushless thruster motors regulated by Electronic Speed Controllers (ESC). Motor selection was based on thrust-drag analysis: two 20 kg-thrust motors produce a combined maximum thrust of 2 × 196 N = 392 N. Under no-load conditions, beach water drag resistance of ±10–15 N yields an estimated maximum speed of 4–5 m/s (14–18 km/h). Under maximum operational load (95 kg total system weight), drag increases to ±70–100 N, yielding an estimated speed of 2–3 m/s (7–11 km/h) sufficient for rapid yet controlled victim rescue. The power system uses two 6S 22.2V 16,000 mAh LiPo batteries connected in series (44.4V, 16 Ah total, 710 Wh capacity). At full throttle both motors draw 2 × 26 A × 44 V = 2,288 W, giving an endurance of approximately 18.6 minutes; at realistic operational power (±1,400 W) endurance extends to approximately 30 minutes. The ground station power system was designed with solar panels and a wind turbine feeding a battery bank for 24-hour autonomous operation.

Manufacturing Design

The buoy hull follows a catamaran form factor in fiberglass for stability in ocean waves, with propellers positioned below the waterline to maximize thrust efficiency and minimize cavitation.

AI Model Development

The YOLOv11 model was fine-tuned using a custom beach dataset collected at Drini Beach, Gunungkidul, Indonesia. The initial training phase used a single-condition dataset variant (daytime, clear weather) to produce a baseline model robust to partially visible and occluded subjects. A second training phase expanded the dataset to include multiple condition variants covering four area types (rip current, rough waves, moderate waves, calm water), three weather conditions (sunny, cloudy, overcast), four time-of-day conditions (morning, afternoon, evening, night), and three tidal levels (high, normal, low tide). The dataset was annotated with bounding boxes for the human class and split into training, validation, and test subsets at an 80/10/10 ratio. Fine-tuning was performed using the Ultralytics framework on a GPU-equipped workstation, with early stopping applied based on validation mAP to prevent overfitting. DeepLabv3+ was similarly fine-tuned on a beach-specific segmentation dataset to enable accurate waterline delineation under varying lighting and wave conditions.

Implementation

At this stage the designs were realized into functional prototypes. Software development included coding the Computer Vision AI pipeline, the buoy autopilot waypoint system, and the monitoring dashboard with automatic alarm triggering. Hardware production involved assembly of the buoy electronics, IoT microcontrollers, GPS and inertial sensors, and the ground station power management system. Manufacturing covered fiberglass hull fabrication, ground station pole construction, and waterproof casing production, followed by Quality Control (QC) inspection.

Testing

System testing was conducted in two rounds, an initial evaluation and a post-revision evaluation following performance improvements at two sites: the laboratory of Mitra PT. Mataran Karya, and

Drini Beach, Gunungkidul (the early-adopter deployment site). Testing comprised five evaluation protocols:

a. Software reliability testing using Blackbox and Whitebox methods to verify all functional pathways.
b. AI camera accuracy testing, measuring the accuracy of 4-point GPS coordinate mapping from drone-assisted perspective transformation.
c. Autopilot waypoint accuracy testing, measuring the buoy's navigational error (target: ≤1 meter) when dispatched to AI-generated victim coordinates.
d. Waterproof rating testing to the IP68 standard (immersion at 1.5 m depth for 30 minutes).
e. Buoy operational capacity testing, measuring load capacity, motor speed, and battery endurance under loads of 50 kg, 75 kg, and 100 kg at throttle levels of 50%, 75%, and 100%, repeated five times per condition to compute mean speed and standard deviation.

Field tests were conducted across four environmental variation axes: area type (rip current, rough waves, calm water), weather (sunny, cloudy, rainy), time of day (morning, afternoon, evening, night), and tidal level (high tide, normal tide, low tide). Ground truth for AI detection tests was established by manual annotation of test video frames by two independent annotators, with inter-annotator agreement verified before scoring. Buoy speed ground truth was measured by stopwatch over a fixed 10-metre course, consistent with the methodology reported in the companion buoy stability study [13].

Results and Discussions

The Smart BEVISAT prototype integrates three subsystems: an Artificial Intelligence (AI)-based surveillance camera, an Internet of Things (IoT)-based safety buoy, and a self-powered ground station. Testing was conducted in two phases: an initial evaluation phase and a post-revision evaluation phase at the Mitra laboratory and at Drini Beach, Gunungkidul, which was designated as the deployment site for early adopters.

IoT Safety Buoy

Physical Design and Buoyancy

This safety buoy utilizes a catamaran-type USV hull made of lightweight fiberglass. Two brushless propulsion motors controlled by an Electronic Speed Controller (ESC) generate differential thrust for directional maneuvering. The autopilot controller integrates data from GPS, a gyroscope, and an accelerometer to autonomously navigate to the victim's GPS coordinates received from the AI camera subsystem. This is a key functional difference from previous remote-controlled life raft prototypes, which required continuous manual input from an operator [13]. A technical diagram of the IoT life raft can be seen in Figure 2.

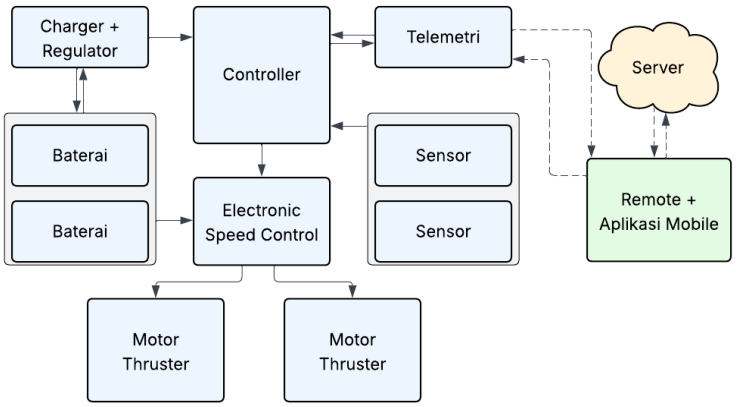


Figure 2. Float diagram scheme.

The total gross buoyancy of the float system, consisting of the fiberglass hull and cork buoyancy elements, is 205.8 kg, calculated using Archimedes' principle ($F = \rho \times g \times V$) as shown below. After subtracting the dry mass of the buoy itself, which is approximately 15 kg, the net carrying capacity is about 190 kg, far exceeding the operational design target of 80–100 kg and providing a substantial safety margin for victim rescue scenarios, which is obtained from the calculation below:

```
Float size              = 100 cm × 70 cm × 30 cm
Float volume            = 0.21 m³
Floating volume         = 0.21 m³ × buoyancy capacity 60%
                        = 0.126 m³
Buoyancy force          = ρ_water × g × V
                        = 1,000 × 9.81 × 0.126
                        = 1,236 N or 126 kg
Cork volume (50%)       = 0.21 m³ × 50% = 0.105 m³
Cork density            = 0.24 g/cm³ = 240 kg/m³
Buoyancy force of cork  = ρ_water × g × V_cork
                        = 1,000 × 9.81 × 0.105
                        = 1,030.5 N or 105 kg
Weight of cork          = ρ_cork × g × V_cork
                        = 240 × 9.81 × 0.105
                        = 247.2 N or 25.2 kg
Net buoyancy of cork    = 1,030.5 N – 247.2 N
                        = 783.3 N or 79.8 kg
Total buoyancy force    = 126 kg + 79.8 kg
                        = 205.8 kg
```

Figure 3. Electric buoy.

The weight of the float is estimated to be about 15 kg, consisting of a U-shaped fiber body weighing about 8 kg, 2 motors + ESC weighing about 2 kg, 2 batteries (2 kg × 2) weighing about 4 kg, and a controller + electronics weighing about 1 kg. The picture of the float can be seen in figure 3. The specifications of the motor and battery are calculated as follows:

```
Max thrust      = 2x20kg thrust
                = 2x196N
                = 392N
Without load
Beach water drag resistance  ±10-15 N
Estimated speed = 4-5m/s (14-18km/h)

With maximum load (95kg total)
Drag increases to ±70-100N

Estimated speed = 2-3m/s (7-11 km/h)
```

The battery power of this buoy is calculated based on:

```
Battery Data
2 x 16,000 mAh 6S 22.2V → in series → 44.4V, 16 Ah total
Capacity (Wh)              = 44.4V x 16Ah = 710 Wh

Motor Power Consumption
Full throttle power        = 2 motors x (26 A x 44V) = 2288
W total,
Full power                 = 710 Wh ÷ 2288 W = 18.6 minutes
Realistic (±60% × 1400 W)= 710 Wh ÷ 1400 W = 30.4 minutes
No load (±800 W)           = 710 Wh ÷ 800 W  = 53.2 minutes
```

Performance Characteristic

Detailed performance characterisation of the electric safety buoy including measured speed under varying loads (50, 75, 100 kg) and throttle levels (50%, 75%, 100%), battery endurance, remote-control range, and PID stability behaviour is reported in the companion study by Kusrini et al. [13], which forms the hardware foundation of Smart BEVISAT. Key results relevant to the present integrated system are summarised in Table 1.

Table 1. Summary of IoT safety buoy performance

Parameter	Value	Condition
Mean speed — 100 kg load, 100% throttle	1.20 m/s (SD = 0.049, n = 5)	Stopwatch, 10 m course
Mean speed — 50 kg load, 100% throttle	1.53 m/s (SD = 0.062, n = 5)	Stopwatch, 10 m course
Battery endurance at 100% throttle	30 minutes (all loads)	Full discharge trial
Maximum carrying capacity	100 kg	Static buoyancy test
Effective remote-control range	300 m (500 m with latency)	RF 434 MHz module

What Smart BEVISAT adds beyond the companion study is the autonomous dispatch pipeline: when the AI camera detects a safety-zone violation, it computes the GPS coordinates of the subject and transmits them to the buoy autopilot, which then navigates to that position without operator intervention. This end-to-end integration (from AI detection to autonomous buoy arrival) is the primary technical contribution of the present work and has no direct precedent in the prior literature [10], [11], [12].

AI surveillance camera Functional Results

The AI camera subsystem successfully performed all designed functions under daytime coastal conditions. Figure 4 shows representative YOLOv11 detection output with bounding boxes drawn around subjects both on land and at the waterline. A technical diagram of the AI camera, from object detection to notification sending, can be seen in figure 5. The complete function-algorithm mapping is provided in Table 2.

Figure 4. Object detection

Table 2. AI surveillance camera functions and algorithms

Function	Algorithm / Technology
Human object detection	YOLOv11 (Ultralytics), optimized with TensorRT
Multi-object tracking	ByteTrack
Image perspective correction	OpenCV Perspective Transformation
Waterline segmentation	DeepLabv3+
Zone boundary determination	Shapely buffer polygon
Zone crossing detection	Shapely + OpenCV visualization
Local-to-GPS coordinate conversion	Grid/matrix method post-perspective transform
Alert audio playback	Python pygame / playsound
Real-time monitoring dashboard	RTSP/WebRTC + WebSocket + ReactJS/VueJS

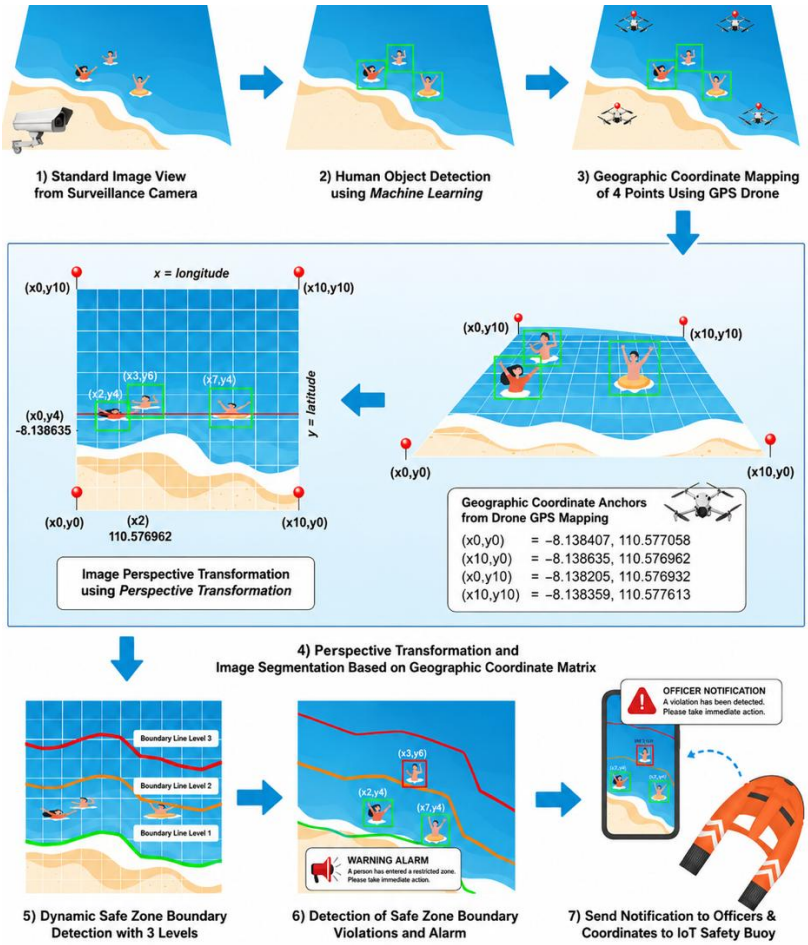


Figure 5. Technical diagram of an AI camera

Ground Station

The ground station integrates solar panels and a wind turbine feeding a battery bank, providing 24-hour autonomous operation without grid power (Figure 6). It supplies wireless internet connectivity for remote dashboard monitoring and recharges the buoy's 48V LiPo battery between deployments. The ground station layout can be seen in Figure 6.

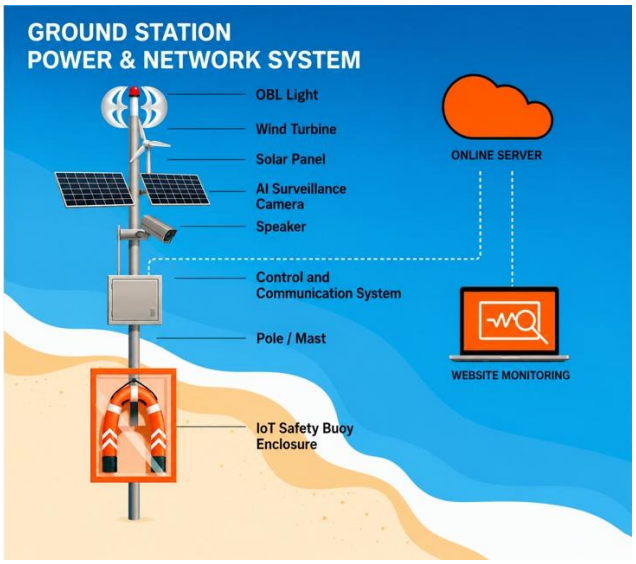


Figure 6. Ground station

Key Performance Indicator (KPI) Summary

Table 3 compares KPI targets against test outcomes. Buoy-level KPIs are sourced from the companion study [13]; system-level KPIs (software reliability, AI detection, waypoint accuracy) reflect Smart BEVISAT's integrated testing.

Table 3. KPI targets vs. measured results.

KPI	Target	Measured / Status
Software reliability (Blackbox / Whitebox)	100%	100% — all paths passed ✓
Zone crossing detection latency	< 2 s	< 2 s — alarm simulation ✓
Human detection accuracy (mAP)	> 55%	Full-dataset evaluation in progress *
Buoy speed (100 kg, 100% throttle)	> 1.0 m/s	1.20 m/s [13] ✓
Buoy waypoint accuracy	Error ≤ 1 m	Field evaluation in progress *
Waterproof rating	IP68, 1.5 m/30 min	Passed immersion test ✓
Buoy carrying capacity	100 kg / 30 min	100 kg / 30 min [13] ✓

* Results for mAP and waypoint accuracy will be reported after completion of the second-round field evaluation currently in progress at Drini Beach.

Discussions

The integrated Smart BEVISAT system achieves its primary design objective: an autonomous, end-to-end pipeline from real-time AI detection of a safety-zone violation to autonomous buoy dispatch to victim coordinates. This pipeline is the defining novzelty of the present work relative to both the companion buoy study [13] and the broader related literature [10], [11], [12].

The companion study [13] demonstrated that the catamaran buoy hull, PID stability controller, and RF remote system can reliably operate up to 300 m and carry loads up to 100 kg.

Smart BEVISAT replaces the manual RF control loop with a GPS waypoint autopilot driven by AI-computed coordinates, eliminating the dependency on operator proximity and reaction time. In real accident scenarios such as the January 2025 Drini Beach incident where 13 students were swept away simultaneously [4] a manual operator cannot reliably direct a buoy to multiple victims; autonomous dispatch addresses this directly.

Compared to Alotaibi [10], whose ResNet50-based pool safety system achieves 99% detection accuracy in a controlled indoor environment, the Smart BEVISAT AI camera operates under substantially more challenging open-ocean conditions: variable lighting, wave-induced motion blur, partial occlusion by surf, and dynamic waterline geometry. The multi-algorithm pipeline (YOLOv11 + ByteTrack + DeepLabv3+) was specifically designed to handle these conditions. Full mAP quantification across all four environmental variant axes (area type, weather, time of day, tidal level) is ongoing. Testing of area variations can be seen in Figure 7.

Compared to Campo et al. [12], whose LoRa wristband platform improves lifeguard situational awareness, Smart BEVISAT requires no visitor-worn hardware, removing adoption friction in public tourist settings. The trade-off is line-of-sight camera dependency; deployment of overlapping camera zones is recommended for beaches with complex topography or large crowds.

Three limitations of the current system warrant acknowledgement. First, AI camera performance under night-time and rain conditions has not yet been fully quantified, motivating infrared camera integration as near-term future work. Second, the 30-minute battery endurance at full throttle is sufficient for a single-incident response but may be limiting in multi-incident scenarios; buoy-mounted solar charging is a planned enhancement. Third, the GPS waypoint autopilot has been validated in calm-to-moderate wave conditions; further PID controller tuning is needed for high-swell rip-current environments.

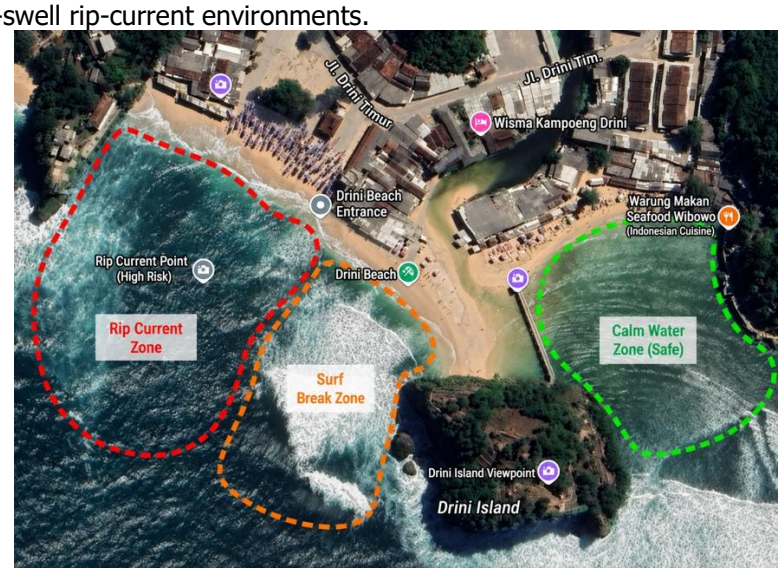


Figure 7. Testing area variations

Conclusion

This research has successfully developed a prototype Smart Beach Visitor Safety System (Smart BEVISAT) as a technological solution to address the high risk of accidents in beach tourist areas. Through the application of the waterfall methodology, the system consisting of AI Surveillance Cameras, IoT Safety Buoys, and Ground Stations was successfully implemented in accordance with the analyzed requirements. The preventive function of the system is realized through AI Surveillance Cameras, which have been proven capable of detecting and tracking human objects and automatically identifying violations of safe zone boundaries. Meanwhile, the reactive function is supported by IoT Safety Buoys, which are designed with adequate technical specifications to perform emergency rescue operations quickly and autonomously. Testing results confirm the system's technical viability: the IoT safety buoy achieved a measured speed of 1.20 m/s under a 100 kg load (SD = 0.049, n = 5) with a 30-minute battery endurance at full throttle, a gross buoyancy of 205.8 kg providing a net carrying capacity of approximately 190 kg, and a maximum

effective remote-control range of 300 m. Software reliability testing returned a 100% pass rate across all functional pathways, and IP68 waterproof certification was confirmed at 1.5 m immersion for 30 minutes. The integration of AI and IoT technologies in this system shows great potential for improving the efficiency and effectiveness of coastal surveillance, reducing dependence on manual monitoring, and ultimately reducing the number of fatalities due to marine accidents. The implementation of this system is expected to make a significant contribution to improving safety standards and supporting the sustainability of the maritime tourism industry in Indonesia.

Acknowledgments

This research is funded by the Ministry of Higher Education, Science and Technology under the 2025 Matching Fund Program in accordance with Decree Number: 09/C4/O/2025 dated July 14, 2025. Contract Number 144/PKS/C.C4/PPK.DHK/VII/2025. University Contract Number 002/MoA/AMIKOM/VII/2025 dated July 15, 2025. Subcontract Number 041/KONTRAK-LPPM/AMIKOM/VII/2025 dated July 18, 2025. The research was also supported by PT. Mataran Karya and AMIKOM University Yogyakarta. We would like to express our gratitude to all parties who have helped make this research possible.

References

[1] D. Kaczan *et al.*, *Hot Water Rising: The Impact of Climate Change on Indonesia's Fisheries and Coastal Communities*. World Bank, 2023. doi: 10.1596/40564.

[2] D. Jenderal, P. Kelautan, and D. A. N. Ruang, "Direktorat," pp. 1–143, 2024.

[3] M. Woods, W. Koon, and R. W. Brander, "Identifying risk factors and implications for beach drowning prevention amongst an Australian multicultural community," *PLoS One*, vol. 17, no. 1, p. e0262175, Jan. 2022, doi: 10.1371/journal.pone.0262175.

[4] P. Gunungkidul, "Pencarian Masih Dilakukan, 13 Pelajar Asal Mojokerto Terseret Ombak di Pantai Drini, Tiga Meninggal Dunia, Satu Hilang," Polres Gunungkidul. [Online]. Available: https://jogja.polri.go.id/gunungkidul/tribrata-news/online/detail/pencarian-masih-dilakukan--13-pelajar-asal-mojokerto-terseret-ombak-di-pantai-drini--tiga-meninggal-dunia--satu-hilang.html

[5] D. Armiady, "Absensi Kehadiran Menggunakan Kamera Pengawas Berbasis Teknologi Computer Vision," *JURNAL TIKA*, vol. 6, no. 02, pp. 140–146, Jun. 2021, doi: 10.51179/tika.v6i02.541.

[6] M. T. Subha Devi, M. Dhanalakshmi, S. A, S. ML, and L. N, "Anomaly Detection in Video Surveillance," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*, IEEE, Apr. 2024, pp. 1–3. doi: 10.1109/I2CT61223.2024.10543949.

[7] R. Renugadevi and L. H. Medida, "Artificial Intelligence and IoT-Based Disaster Management System," in *Predicting Natural Disasters With AI and Machine Learning*, IGI Global Scientific Publishing, 2024, pp. 135–146. doi: 10.4018/979-8-3693-2280-2.ch006.

[8] R. Srinath, J. Vrindavanam, V. P. Vasudev, S. Supreeth, H. Raj, and A. Kesarwani, "A Machine Learning Approach for Localization of Suspicious Objects using Multiple Cameras," in *2020 IEEE International Conference for Innovation in Technology (INOCON)*, IEEE, Nov. 2020, pp. 1–6. doi: 10.1109/INOCON50539.2020.9298364.

[9] P. Praveen and K. P. Kumar, "Wearable Location Tracker During Disaster using AI and IoT," in *2024 3rd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, IEEE, Jun. 2024, pp. 208–213. doi: 10.1109/ICAAIC60222.2024.10575619.

[10] A. Alotaibi, "Automated and Intelligent System for Monitoring Swimming Pool Safety Based on the IoT and Transfer Learning," *Electronics (Basel).*, vol. 9, no. 12, p. 2082, Dec. 2020, doi: 10.3390/electronics9122082.

[11] M. C. Domingo, "Deep Learning and Internet of Things for Beach Monitoring: An Experimental Study of Beach Attendance Prediction at Castelldefels Beach," *Applied Sciences*, vol. 11, no. 22, p. 10735, Nov. 2021, doi: 10.3390/app112210735.

[12] G. Campo, D. Carnemolla, M. Di Raimondo, C. Santoro, and F. F. Santoro, "Architecting a Privacy-Aware IoT-Enabled Platform for Bathers Monitoring in Beach Resort: A Case

Study," in *2024 IEEE Conference on Pervasive and Intelligent Computing (PICom)*, IEEE, Nov. 2024, pp. 178–183. doi: 10.1109/PICom64201.2024.00033.
[13] K. Kusrini, E. Pramono, E. H. Muktafin, B. Setiaji, and A. D. Putra, "Perancangan Prototype Pelampung Keselamatan Elektrik dengan Algoritma Proportional Integral Derivative Untuk Kestabilan di Air," *Journal of Information System Research (JOSH)*, vol. 6, no. 1, pp. 321–329, Oct. 2024, doi: 10.47065/josh.v6i1.6021.
[14] A. Saravanos, "A Brief History of the Waterfall Model: Past, Present, and Future," Dec. 2025, doi: 10.1145/3783862.3783879.
[15] Pedroza, G., Paquet, M., & Dion, B. (2025). *A Digital Engineering Methodology for Design, Exploration and Validation of Safety-Critical Software for Integrating AI-based Algorithms*. INCOSE International Symposium. https://doi.org/10.1002/iis2.70011

Matrix: Jurnal Manajemen Teknologi dan Informatika
Volume 16 Issue 2 Year 2026 Pages 74-85
DOI: http://dx.doi.org/10.31940/matrix.v16i2.74-85
URL: http://ojs2.pnb.ac.id/index.php/matrix

Video-based human activity recognition analysis using YOLOv26

Santi Rahayu [1*], Imam Riadi [2], Andri Pranolo [3]

[1,3] *Informatics Study Program, Universitas Ahmad Dahlan, Indonesia*
[1] *Informatics Engineering Study Program, Universitas Pamulang, Indonesia*
[2] *Information System Study Program, Universitas Ahmad Dahlan, Indonesia*

*****Corresponding Author: *2536083045@webmail.uad.ac.id*

Abstract: Human Activity Recognition (HAR) is a rapidly growing research field in computer vision and has various applications, such as intelligent surveillance systems, sports analysis, healthcare, and human-computer interaction. This study aims to analyze the performance of the YOLOv26 Classification model in recognizing video-based human activities using the UCF101 dataset. This study uses six activity classes, namely JumpingJack, Punch, PushUps, Typing, WalkingWithDog, and WritingOnBoard. The research method is carried out through video frame extraction using a uniform sampling technique, the formation of an image classification dataset, YOLOv26 model training, and evaluation at the frame and video levels. Experiments were conducted using 36 configuration combinations consisting of four YOLOv26 model variants (YOLOv26-n, YOLOv26-s, YOLOv26-m, and YOLOv26-l), three variations in the number of frames (8, 16, and 24 frames), and three variations in the number of epochs (50, 100, and 150 epochs). Video evaluation was conducted using a majority voting approach with accuracy, precision, recall, and F1-score metrics. The results showed that all configurations produced video accuracy above 93%. The best configuration was obtained with the YOLOv26-m model with 16 frames and 50 epochs, which achieved video accuracy of 98.26%, precision of 98.15%, recall of 98.37%, and F1-score of 98.23%. Confusion matrix analysis showed that most predictions were on the main diagonal, indicating the model's ability to distinguish human activities with a low error rate. These results prove that YOLOv26 Classification has excellent performance for video-based human activity recognition and has the potential to be applied to various applications based on automatic human activity analysis.

Keywords: Human activity recognition, YOLOv26 classification, UCF101 dataset, video classification, majority voting, computer vision.

History Article: Submitted 6 June 2026 | Revised 13 June 2026 | Accepted 22 June 2026

How to Cite: S. Rahayu, I. Riadi, and A. Pranolo, "*Video-based human activity recognition analysis using YOLOv26,*" *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 16, no. 2, pp. 74–85, 2026, doi: 10.31940/matrix.v16i2.74-85.

Introduction

Human Activity Recognition (HAR) is one of the fastest-growing research areas in computer vision and has attracted considerable attention due to its wide range of applications in intelligent surveillance systems, healthcare monitoring, sports analysis, human-computer interaction, and smart environments [1], [2], [3]. The ability of computer systems to automatically recognize and classify human activities from video data enables more efficient monitoring, decision-making, and behavioral analysis in real-world scenarios. With the increasing availability of video acquisition devices and surveillance cameras, the development of accurate and efficient HAR systems has become an important research topic.

Video-based HAR remains a challenging task because human activities contain both spatial and temporal information. Spatial information represents the visual appearance of objects and body postures in individual frames, while temporal information describes motion patterns and activity evolution across consecutive frames [2], [4]. In addition, variations in lighting conditions, camera viewpoints, background complexity, occlusion, and motion speed can significantly affect recognition performance [5]. These challenges require robust deep learning models capable of effectively capturing discriminative features from video sequences.

Recent advances in deep learning have significantly improved HAR performance. Convolutional Neural Networks (CNNs) have been widely used to extract spatial features, while temporal dependencies are commonly modeled using architectures such as Long Short-Term Memory (LSTM), Gated Recurrent Units (GRU), Temporal Convolutional Networks (TCN), and Transformers. Jindal et al. combined feature optimization with ResNet101V2 and achieved an accuracy of 96.16% on the UCF101 dataset [6]. Ahmad et al. employed a VGG16-BiGRU framework for HAR and demonstrated the effectiveness of combining spatial and temporal learning [7]. Similarly, Shin et al. utilized a hybrid ConvNeXt-TCN architecture and reported an accuracy of 98.46% on UCF101 [8]. Other studies have explored Selective Kernel Networks [9], graph-based learning [10], and ensemble voting strategies [11] to improve activity recognition performance.

Several recent studies have also investigated advanced spatiotemporal learning approaches. Vrskova et al. demonstrated that 3D Convolutional Neural Networks (3DCNNs) can effectively learn spatial and temporal information simultaneously [2]. Alshaya introduced Neuro-Prismatic Video Models based on UniFormerV2 to capture complex temporal relationships in video data [12]. Zahra et al. proposed Adaptive Text Prompting for zero-shot action recognition [13], while Li et al. introduced Mamba-YOWO for efficient spatiotemporal action detection [4]. Furthermore, Isik and Ozcan proposed a novel activation function that improved HAR classification performance on benchmark datasets [14]. Although these methods have achieved promising results, many of them require complex architectures and high computational resources, making practical deployment more challenging.

The YOLO (You Only Look Once) family has become one of the most successful deep learning frameworks for real-time visual recognition tasks due to its balance between accuracy and computational efficiency. Previous studies have demonstrated the effectiveness of YOLO-based approaches for activity recognition and action localization [15], [16]. More recently, YOLOv26 has been introduced as a new generation architecture featuring an NMS-Free design and end-to-end optimization, which significantly improves inference efficiency and deployment performance [17], [18]. Studies have shown that YOLOv26 variants can provide excellent accuracy while maintaining lower computational costs compared with previous generations [19], [20]. Despite these advantages, research investigating the use of YOLOv26 Classification for video-based human activity recognition remains very limited.

The UCF101 dataset is one of the most widely used benchmark datasets for evaluating HAR systems. It consists of 13,320 videos distributed across 101 activity categories collected from realistic environments with substantial variations in motion patterns, viewpoints, and backgrounds [5]. Although numerous studies have reported excellent results on UCF101 using CNNs, 3DCNNs, Transformers, and hybrid architectures [21], [8], [9], the performance characteristics of YOLOv26 Classification on this benchmark dataset have not yet been comprehensively analyzed. Furthermore, the influence of frame sampling strategies and model scales on YOLOv26-based activity recognition performance remains unclear.

Therefore, this study aims to analyze the performance of YOLOv26 Classification for video-based human activity recognition using the UCF101 dataset. Six activity classes, namely JumpingJack, Punch, PushUps, Typing, WalkingWithDog, and WritingOnBoard, were selected for evaluation. The proposed framework employs uniform frame sampling to extract representative frames from videos, followed by image classification using YOLOv26 and majority voting to obtain video-level predictions. Comprehensive experiments were conducted using 36 configuration combinations involving four YOLOv26 variants (YOLOv26-n, YOLOv26-s, YOLOv26-m, and YOLOv26-l), three frame sampling settings (8, 16, and 24 frames), and three epoch configurations (50, 100, and 150 epochs). The findings provide a comprehensive evaluation of YOLOv26 Classification for HAR and offer insights into the relationship between model complexity, temporal representation, and recognition performance.

Methodology

This study uses an experimental method to analyze the performance of the YOLOv26 Classification model in recognizing video-based human activities. The dataset used is UCF101, which is one of the benchmark datasets for Human Activity Recognition (HAR) and can be downloaded at https://www.crcv.ucf.edu/data/UCF101.php. Of the 101 activity classes available

in the dataset, this study only uses six activity classes, namely JumpingJack, Punch, PushUps, Typing, WalkingWithDog, and WritingOnBoard. All stages of the study include dataset selection, reading official data sharing files, determining experimental parameters, extracting video frames, forming a classification dataset, training the YOLOv26 model, evaluating at the frame and video level, storing experimental results, and analyzing the results to obtain the best configuration. The overall research flow is shown in Figure 1.

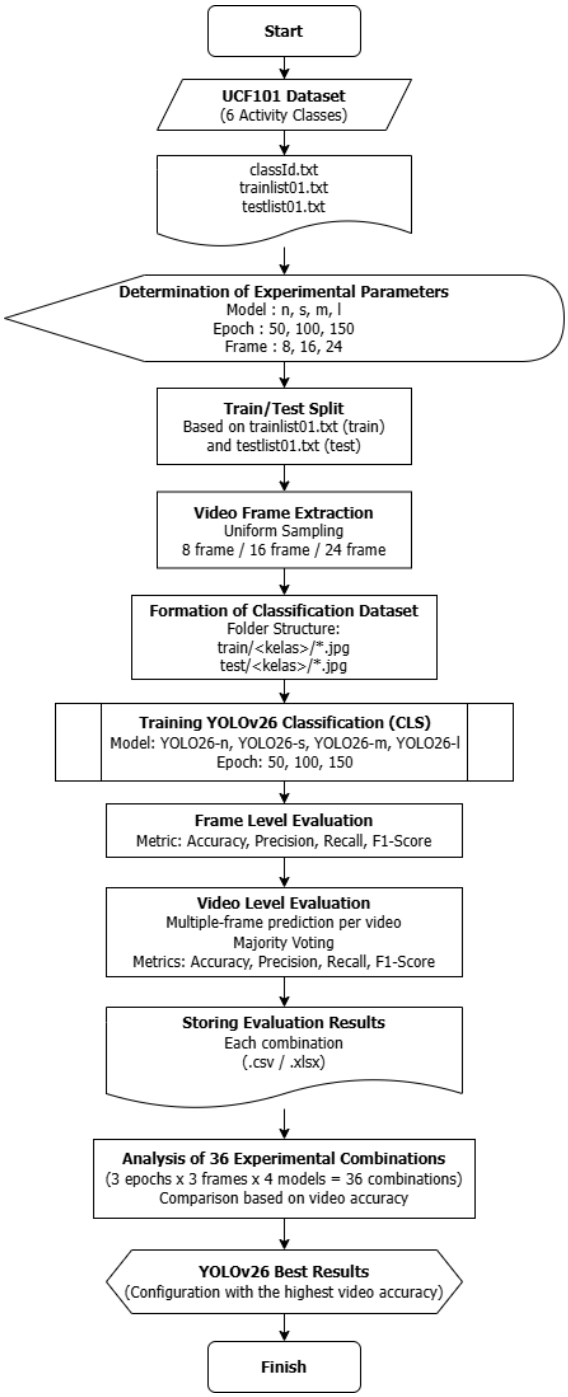


Figure 1. Research methodology flowchart

The number of frames, 8, 16, and 24, was chosen to represent three different levels of temporal resolution. Eight frames were used to represent low temporal information, resulting in lower computational time. Sixteen frames were chosen as an intermediate level, expected to

capture key movement patterns without producing excessive information redundancy. Twenty-four frames were used to test whether increasing temporal information could improve activity classification accuracy. Epoch variations of 50, 100 and 150 were chosen to analyze the effect of training duration on model performance. A value of 50 epochs was used as the initial training baseline, 100 epochs represented intermediate training, and 150 epochs was used to evaluate whether additional learning processes could improve the model's generalization ability or actually cause overfitting.

The model architecture used in this study is based on the YOLOv26 Classification implementation in the Ultralytics framework. Since the official YOLOv26 documentation does not yet provide a detailed visualization of the classification architecture, the model representation is constructed based on the network configuration and component identification results during the training process. Conceptually, the architecture consists of an input image stage, a backbone for feature extraction, a C2PSA (Channel and Spatial Attention)-based feature aggregation module, Global Average Pooling (GAP), a Fully Connected Layer, Softmax, and an output class that generates probabilities for six activity classes. The model architecture representation is shown in Figure 2.

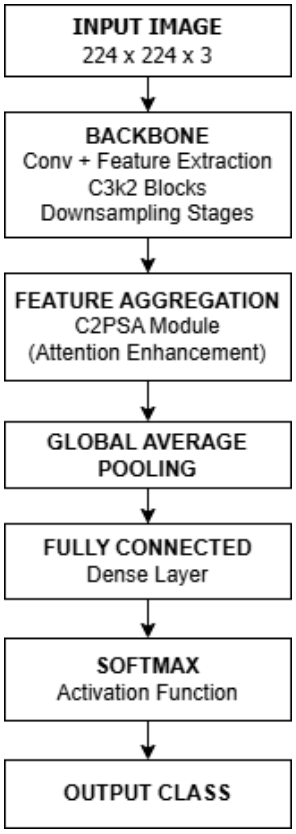


Figure 2. Conceptual architecture of YOLOv26 Classification, the result of interpreting the model implementation in the Ultralytics framework

In addition to the network structure, each YOLOv26 variant has a different model complexity. These differences can be observed from the number of layers, the number of parameters, and the computational requirements expressed in GFLOPs. This information is obtained from the training logs generated during the training process using the Ultralytics framework. Increasing the model size from YOLOv26-n to YOLOv26-l results in an increase in the number of parameters and computational complexity. YOLOv26-n is the lightest model with 1.54 million parameters and a computational requirement of 3.3 GFLOPs, while YOLOv26-l is the largest model with 12.84 million parameters and 49.8 GFLOPs. These differences in model capacity allow for an analysis of the effect of network size on human activity recognition

performance on the UCF101 dataset. A summary of the characteristics of each model is shown in Table 1.

Table 1. Characteristics of the YOLOv26 Classification variant based on the results of model identification in the training process

Model	Layer	Parameter	GFLOPs
YOLOv26-n	86	1.54 M	3.3
YOLOv26-s	86	5.45 M	12.1
YOLOv26-m	106	10.36 M	39.6
YOLOv26-l	176	12.84 M	49.8

All models were trained using the Ultralytics framework with the AdamW optimizer, an initial learning rate of 0.001, a momentum of 0.9, an input image size of 224×224 pixels, and an early stopping mechanism with a patience value of 20 epochs. The batch size was adjusted to the model complexity, namely 8 for YOLOv26-n and YOLOv26-s, 4 for YOLOv26-m, and 2 for YOLOv26-l to avoid GPU memory limitations. Training was run on an NVIDIA GeForce RTX 3050 Laptop GPU with CUDA 12.1.

Results and Discussions

The experimental results code for this research is in here .Based on the evaluation results, there are three configurations that produce the highest video accuracy of 98.26%, namely YOLOv26-m with 16 frames and 50 epochs, YOLOv26-l with 16 frames and 100 epochs, and YOLOv26-l with 16 frames and 150 epochs. Despite having the same accuracy value, the YOLOv26-m configuration shows higher precision, recall, and F1-score values than the two YOLOv26-l configurations. In addition, the model requires fewer epochs, making it more efficient in terms of training time and computational requirements. Therefore, YOLOv26-m with 16 frames and 50 epochs was chosen as the best configuration in this study. The model with the highest accuracy can be seen in Table 2.

Table 2. Results of 36 combinations of YOLOv26 classification experiments

No	Epoch	Frame	Model	Video Accuracy	Video Precision	Video Recall	Video F1-Score
1	50	16	YOLOv26-m	0.983	0.981	0.984	0.982
2	100	16	YOLOv26-l	0.983	0.981	0.982	0.981
3	150	16	YOLOv26-l	0.983	0.981	0.982	0.981
4	100	16	YOLOv26-m	0.978	0.981	0.974	0.976
5	150	8	YOLOv26-s	0.978	0.981	0.974	0.976
6	150	16	YOLOv26-m	0.978	0.981	0.974	0.976
7	50	8	YOLOv26-s	0.970	0.971	0.965	0.966
8	50	16	YOLOv26-s	0.970	0.972	0.965	0.967
9	50	16	YOLOv26-l	0.970	0.967	0.965	0.966
10	50	24	YOLOv26-n	0.970	0.971	0.965	0.966
…	…	…	…	…	…	…	…
36	100	8	YOLOv26-l	0.939	0.951	0.938	0.937

Comparison of video accuracy values obtained from 36 experimental combinations involving variations of the YOLOv26 model, the number of frames, and the number of epochs. In general, all configurations produced high performance with video accuracy values above 93%, indicating that YOLOv26 is able to recognize human activity well on the UCF101 dataset. The best configuration was obtained with the YOLOv26-m model with 16 frames and 50 epochs, which achieved a video accuracy of 98.26%. In addition, it can be seen that the use of 16 frames tends to produce higher accuracy than the use of 8 frames or 24 frames. Although several configurations

of the YOLOv26-l model also achieved similar accuracy values, the YOLOv26-m model was chosen as the best configuration because it was able to provide higher precision, recall, and F1-score values with a fewer number of epochs. These results indicate that the choice of the number of frames and model complexity has a significant influence on the performance of video-based human activity recognition using YOLOv26, as shown in Figure 3.

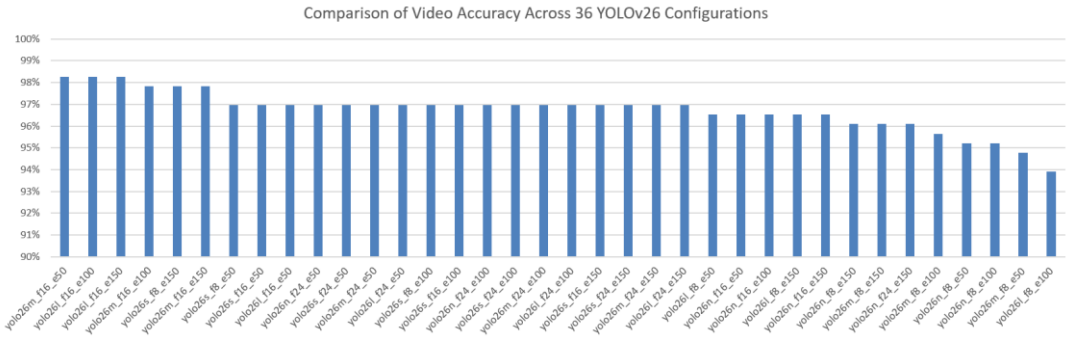


Figure 3. Comparison of video accuracy on 36 experimental combinations consisting of variations of the YOLOv26 model, number of frames, and number of epochs

To better understand the significance of the obtained results, the performance of the proposed YOLOv26 Classification framework was compared with several previous HAR studies conducted on the UCF101 dataset. Jindal et al. [6] reported an accuracy of 96.16% using a feature optimization strategy combined with ResNet101V2, while Ahmad et al. [22] achieved 91.79% using a VGG16-BiGRU architecture. Khare et al. [23] reported an accuracy of 96.23% using a micro-network deep convolutional neural network, and Shin et al. [8] achieved 98.46% using a hybrid ConvNeXt-TCN framework. In this study, the best YOLOv26-m configuration achieved a video accuracy of 98.26%, precision of 98.15%, recall of 98.37%, and F1-score of 98.23%. These results indicate that YOLOv26 Classification is capable of achieving performance comparable to state-of-the-art HAR approaches while employing a simpler pipeline based on frame extraction and majority voting.

The superior performance obtained by YOLOv26-m suggests that model scale plays an important role in balancing representation capacity and generalization ability. Although YOLOv26-l contains a larger number of parameters and higher computational complexity, its performance was not consistently better than YOLOv26-m. In fact, both models achieved the same maximum accuracy of 98.26% under certain configurations, while YOLOv26-m produced slightly higher precision, recall, and F1-score values using fewer training epochs. This finding indicates that increasing model complexity does not necessarily improve recognition performance and may introduce unnecessary computational overhead. Therefore, the medium-scale architecture provides a more favorable trade-off between accuracy and efficiency for the selected HAR classes.

Another important finding is the influence of frame sampling on recognition performance. The results show that configurations using 16 frames generally outperformed those using 8 or 24 frames. This observation suggests that 16 frames provide an optimal balance between spatial and temporal information. When only 8 frames are used, some motion transitions may be omitted, limiting the model's ability to capture complete activity patterns. Conversely, increasing the number of frames to 24 may introduce redundant information between adjacent frames, reducing the contribution of additional temporal information. Similar observations have been reported in previous HAR studies, where excessive temporal redundancy did not always lead to performance improvements [4], [8].

The effectiveness of the proposed majority voting strategy can also be observed from the high video-level performance achieved across all configurations. While classification is performed independently on individual frames, majority voting aggregates frame-level predictions into a more robust video-level decision. This mechanism helps reduce the impact of occasional frame misclassifications caused by motion blur, pose variation, or background complexity. The consistently high video accuracy obtained in this study demonstrates that majority voting is an effective and computationally efficient alternative to more complex temporal modeling approaches such as LSTM, GRU, or Transformer-based architectures. As a result, the proposed

framework offers a practical solution for real-time HAR applications where computational efficiency is an important consideration.

The best model performance analysis was performed using a confusion matrix to evaluate the distribution of predictions for each activity class. The test results showed that most samples were correctly classified, as indicated by the dominance of values on the main diagonal of the matrix. The WritingOnBoard, Typing, WalkingWithDog, Punch, and JumpingJack classes had excellent recognition rates with a relatively small number of errors. The PushUps class still showed some misclassifications, especially against Punch and WalkingWithDog. Nevertheless, overall, the model was able to distinguish the six human activities very well. The confusion matrix results for the best YOLOv26-m model with 16 frames and 50 epochs are shown in Figure 4.

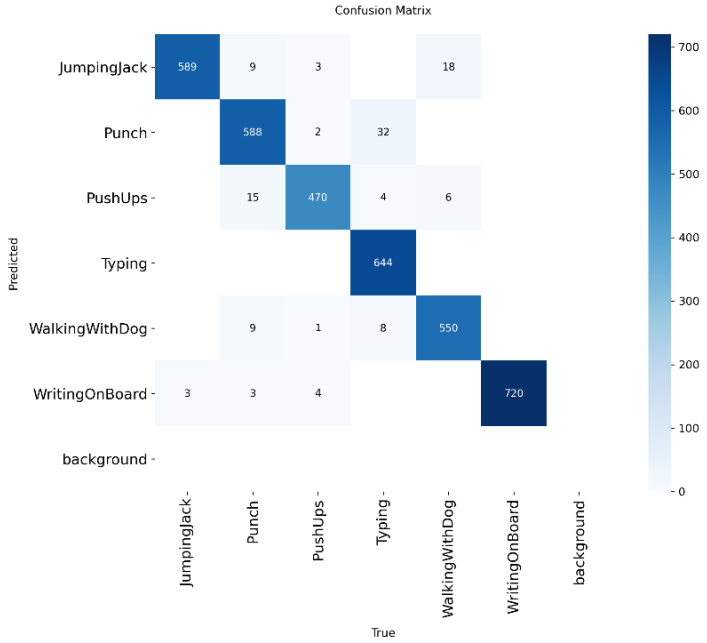


Figure 4. Confusion matrix of human activity classification results using the YOLOv26-m model with 16 frames and 50 epochs

The normalization matrix shows the proportion of correct predictions for each activity class. A value close to 1 indicates that the model has high classification ability for that class. The normalization matrix can be seen in Figure 5.

Figure 5. Normalized confusion matrix of the best YOLOv26-m model with 16 frames and 50 epochs

The confusion matrix analysis further confirms the effectiveness of the proposed framework. Most predictions were concentrated along the main diagonal, indicating that the model successfully distinguished the six activity classes with a low error rate. Activities such as WritingOnBoard and Typing achieved particularly strong recognition performance due to their distinctive visual characteristics and relatively consistent body movements. In contrast, several misclassifications were observed between PushUps and Punch, as both activities involve repetitive upper-body movements that may appear visually similar in certain frames. Similar confusion patterns have been reported in previous HAR studies using UCF101, where activities with overlapping motion characteristics are generally more difficult to distinguish accurately [2], [7]. Nevertheless, the overall number of misclassified samples remained very small, demonstrating the robustness of the YOLOv26 Classification model for video-based activity recognition.

Based on the training curve, the loss function value, indicated by the training loss, experienced a significant decrease from around 0.46 in the initial epoch to nearly 0.02 at the end of training. This decrease indicates that the model is increasingly able to minimize classification errors during the learning process. Meanwhile, the accuracy_top1 and accuracy_top5 values tended to increase and reached a stable condition, indicating that the model has achieved convergence. These results indicate that the model successfully learned human activity patterns without showing any signs of instability during the optimization process. A visualization of the changes in the loss function and accuracy values during training is shown in Figure 6.

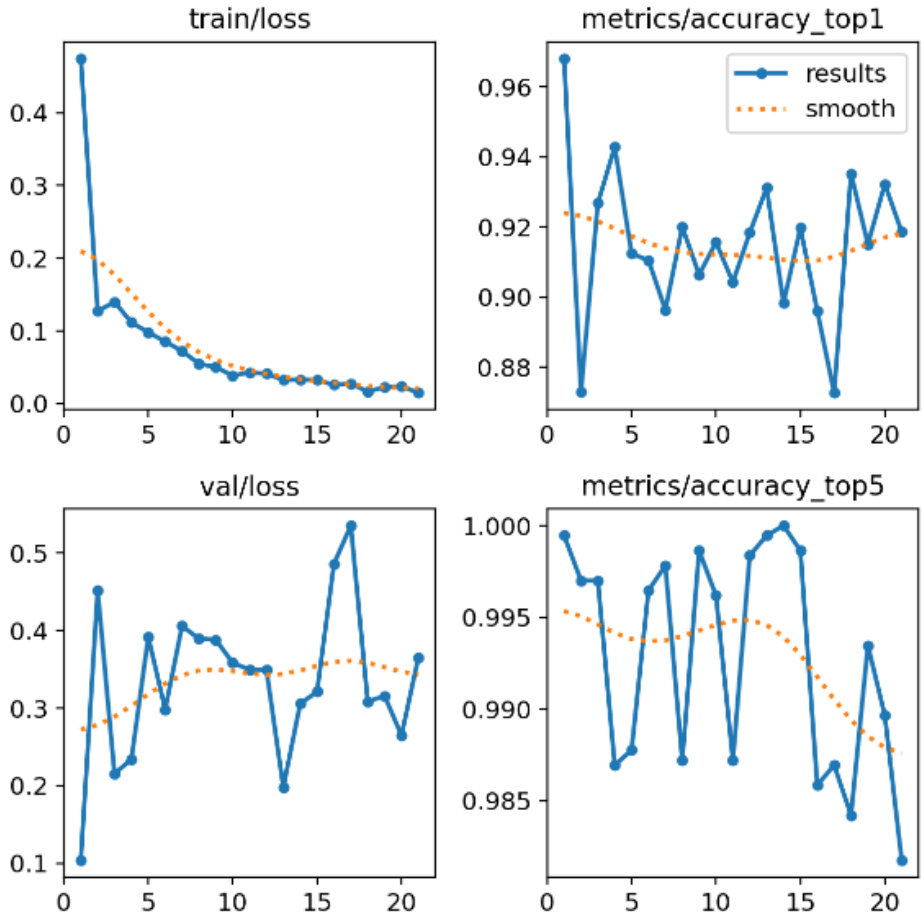


Figure 6. Training and validation curves of the YOLOv26-m model using 16 frames and 50 epochs

To visually illustrate the model's performance, several examples of prediction results on the validation data are shown. The label on each frame indicates the predicted class and the confidence value generated by the model. Each frame shows the predicted activity label and the confidence score generated by the model. The prediction results show that the model is able to recognize activities such as WritingOnBoard, PushUps, and Typing with a high level of confidence,

as indicated by confidence values approaching 1.0 in most samples. Although there are some frames with lower confidence values, such as the PushUps activity, the model is still able to classify the activity correctly. These results indicate that the YOLOv26-m model with 16 frames and 50 epochs has good ability to recognize various human activities on data that was not used during the training process, as shown in Figure 7.

Figure 7. Example of the best prediction results of the YOLOv26-m model with 16 frames and 50 epochs on the validation data

The results showed that using 16 frames resulted in higher accuracy compared to using 8 or 24 frames. This indicates that this number of frames provides an optimal balance between spatial and temporal information. In the 8-frame configuration, some motion information is potentially lost due to too little temporal representation. Conversely, in the 24-frame configuration, there is the possibility of redundant information between frames that does not contribute significantly to the classification process. Therefore, 16 frames is the most effective number in representing the characteristics of human activity in the UCF101 dataset used in this study.

Further research could be conducted by expanding the number of activity classes and using all classes in the UCF101 dataset or larger datasets such as HMDB51 and Kinetics-400 to test the model's generalization ability in more complex scenarios. Furthermore, future research could explore the use of a larger number of frames or more adaptive frame selection methods to better represent temporal information in videos. Developments could also be made by combining YOLOv26 Classification with temporal learning-based models such as Long Short-Term Memory (LSTM), Temporal Convolutional Network (TCN), or video-based Transformer to more optimally utilize inter-frame relationships. Furthermore, testing in real-world environments with varying lighting, shooting angles, backgrounds, and video quality is also necessary to determine the

model's robustness in practical implementation. Finally, further research could evaluate computational efficiency aspects such as inference time, GPU memory usage, and resource consumption on edge computing devices or real-time systems so that the developed model not only has high accuracy but can also be implemented effectively in real operational environments.

Conclusion

This study successfully implemented the YOLOv26 Classification model to recognize six human activities in the UCF101 dataset: JumpingJack, Punch, PushUps, Typing, WalkingWithDog, and WritingOnBoard. The research process involved extracting video frames using a uniform sampling method, creating an image classification dataset, training the YOLOv26 model with four architecture variants (YOLOv26-n, YOLOv26-s, YOLOv26-m, and YOLOv26-l), and evaluating it at the frame and video levels using a majority voting approach. All experiments were conducted on 36 configuration combinations, varying in the number of epochs (50, 100, and 150), the number of frames (8, 16, and 24), and the YOLOv26 model variants.

The test results showed that all configurations were capable of producing high performance with video accuracy values above 93%. Based on an analysis of 36 experimental combinations, the best configuration was obtained by YOLOv26-m with 16 frames and 50 epochs, achieving a video accuracy of 98.26%, a precision of 98.15%, a recall of 98.37%, and an F1-score of 98.23%. These results indicate that the combination of a sufficiently representative number of frames and medium model complexity can produce optimal performance without requiring a larger number of epochs. These findings also indicate that increasing the number of epochs does not always result in a significant increase in accuracy, as some configurations with 100 and 150 epochs produced similar or even lower accuracy values than the best configuration.

A confusion matrix analysis showed that the best model was able to distinguish most activities with a very low error rate. Most predictions were concentrated on the main diagonal of the matrix, indicating that the model successfully recognized the correct class in the majority of the test data. The WritingOnBoard and Typing activities were the easiest to recognize, while relatively more misclassifications occurred for PushUps and Punch, which have similar body movement characteristics across several video frames. However, the number of errors was still very small compared to the number of correct predictions, so it did not affect the overall performance of the model.

The training curve shows that the loss function value consistently decreased during the training process, while the accuracy value tended to increase and reached a state of convergence. This indicates that the model successfully learned the human activity patterns in the dataset without experiencing instability during the optimization process. Furthermore, examples of prediction results on the validation data demonstrate that the model is capable of providing appropriate class predictions with a high degree of confidence across the various human activities tested. Overall, the research results demonstrate that YOLOv26 Classification has excellent capabilities for video-based human activity recognition through a frame extraction and majority voting approach, thus offering potential applications in various applications such as intelligent surveillance systems, human behavior analysis, sports, healthcare, and human-computer interaction.

Acknowledgments

The authors would like to thank Universitas Ahmad Dahlan, Yogyakarta, Indonesia, for the support and facilities provided during the completion of this research.

References

[1] A. Roy, K. D. Gaikwad, A. J. Hude, J. S. Shinde, J. Parekh, and H. R. Nagrani, "Explainable AI (XAI) for object detection and application to satellite imagery," *IEEE Access*, vol. 13, pp. 185597–185623, 2025, doi: 10.1109/ACCESS.2025.3624198.

[2] R. Vrskova, R. Hudec, P. Kamencay, and P. Sykora, "Human activity classification using the 3DCNN architecture," *Appl. Sci.*, vol. 12, no. 2, pp. 1–17, 2022, doi: 10.3390/app12020931.

[3] H. Alshaya, "Neuro-Prismatic video models for causality-aware action recognition in neural rehabilitation systems," *Mathematics*, vol. 14, no. 8, Apr. 2026, doi: 10.3390/math14081341.
[4] M. Li, X. Jiangbo, W. Lu, D. Xinguan, and S. Shuang, "Mamba-YOWO an efficient spatio temporal representation framework for action detection," vol. 48, pp. 1–11, 2026.
[5] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," Dec. 2012, [Online]. Available: http://arxiv.org/abs/1212.0402.
[6] S. Jindal, M. Sachdeva, and A. K. S. Kushwaha, "Quantum behaved intelligent variant of gravitational search algorithm with deep neural networks for human activity recognition Dept . of Computer Science and Engineering I . K . Gujral Punjab Technical University , Jalandhar , India Guru Ghasidas Vishwavi," vol. 50, no. 2, pp. 1–20, 2023.
[7] T. Ahmad, J. Wu, H. S. Alwageed, F. Khan, J. Khan, and Y. Lee, "Human Activity recognition based on deep-temporal learning using convolution neural networks features and bidirectional gated recurrent unit with features selection," *IEEE Access*, vol. 11, no. April, pp. 33148–33159, 2023, doi: 10.1109/ACCESS.2023.3263155.
[8] J. Shin, M. A. M. Hasan, M. Maniruzzaman, S. Nishimura, and S. Alfarhood, "Video-Based human activity recognition using hybrid deep learning model," *C. - Comput. Model. Eng. Sci.*, vol. 143, no. 3, pp. 3615–3638, 2025, doi: 10.32604/cmes.2025.064588.
[9] B. Srinivasaiah and J. Ramegowda, "Human activity recognition using selective kernel network-2D convolutional neural network with ArcFace loss," *IAES Int. J. Artif. Intell.*, vol. 15, no. 1, pp. 350–360, 2026, doi: 10.11591/ijai.v15.i1.pp350-360.
[10] A. A. R. K. Bsoul, "Human activity recognition using graph structures and deep neural networks," *Computers*, vol. 14, no. 1, 2025, doi: 10.3390/computers14010009.
[11] U. M. Butt, H. A. Ullah, S. Letchmunan, I. Tariq, F. H. Hassan, and T. W. Koh, "Leveraging transfer learning for spatio-temporal human activity recognition from video sequences," *Comput. Mater. Contin.*, vol. 74, no. 3, pp. 5017–5033, 2023, doi: 10.32604/cmc.2023.035512.
[12] H. Alshaya, "Neuro-Prismatic video models for causality-aware action recognition in neural rehabilitation systems," *Mathematics*, vol. 14, no. 8, 2026, doi: 10.3390/math14081341.
[13] S. Zahra, N. T. Cao, and T. Kim, "Adaptive text prompting: A training-free framework for zero-shot action recognition," *IEEE Access*, vol. 14, no. February, pp. 45165–45178, 2026, doi: 10.1109/ACCESS.2026.3673165.
[14] Z. V. Isik and T. Ozcan, "ExpAbsTanH: A novel activation function for image and video-based human activity recognition," *IEEE Access*, vol. 14, no. January, pp. 8484–8505, 2026, doi: 10.1109/ACCESS.2026.3653891.
[15] S. Shinde, A. Kothari, and V. Gupta, "YOLO based human action recognition and localization," *Procedia Comput. Sci.*, vol. 133, no. 2018, pp. 831–838, 2018, doi: 10.1016/j.procs.2018.07.112.
[16] M. Elnady and H. E. Abdelmunim, "A novel YOLO LSTM approach for enhanced human action recognition in video sequences," *Sci. Rep.*, vol. 15, no. 1, pp. 1–30, 2025, doi: 10.1038/s41598-025-01898-z.
[17] S. Jing, R. Mai, and X. Gao, "A Method for down quality inspection : YOLO-Based impurity detection and quality quantification," 2026.
[18] C. Julio, F. Silva, C. Del-valle-soto, and C. Bran, "Comparative analysis of previous YOLO detectors and YOLOv26s for real-time weapon detection in video surveillance," no. April, pp. 1–20, 2026, doi: 10.3389/fcomp.2026.1789702.
[19] Y. Aydin, U. Isikdag, S. M. Nigdeli, G. Bekdas, and C. Cakiroglu, "Image-Based Classification of concrete carbonation using YOLO models," *Mater. MDPI*, vol. 19, no. 2198, pp. 1–44, 2026.
[20] H. Keshk and A. Abdallah, "Real-Time UAV-Based oil pipeline and visual anomaly detection using YOLOv26n: A dataset and edge-deployment study," *Drones*, vol. 10, no. 4, 2026, doi: 10.3390/drones10040255.
[21] S. Khan *et al.*, "Enhanced spatial stream of two-stream network using optical flow for human action recognition," *Appl. Sci.*, vol. 13, no. 14, 2023, doi: 10.3390/app13148003.
[22] T. Ahmad, J. Wu, H. S. Alwageed, F. Khan, J. Khan, and Y. Lee, "Human activity recognition based on deep-temporal learning using convolution neural networks features and bidirectional gated recurrent unit with features selection," *IEEE Access*, vol. 11, pp.

33148–33159, 2023, doi: 10.1109/ACCESS.2023.3263155.
[23] A. Khare, A. Kushwaha, and O. Prakash, "Human activity recognition in a realistic and multiview environment based on two-dimensional convolutional neural network," *J. Artif. Intell. Technol.*, vol. 3, no. 3, pp. 100–107, 2023, doi: 10.37965/jait.2023.0163.

Matrix: Jurnal Manajemen Teknologi dan Informatika
Volume 16 Issue 2 Year 2026 Pages 86-100
DOI: http://dx.doi.org/10.31940/matrix.v16i2.86-100
URL: http://ojs2.pnb.ac.id/index.php/matrix

Daily USD-IDR exchange rate prediction using Long Short-Term Memory (LSTM) with macroeconomic indicators and recursive multi-step prediction

Ni Luh Gede Ina Ari Richardi [1]*, Putu Indah Ciptayani [2], I Putu Bagus Arya Pradnyana [3]

[1,2,3] *Software Engineering Technology Study Program, Politeknik Negeri Bali, Indonesia*

*Corresponding Author: *luhgedeiar@gmail.com*

Abstract: The foreign exchange market's high volatility and non-linear dynamics pose significant challenges for accurate currency price forecasting. This study develops a predictive model for the daily USD-IDR exchange rate using Long Short-Term Memory (LSTM) method integrated with macroeconomic indicators. Historical price data, global oil prices, interest rates, and the US Dollar Index (DXY) were collected by fetching from Yahoo Finance and FRED. A comprehensive experimental design comprising 432 configurations was evaluated across varying data periods, train-validation-test splits, feature combinations, and hyperparameters. Results indicate that a 10-year historical dataset yields the most stable and accurate performance, achieving a test MAPE of 0.52% and R^2 of 0.904. The integration of macroeconomic variables improved predictive capability, though the impact of DXY was found to be conditional on data split ratios and hyperparameter settings. The optimal model was deployed as an interactive web application using Streamlit, enabling users to forecast USD-IDR rates up to seven days ahead via recursive multi-step forecasting. The proposed system provides a practical, accessible tool for retail traders and financial analysts to support informed decision-making in volatile forex markets

Keywords: Long Short-Term Memory, USD-IDR, deep learning, foreign exchange.

History Article: Submitted 25 May 2026 | Revised 18 July 2026 | Accepted 30 July 2026

How to Cite: N. L. G. I. A. Richardi, P. I. Ciptayani, and I. P. B. A. Pradnyana, "*Daily USD-IDR exchange rate prediction using Long Short-Term Memory (LSTM) with macroeconomic indicators and recursive multi-step prediction,*" *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 16, no. 2, pp. 86–100, 2026, doi: 10.31940/matrix.v16i2.86-100

Introduction

Foreign Exchange (forex) is the largest currency trading market in the world [1]. According to the Bank of International Settlements, the average daily trading volume in forex market reached approximately $9.6 trillion (Rp161.81 quadrillion) in April 2025 [2]. In general, forex refers to the mechanism for trading one country's currency against another country's currency, such as the Euro (EUR) against the United States Dollar (USD), the United States Dollar (USD) against the Indonesian Rupiah (IDR), and others [3], [4].The forex market is uniquely characterized by its global operation across four different time zones for 24 hours a day, 5 days a week [5][6]. Initially, this market mostly dominated by large institutions, however it has recently attracted increasing interest from individual investors due to its high liquidity, flexibility and potential for substantial profits within a short period of time [4], [5], [7].

Despite the significant profit opportunities, forex also involves substantial risks. Forex is characterized by high volatility, irregularity, non-linearity, and fluctuations, making it a challenging for novice traders [6]. Furthermore, exchanges rate fluctuation in forex are influenced by various macroeconomic factors such as interest rates, global oil prices, and others economic indicator, which require more complex analysis than basic technical analysis [8], [9].These complexities often make it difficult for novice traders to accurately interpret and predict market movements, thereby increasing the likelihood of erroneous trading decisions and financial losses [6].

To mitigate these risks, developing a currency price prediction system using the LSTM (Long Short-Term Memory) method, which integrates historical data with macroeconomic indicators, can be an effective solution. The LSTM is selected due its capability of process long

time series data, retain long-term patterns, capturing non-linear relationships in the data, and integrate macroeconomic variables into the model learning process [8], [9].

Several previous studies have demonstrated the effectiveness of LSTM in predicting currency exchange rates. For instance, Wijaya et al., successfully predicted the EUR-USD and GBP-USD exchange rates using the LTSM method with high accuracy [4]. Similarly, Poernomo achieved accurate prediction of the Indonesian Rupiah against various Southeast Asia currencies using LSTM with a high level of accuracy (MAPE < 10%) [8]. In a relates study, Dita Ardriani et al., successfully predicted the USD-IDR exchange rate using daily historical data with LSTM, but acknowledged the model's limitations in handling volatility due to unexpected events [9]. However, these studies primarily focused on model experimentation and predictive evaluations without developing an integrated system that combines historical data with macroeconomic data in a single prediction model. To address this gap, this study proposes the development of a website-based exchange rate prediction system that integrate historical data with macroeconomic data, aiming to enhance prediction accuracy while ensuring user-friendly accessibility.

The development of this web-based exchange rate prediction system using LSTM, incorporating both historical and macroeconomic data, is expected to provide valuable insights into currency market movements, particularly for novice traders. Moreover, the web-based implementation ensures broad accessibility, facilitating widespread adoption among users. This research also aims to pave the way for future advancements in integrating predictive technologies with broader digital financial solutions.

Methodology

This study employs a quantitative experimental approach to determine the optimal LSTM configuration for daily USD-IDR exchange rate prediction. The research pipeline consists of six main stages: data collection, data preprocessing, experimental setup, model training, model evaluation, and web implementation, as illustrated in Figure 1.

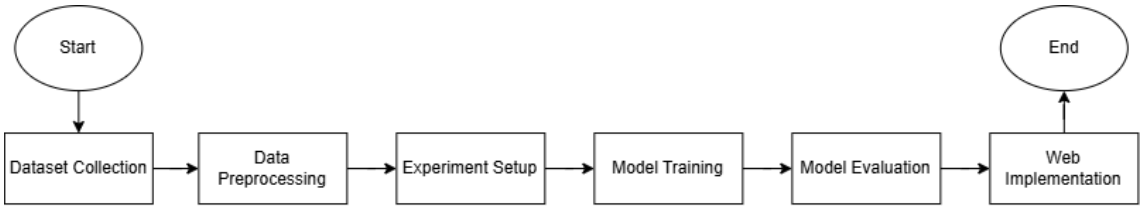


Figure 1. Research workflow

Data Collection

Daily frequency data were retrieved from Yahoo Finance and the Federal Reserve Economic Data (FRED). Four datasets were utilized: USD-IDR closing price (primary target), global oil prices, interest rates, and the US Dollar Index (DXY). Data were collected using Python libraries across three time periods: short period (2020–2025), medium period (2015–2025), and long period (2010–2025).

Data Preprocessing

This stage is conducted to ensure that the data is ready for use in the model development process. The collected raw data undergoes several procedures as follows:

Resampling

The resampling stage is conducted to ensure that all utilized data shares the same frequency [10]. In this study, the interest rate data undergoes resampling, where it is converted from a monthly frequency to a daily frequency to align with the other datasets. This process generates new data entries for each trading day within a month, and the newly created rows are filled using the forward fill (ffill) method. Consequently, the same monthly interest rate value is propagated to every trading day until the next monthly observation becomes available. For example, in the short-period dataset, the interest rate data originally consisted of 72 monthly observations. After the resampling process, it expanded to 2,192 daily observations while

preserving the same monthly interest rate values across all trading days within each month. As shown in Figure 2 (a), the excerpts interest rate data before resampling is presented, while Figure 2 (b) shows the excerpts interest rate data after resampling.

	DATE	RATE
67	1 Aug 2025	6.07
68	1 Sep 2025	5.76
69	1 Oct 2025	5.53
70	1 Nov 2025	5.47
71	1 Dec 2025	5.46

(a)

	Date	RATE
2187	27 Dec 2025	5.46
2188	28 Dec 2025	5.46
2189	29 Dec 2025	5.46
2190	30 Dec 2025	5.46
2191	31 Dec 2025	5.46

(b)

Figure 2. (a) Interest rate data before resampling (b) Interest rate data after resampling

Check and Fill Missing Value

This stage is conducted to verify that all utilized data is complete and maintains chronological order. In this study, the forward fill and backward fill methods are employed for data imputation, as these techniques help preserve the continuity of the time-series data [11], [12].

Check and Drop Duplicate Entries

This stage is conducted to ensure that there no duplicate records exist to maintaining data quality. Duplicate records may arise when merging data from multiple sources [13]. In this study, duplication checks are conducted on the date column to guarantee data validity.

Data Splitting

This stage is conducted to divide the dataset into several parts, each of which will be used for a specific purpose. In this study, the data is divided into three parts: training set, validation set, and testing set. The training set is used to train the model, the validation set is employed for hyperparameter tuning and performance model evaluation, and the testing set is reserved for assessing the model's final performance. Dataset partitioning is crucial to ensure the model operates optimally on unseen data and avoids overfitting to the training data [14].

Feature Scaling

At this stage, the data is transformed from its original range to a standardized scale between 0 and 1. The Min-Max Normalization method is applied for feature scaling in this study. This technique ensures that all attribute values are constrained within a consistent range, enabling fair comparison and unbiased analysis [15]. The scaling process was performed only on the training data to prevent data leakage. After the minimum and maximum parameters were obtained from the training data, the transformation process was applied to the training, validation, and testing datasets. As shown in Figure 3 (a), the data before the scaling process, while Figure 3 (b) shows the data after the scaling process.

Date	USD_IDR	OIL_PRICE	RATE
2015-01-01	12390.000000	56.419998	7.122027
2015-01-02	12390.000000	56.419998	7.122027
2015-01-05	12480.000000	53.110001	7.122027
2015-01-06	12625.000000	51.099998	7.122027
2015-01-07	12635.000000	51.150002	7.122027

(a)

Scaling selesai (MinMax 0-1)

Date	USD_IDR	OIL_PRICE	RATE
2015-01-01	0.000000	0.341371	0.680309
2015-01-02	0.000000	0.341371	0.680309
2015-01-05	0.021872	0.310907	0.680309
2015-01-06	0.057111	0.292407	0.680309
2015-01-07	0.059541	0.292867	0.680309

(b)

Figure 3. (a) Data before scaling (b) Data after scaling

Create Sequence

Sequence creation in this study is performed using the sliding window method, where time-series data is extracted according to the time step parameter value to predict the price for the subsequent day [16]. This approach enables the model to effectively learn temporal patterns and trends within the data.

Experiment Setup

To determine the best model configuration, experiments involving several parameters and hyperparameters were required. The search for the optimal model configuration was conducted using the Grid Search approach. This method systematically evaluates all possible combinations of predefined hyperparameter values. In this research, a total of 432 experimental combinations were conducted by combining various configurations of data split ratios, historical periods, macroeconomic feature sets, and hyperparameters such ad learning rate, units, time step dan batch size, as detailed in Table 1.

Table 1. Experiment configuration

Parameter	Value	Total	Description
Data Splitting			
Train:Val:Test	Scenario 1: 70%:15%:15% Scenario 2: 80%:10%:10%	2	Data splitting scenarios
Dataset Configuration			
Time Period	5 years, 10 years, 15 years	3	Historical data length
Feature Selection			
Macroeconomic Variable	Scenario 1: Oil Price & Interest Rate Scenario 2: Oil Price, Rate & DXY	2	External feature combinations
Hyperparameters			
Learning Rate	0.001, 0.005, 0.01	3	Adam optimizer learning rate
Units	50, 100	2	Hidden state in LSTM layer
Time Step	30, 60, 90	3	Sliding window size (days)
Batch Size	32, 64	2	Training Batch Size

Model Training

The model was developed using the LSTM which is designed to overcome the problem of vanishing gradient and exploding gradient in RNN by using memory cells regulated by three gates: input gate, forget gate and output gate which control the entry and exit of information from the memory cell [17]. The architecture of the model built with consisting of an LSTM layer as the input layer, followed by a dropout layer to prevent overfitting by randomly deactivating a portion of the neurons, and two consecutive dense layers: a fully connected layer with 32 neurons and a layer without an activation function containing 1 neuron that acts as the output layer.

Each experiment was validated using 3-fold time series cross-validation with Time Series Split method. This method preserves the chronological order of the data by applying an expanding window strategy, where the training data gradually increases in each fold, while validation is conducted on the subsequent unseen time period. This approach was chosen to prevent information leakage from future data and to produce a more realistic evaluation of model performance on time-series data. After identifying the optimal configuration, the model was retrained on separate training and validation sets and evaluated using some evaluation matrix on the unseen test set.

Model Evaluation

After training the model with the optimal configuration, evaluation was performed using data from the test set, which the model had not previously encountered. In this study, two evaluation metrics were employed to assess model performance: Mean Absolute Percentage Error (MAPE) and R-Squared (R^2). MAPE was used to quantify the average prediction error of the model, as formulated in Equation (1) [18]. Meanwhile, R-Squared was utilized to measure the proportion of data variance explained by the model and to determine whether the model successfully captures the underlying patterns and trends in exchange rate movements, as expressed in Equation (2) [6], [9]. The best-performing model was selected based on the lowest MAPE and the highest R^2 values.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right| x\ 100\% \quad (1)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_t)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \quad (2)$$

Web Implementation

After obtaining the best model configuration, the final stage was to implement the model into an interactive website that can be widely used by users. In this study, the user interface was developed using the Streamlit framework, which accepts date inputs from users and generates exchange rate predictions along with graphical visualizations of price movements.

The Recursive Multi-step Forecasting method was applied in the system and operates iteratively, where the prediction result from the first day is reused as an input feature to predict the following day [19]. This approach enables users to generate forecasts for several days ahead. Considering the error accumulation characteristics of recursive forecasting, the prediction scope in this system is limited to a maximum horizon of 7 days ahead in order to maintain the reliability and stability of prediction accuracy.

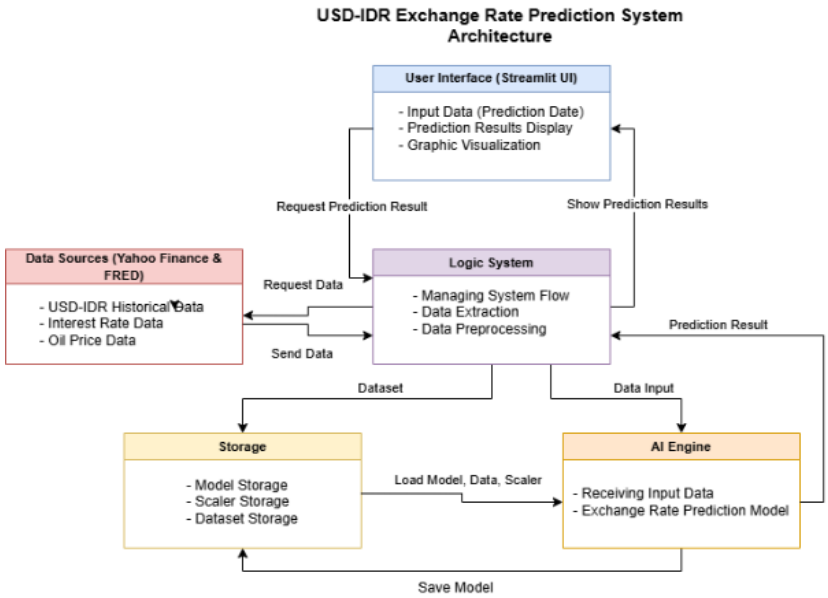


Figure 4. System architecture

Figure 4 illustrates the architecture of the system. The system consists of five main components. The first component is the user interface, which functions as the medium through which users interact with the system. The second component is the system logic, which manages the entire system workflow, starting from data retrieval from external sources, data extraction,

data preprocessing, data integration, data storage, sending user inputs to be processed by the AI model, and forwarding the prediction results to be displayed to users.

The third component is the data source module, which provides the data required by the system. The fourth component is data storage, which serves as a repository for important system components such as prediction models, datasets, and scalers. Finally, the AI engine acts as the core module of the prediction system, where the prediction process is carried out. The model used contains parameters learned during the training process and is utilized to predict exchange rate values based on the prepared dataset.

Results and Discussions

Results

Model Experiment

After conducting 432 experiments validated using 3-fold time series cross-validation using Time Series Split method by combining variations in data periods, data split ratios, macroeconomic feature configurations, and hyperparameter combinations, the results showed that using 10 years of data provided the best and most stable performance across both data-splitting scenarios. As summarized in Table 2, the experiments with IDs EXP-3 and EXP-4 under the 70:15:15 data split scenario produced the best testing performance, with test MAPE values of 0.52% and 0.48%, and test R^2 values reaching 0.904 and 0.906.

Table 2. Experiment result

Best Exp ID	Data Period	Macro	Best Hyperparameter				Validation		Test	
			Units	Learning Rate	Time Step	Batch Size	MAPE	R2	MAPE	R2
Data Split Scenario 1: 70:15:15										
EXP-1	5 Year	2 (Oil & Rate)	100	0.005	30	64	0.43 ± 0.026	0.872 ± 0.030	0.40	0.735
EXP-2		3 (Oil, Rate & DXY)	50	0.005	30	64	0.54 ± 0.08	0.840 ± 0.01	1.80	-1.12
EXP-3	10 Year	2	100	0.001	30	64	0.69 ± 0.23	0.907 ± 0.07	0.52	0.904
EXP-4		3	50	0.005	30	32	0.64 ± 0.11	0.923 ± 0.03	0.48	0.906
EXP-5	15 Year	2	100	0.001	90	64	6.28 ± 5.38	-1.13 ± 1.55	5.68	-4.546
EXP-6		3	100	0.001	60	64	7.32 ± 5.36	-1.55 ± 1.30	2.14	0.175
Data Split Scenario 2: 80:10:10										
EXP-7	5 Year	2	100	0.01	30	64	0.51 ± 0.09	0.888 ± 0.022	0.40	0.715
EXP-8		3	50	0.01	30	64	1.03 ± 0.57	0.571 ± 0.381	0.84	0.258
EXP-9	10 Year	2	50	0.005	60	32	0.56 ± 0.13	0.926 ± 0.034	0.54	0.588

EXP-10	15 Year	3	50	0.001	60	64	0.58 ± 0.07	0.908 ± 0.049	0.71	0.538
EXP-11		2	50	0.001	30	64	7.20 ± 5.81	-1.389 ± 2.010	4.67	-2.63
EXP-12		3	50	0.001	90	64	5.25 ± 3.38	-0.551 ± 1.191	3.94	-1.737

Based on Table 2, the comparison of data period lengths shows a significant impact, particularly on the R^2 values. The dataset with a 10-year period (2015–2025) produced the best and most stable accuracy, whereas the dataset with a 5-year period (2020–2025) achieved good test MAPE and R^2 values under several configurations but was still unable to effectively capture the underlying patterns and trends in the data. Meanwhile, the dataset with a 15-year period (2010–2025) produced the worst performance, with the test MAPE increasing to as high as 5.68% and negative R^2 values.

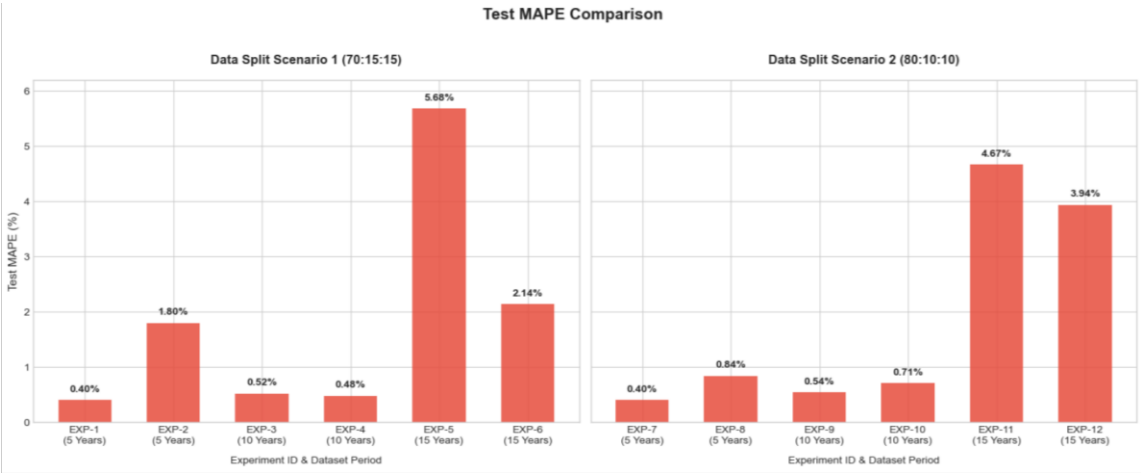


Figure 5. MAPE comparison chart

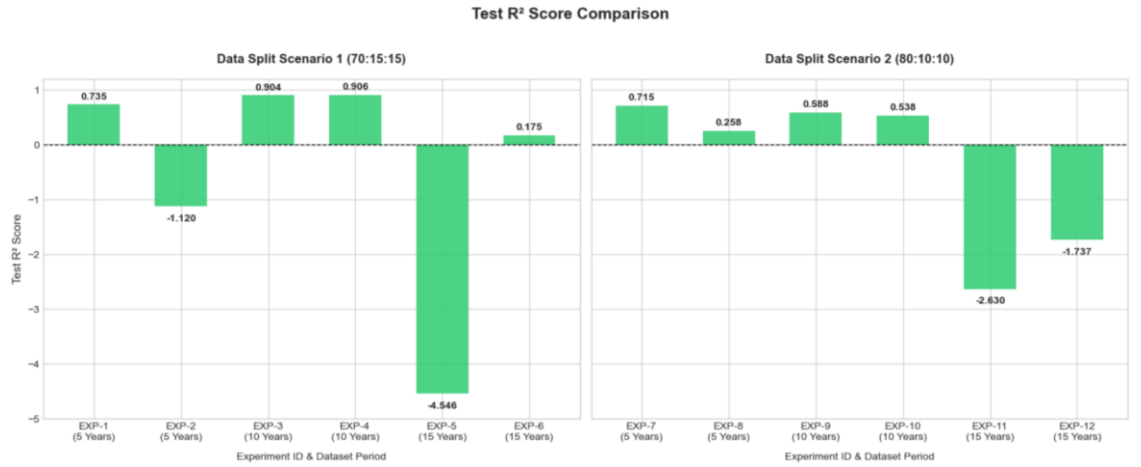


Figure 6. R^2 comparison chart

The addition of the Dollar Index (DXY) as a third macroeconomic feature did not consistently improve test accuracy. In the first data-splitting scenario, the integration of the DXY feature in experiment ID EXP-4 successfully reduced the test MAPE from 0.52% to 0.48% and increased the R^2 value. However, in the second scenario, the configuration with only two macroeconomic features resulted in a lower test MAPE of 0.54% compared to 0.71% for the

configuration with three macroeconomic features. This indicates that the contribution of the DXY feature is conditional and depends on the combination of hyperparameters and the data split ratio.

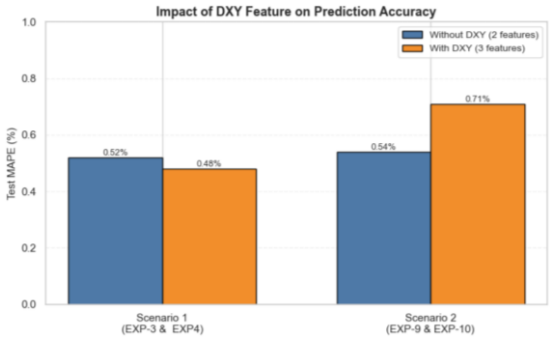


Figure 7. DXY impact chart

There were also significant differences between the validation and test accuracy results in several configurations. Although some experiments achieved very low validation MAPE values (<1%) with small standard deviations, the performance on the test data declined considerably, particularly in the 15-year dataset and several configurations of the 5-year dataset. The negative test R^2 values in EXP-2, EXP-5, EXP-11, and EXP-12 indicate possible overfitting to the validation data or the model's inability to generalize patterns to entirely new time periods.

Based on the tuning results, learning rates of 0.001 and 0.005 showed the most stable performance with the lowest errors. A time step of 30 days was sufficiently effective in capturing daily patterns, while longer time steps of 60–90 days tended to increase validation error variance. LSTM unit sizes of 50 and 100 produced comparable performance, whereas batch sizes of 32–64 did not show statistically significant differences. The parameter combination that most consistently produced high accuracy consisted of a 10-year data period, learning rates of 0.001–0.005, and a 30-day time step.

To evaluate the specific impact of macroeconomic indicators on prediction accuracy, a baseline experiment was conducted excluding macroeconomic data. This comparison was applied to the best-performing models, EXP-3 and EXP-4, utilizing identical datasets, data split configurations, and architectural workflows. As summarized in Table 3, the integration of macroeconomic variables substantially enhances the forecasting accuracy.

Table 3 Comparison of model configuration with and without macroeconomic features.

Feature Configuration	Test MAPE (%)	Test R^2
Historical USD/IDR Only (Baseline)	1.50	0.531
Historical USD/IDR + Interest Rate + Oil Price (EXP-3)	0.52	0.904
Historical USD/IDR + Interest Rate + Oil Price + DXY (EXP-4)	0.48	0.906

Furthermore, to evaluate the model practical viability for extended forecasting horizons, the trained models were subjected to a Recursive Multi-step Forecasting evaluation. Since the LSTM models were inherently trained on a single-step-ahead prediction framework using the sliding window approach, generating a multi-step forecast required the system to iteratively feed its own prior predictions back into the feature matrix as input for subsequent days [19].

Testing was conducted across various forecasting horizons up to a maximum of seven days ahead. The limitation of selecting prediction dates to a maximum of seven days ahead is due to the recursive multi-step forecasting mechanism, where the model predicts the exchange rate for the first day using actual historical data. However, to predict the second day, the model must use its own prediction result from the first day as an input feature, treating it as actual data. As a result, any prediction error generated in the first step will propagate through subsequent iterations and gradually increase over time [20]. Therefore, a seven-day limit was established as

the maximum forecasting horizon in order to maintain the accuracy and stability of the LSTM model.

Recursive evaluation was performed on the two best models (EXP-3 and EXP-4), where the model with the best performance was selected and implemented in the interactive web application, ensuring reliable short-term forecasting within the limited 7-day forecasting window. Based on the recursive evaluation of the models presented in Table 5 and Table 5, the model with experiment ID EXP-3 was selected to be implemented in the website.

Table 4. EXP-3 model recursive evaluation result

Day	MAPE	R2
+1	0.5020	0.9023
+2	0.5910	0.8768
+3	0.6837	0.8462
+4	0.7859	0.8070
+5	0.8763	0.7628
+6	0.9669	0.7157
+7	1.0538	0.6661
AVG	**0.7799**	**0.7967**

Table 5. EXP-4 model recursive evaluation result

Day	MAPE	R2
+1	0.6933	0.7733
+2	0.9587	0.5601
+3	1.2144	0.2824
+4	1.4705	-0.0454
+5	1.6905	-0.4059
+6	1.9029	-0.7829
+7	2.0926	-11.728
AVG	**1.4318**	**0.1130**

Web Implementation

The best model identified was implemented into a website developed using the Streamlit framework. The developed system implements the Recursive Multi-Step Forecasting method, which is used to predict the USD-IDR exchange rate for several days ahead in a recursive manner. In this study, the forecasting horizon was limited to a maximum of 7 days ahead. The process is carried out by using the prediction result from the first day as a new input feature to predict the following day. The system consists of two main pages: the homepage and the prediction results page.

The homepage is the first page viewed by users when accessing the website. On this page, the system provides a date input feature that can be filled in by users, along with an action button labeled *"Mulai Prediksi"* to run the prediction model. In addition, the page also displays a historical data table containing USD-IDR exchange rate information for the past 7 days, as well as macroeconomic indicators including global oil prices and interest rates, as shown in Figure 8.

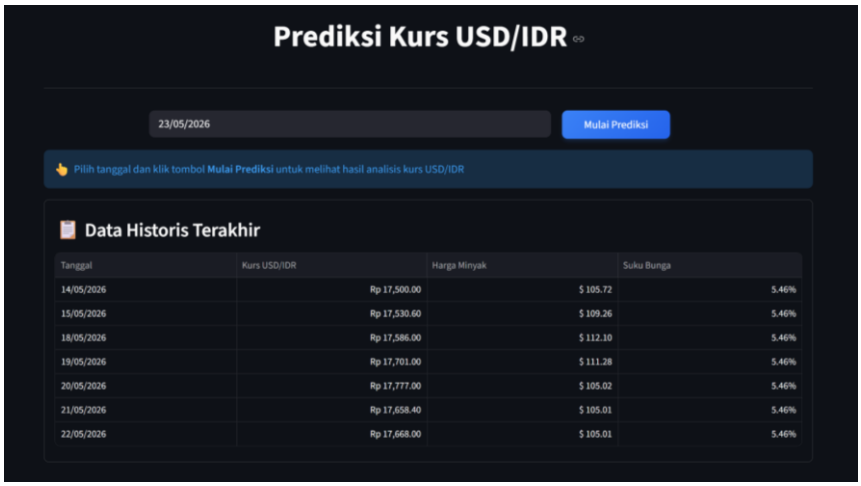


Figure 8. Homepage of the USD-IDR exchange rate prediction system

The second page is the prediction results page. After users select a date and click the "Start Prediction" button, the system processes the data and displays the results in the form of a summary block consisting of four sections, which provide information regarding the predicted exchange rate, model accuracy, the latest global oil price, and the latest interest rate as shown in Figure 9. If users select a prediction horizon of more than one day, the system will automatically display the prediction results in tabular form as shown in Figure 10. In addition, the page also includes an exchange rate chart that displays the exchange rate values from the past 7 days along with the prediction results to visualize the direction of currency movements.

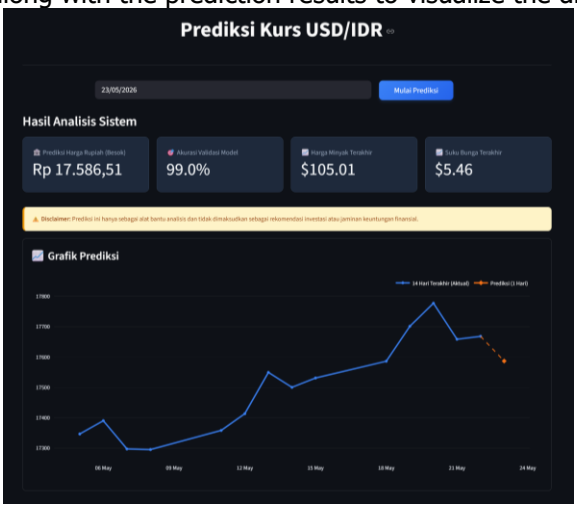


Figure 9. Single prediction result page

Figure 10. Multiple prediction result page

Discussions

The comparison of data period lengths shows a significant impact, particularly on the R^2 value. The dataset with a 5-year period (2020–2025) produced good test accuracy in terms of MAPE and R^2 under several configurations, but it was not sufficient to capture the underlying patterns and trends in the data. This indicates that the model experienced underfitting due to the relatively short data period. This finding is consistent with previous studies conducted by Poernomo [8] and Wijaya et al. [4], which showed that deep learning models require sufficiently long historical data to effectively capture non-linear market dynamics.

The dataset with a 15-year period (2010–2025) produced the worst accuracy, with the test MAPE increasing to 5.68% and a negative R^2 value. This demonstrates that a longer data period does not always improve prediction accuracy [21]. Data from 15 years ago may no longer be relevant to current exchange rate volatility characteristics, where including overly outdated data contributes minimally or may even negatively affect model performance [22].

The dataset with a 10-year period (2015–2025) produced the best and most stable accuracy because this period provides an optimal balance between volatile data variations caused by unexpected events such as the COVID-19 pandemic and economic policy changes, as well as more stable data and sufficient data quantity. This is important because the model must be able to capture patterns and trends across various market conditions, enabling it to achieve a balanced learning process and avoid overfitting to a specific condition [23], [24].

Compared to the baseline experiment that relied solely on historical data, the integration of macroeconomic factors substantially improved the model's performance, reducing the test MAPE from 1.50% to 0.52% and increasing the R2 score from 0.531 to 0.904. Furthermore, this approach noticeably outperforms existing models in the literature, such as the study conducted by Pahlevi et al. [1], which reported a MAPE value of 2.5%, and the forecasting model by Wahyudi et al. [25], which achieved a MAPE value of 0.81%. This internal head-to-head comparison empirically demonstrates that embedding macroeconomic indicators provides critical external shock signals that historical data alone cannot fully resolve.

Figure 11 Heatmap correlation historical data and DXY macro indicator

Based on Figure 11, the addition of the US Dollar Index (DXY) did not consistently improve accuracy, this is because the DXY feature has a very high correlation value of 0.9 with the USD-IDR as lag feature. This indicating severe multicollinearity, which leads to information redundancy, meaning that DXY fails to provide significant novel information to the dataset [26], [27]. Therefore, feature selection remains an important factor in exchange rate forecasting models. Consequently, this redundancy underscores that rigorous feature selection remains a critical factor in exchange rate forecasting models.

This phenomenon directly explains the significant decline in accuracy observed in the model with ID EXP-4. During the recursive forecasting phase, the model encountered a critical operational mismatch, the USD-IDR values were updated using the system's predicted values, while the DXY values were filled using the forward fill method. This created a mismatch, where the USD-IDR exchange rate moved dynamically while the DXY remained static, thereby triggering substantial increases in prediction error.

Furthermore, the recursive multi-step forecasting evaluation further showed that prediction accuracy gradually decreased as the forecasting horizon increased. This occurred because prediction errors from previous steps were reused as inputs for subsequent predictions, causing error accumulation over time. Nevertheless, the EXP-3 model maintained relatively stable performance within the 7-day forecasting horizon as shown in Table 4, making it suitable for short-term exchange rate forecasting applications. The trend of increasing MAPE values and decreasing R^2 scores across the forecasting horizon can also be visually observed in Figure 12 and Figure 13.

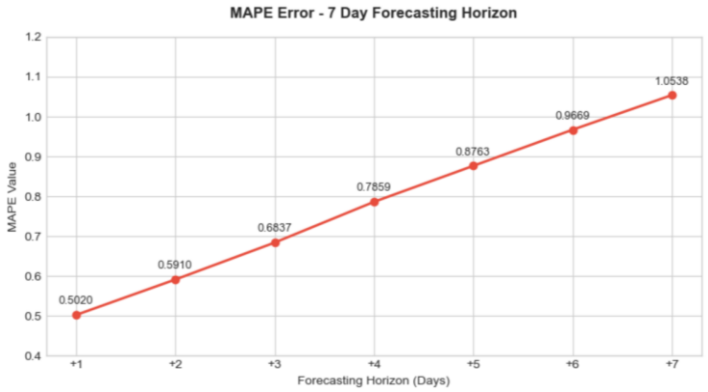


Figure 12. EXP-3 MAPE error chart

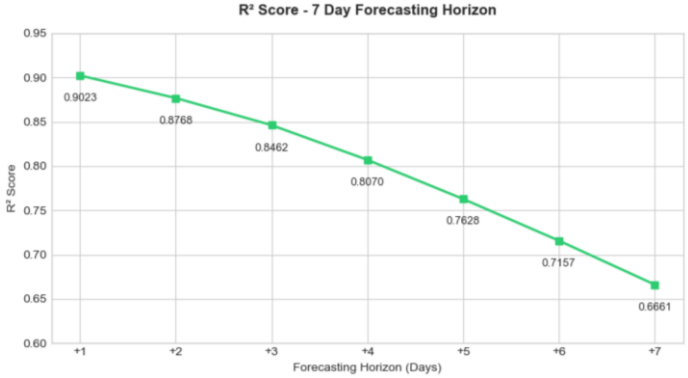


Figure 13. EXP-3 R^2 error chart

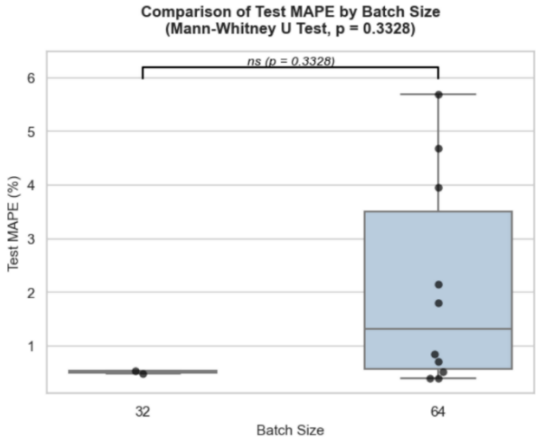


Figure 14 MAPE comparison by batch size

Based on the Mann-Whitney test shown in Figure 14 the comparison of test MAPE values between the batch size 32 and batch size 64 groups yielded a p-value of 0.3328. In standard statistical methodology, a p-value greater than the established significance level $\alpha = 0.05$. This outcome empirically demonstrates that the performance difference between the two batch sizes is not statistically significant. Research results also demonstrate that the best-performing models used different batch sizes, specifically the EXP-3 model with batch size 64: MAPE = 0.52%, R^2 = 0.904 and the EXP-4 model with batch size 32: MAPE = 0.48%, R^2 = 0.906), showing a difference of only 0.04% in MAPE and 0.002 in R^2.

The implementation of the model into an interactive web application using Streamlit demonstrates the practical applicability of the proposed system. The developed application enables users to perform short-term USD-IDR exchange rate forecasting through a user-friendly interface, making the system more accessible for traders and financial analysts. The developed application enables users to perform short-term USD-IDR exchange rate forecasting through a

user-friendly interface, making the system more accessible for traders and financial analysts. In a real-world operational context, this deployment eliminates the need for users to possess advanced programming skills or perform manual data preprocessing. By utilizing a modular architecture, the system automatically pulls data, cleans it, and executes a recursive multi-step forecasting method to provide a restricted 7-day prediction horizon.

This specific 1-week window offers high specificity for short-term risk management, allowing financial officers to make data-driven decisions during high market volatility. However, for practical deployment, users must consider that the model relies heavily on historical patterns and macroeconomic indicators. Therefore, in real-world usage, it is highly recommended to utilize this application as a decision-support tool alongside human expert oversight to mitigate risks from sudden, unpredictable global economic shocks (black swan events).

Conclusion

This study successfully developed and evaluated an LSTM-based forecasting system for daily USD-IDR exchange rates by integrating historical price data with macroeconomic indicators. Through a comprehensive experimental design of 432 configurations, the optimal and stable model was obtained using a 10-year dataset, two macroeconomic features, and hyperparameters consisting of 100 units, a learning rate of 0.001, a 30-day time step, and a batch size of 64, which were determined through a hyperparameter tuning process. While the baseline model relying on historical data yielded a lower accuracy with a test MAPE of 1.50% and an R2 of 0.531, the proposed macro-integrated configuration significantly outperformed it, achieving an optimal test MAPE of 0.52% and an R2 of 0.904. The findings confirm that integrating external economic variables significantly improves short-term currency prediction accuracy compared to models using only historical data. The implementation of the model into an interactive web application using Streamlit provides an accessible data-driven forecasting tool for users.

Although the system demonstrated strong performance under normal and moderately volatile market conditions, several limitations remain, including the potential accumulation of errors in recursive multi-step forecasting, vulnerability to structural changes within extended historical windows, and the absence of real-time sentiment data or policy announcements. Future research should investigate dynamic retraining mechanisms and hybrid deep learning architectures to further improve the model's adaptability.

Acknowledgement

The author gratefully acknowledges the support and contributions from all parties involved in the completion of this article. Particular thanks are expressed to Putu Indah Ciptayani, S.Kom.,M.Cs. and I Putu Bagus Arya Pradnyana, S.Kom.,M.Kom, for their guidance and input throughout the research process.

References

[1] M. R. Pahlevi, "Prediksi Harga Forex Menggunakan Algoritma Long Short-Term Memory," *JNANALOKA*, pp. 69–76, Sep. 2023, doi: 10.36802/jnanaloka.2022.v3-no2-69-76.

[2] Bank of International Settlements, "OTC foreign exchange turnover in April 2025," Bank of International Settlements. Accessed: Jan. 04, 2026. [Online]. Available: https://www.bis.org/statistics/rpfx25_fx.htm

[3] M. Edi, E. Utami, and A. Yaqin, "Prediksi Harga pada Trading Forex Pair USDCHF Menggunakan Regresi Linear," *Jurnal Manajemen Informatika (JAMIKA)*, vol. 13, no. 2, pp. 109–119, Sep. 2023, doi: 10.34010/jamika.v13i2.9826.

[4] A. J. Wijaya, W. Swastika, and O. H. Kelana, "PREDIKSI HARGA FOREIGN EXCHANGE MATA UANG EUR/USD DAN GPB/USD MENGGUNAKAN LONG SHORT-TERM MEMORY," *SAINSBERTEK Jurnal Ilmiah Sains & Teknologi*, vol. 2, Sep. 2021.

[5] Sudarmadji, "Trading Forex Online Individu: Masalah Syariah," *Labs: Jurnal Bisnis dan Manajemen*, vol. 28, 2023.

[6] M. R. Pahlevi, K. Kusrini, and T. Hidayat, "Comparison of LSTM and GRU Models for Forex Prediction," *Sinkron : Jurnal dan Penelitian Teknik Informatika*, vol. 8, no. 4, pp. 2254–2263, Oct. 2023, doi: 10.33395/sinkron.v8i4.12709.

[7] M. Tjitrayudha and Kosasih, "UJI EFEKTIVITAS METODE BREAKOUT SUPPORT RESISTANCE UNTUK PROBABILITAS PADA FOREIGN EXCHANGE MARKET," *MANAJEMEN : JURNAL EKONOMI USI*, vol. 6, no. 2, 2024.

[8] A. Poernomo, "RUPIAH EXCHANGE RATE PREDICTION WITH LONG SHORT-TERM MEMORY ALGORITHM," *Syntax Literate: Jurnal Ilmiah Indonesia*, vol. 10, no. 1, Jan. 2025.

[9] N. N. Dita Ardriani, P. Sugiartawan, and G. A. Santiago, "Deep Learning Approach for USD to IDR Forecasting with LSTM," *JSIKTI: Jurnal Sistem Informasi dan Komputer Terapan Indonesia*, vol. 8, no. 1, pp. 91–100, Jun. 2025.

[10] D. Safira, I. A. Sobari, and S. Rahayu, "PREDIKSI HARGA BBCA MENGGUNAKAN LSTM MULTIVARIAT DENGAN FITUR BI-RATE SEBAGAI INDIKATOR EKSTERNAL," *Information System Journal*, vol. 8, no. 02, pp. 119–128, Dec. 2025, doi: 10.24076/infosjournal.2025v8i02.2372.

[11] F. Kamalov and H. Sulieman, "Time series signal recovery methods: comparative study," Oct. 2021, doi: https://doi.org/10.48550/arXiv.2110.12631.

[12] J. Y. Lee *et al.*, "Comparison of Models for Missing Data Imputation in PM-2.5 Measurement Data," *Atmosphere (Basel).*, vol. 16, no. 4, Apr. 2025, doi: 10.3390/atmos16040438.

[13] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[14] H. Allam, L. Makubvure, B. Gyamfi, K. N. Graham, and K. Akinwolere, "Text Classification: How Machine Learning Is Revolutionizing Text Categorization," *Information (Switzerland)*, vol. 16, no. 2, Feb. 2025, doi: 10.3390/info16020130.

[15] A. Pranolo *et al.*, "Enhanced Multivariate Time Series Analysis Using LSTM: A Comparative Study of Min-Max and Z-Score Normalization Techniques," *ILKOM Jurnal Ilmiah*, vol. 16, no. 2, pp. 210–220, Aug. 2024, doi: 10.33096/ilkom.v16i2.2333.210-220.

[16] U. Brawijaya, I. R. Nashrullah, E. R. Widasari, and B. H. Prasetio, "Implementasi Metode Sliding Window Untuk Peningkatan Perekaman Real-Time Pada Sistem Deteksi Kesadaran Berbasis Sensor Electroencephalography Satu Kanal," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 10, no. 3, pp. 2548–964, 2026, [Online]. Available: http://j-ptiik.ub.ac.id

[17] I. D. Mienye and T. G. Swart, "A Comprehensive Review of Deep Learning: Architectures, Recent Advances, and Applications," *Information (Switzerland)*, vol. 15, no. 12, Dec. 2024, doi: 10.3390/info15120755.

[18] M. K. Najib and S. Nurdiati, "Pemodelan Deret Waktu Menggunakan Non-linear Autoregressive Neural Network: Studi Kasus Prediksi Harga Saham Mandiri," *Jambura Journal of Mathematics*, vol. 7, no. 2, pp. 213–220, Aug. 2025, doi: 10.37905/jjom.v7i2.33397.

[19] H. T. A. Simanjuntak, A. Lumbanraja, G. Samosir, and Regita, "Prediksi Single-Step dan Multi-Step Data Cuaca Menggunakan Model Long Short-Term Memory dan Sarima," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 12, no. 2, pp. 399–410, Apr. 2025, doi: 10.25126/jtiik.2025129444.

[20] N. Fitriyati, M. Liebenlito, and S. Hasna Tsabitah, "Pengaruh Hiperparameter dan Variabel Eksogen pada Prediksi Multi-Langkah Kecepatan Angin menggunakan LightGBM," *JURNAL FOURIER*, vol. 15, no. 1, pp. 1–16, Apr. 2026, doi: 10.14421/fourier.2026.151.1-16.

[21] A. Lazcano, P. Hidalgo, and J. E. Sandubete, "Walking Back the Data Quantity Assumption to Improve Time Series Prediction in Deep Learning," *Applied Sciences (Switzerland)*, vol. 14, no. 23, Dec. 2024, doi: 10.3390/app142311081.

[22] Q. Yu and B. Tolson, "How much historical data do we need? The role of data recency and training period length in LSTM-based rainfall-runoff modeling," *J. Hydrol. (Amst).*, vol. 668, Apr. 2026, doi: 10.1016/j.jhydrol.2026.135046.

[23] K. Jongseon, K. Hyungjoon, K. Hyun Gi, L. Dongjun, and Y. Sungroh, "A comprehensive survey of deep learning for time series forecasting: architectural diversity and open

challenges," *Artif. Intell. Rev.*, vol. 58, no. 7, Jul. 2025, doi: 10.1007/s10462-025-11223-9.

[24] G. Taneva-Angelova and D. Granchev, "Deep Learning and Transformer Architectures for Volatility Forecasting: Evidence from U.S. Equity Indices," *Journal of Risk and Financial Management*, vol. 18, no. 12, Dec. 2025, doi: 10.3390/jrfm18120685.

[25] A. Wahyudi, W. Setiawan, and Y. D. P. Negara, "PREDIKSI KURS MATA UANG RIYAL KE RUPIAH MENGGUNAKAN METODE SUPPORT VECTOR REGRESSION (SVR)," *Jurnal METHODIKA*, Sep. 2024.

[26] J. Y. Le Chan *et al.*, "Mitigating the Multicollinearity Problem and Its Machine Learning Approach: A Review," Apr. 01, 2022, *MDPI*. doi: 10.3390/math10081283.

[27] S. F. Akintade, K. Roy, and S. T. Kim, "Hybrid Deep Machine Learning Feature Selection for High-Dimensional Cybersecurity Data," *IEEE Access*, vol. 13, pp. 172136–172156, 2025, doi: 10.1109/ACCESS.2025.3615582.

Matrix: Jurnal Manajemen Teknologi dan Informatika
Volume 16 Issue 2 Year 2026 Pages 101-116
DOI: http://dx.doi.org/10.31940/matrix.v16i2.101-116
URL: http://ojs2.pnb.ac.id/index.php/matrix

Image classification of rerajahan ulap-ulap Bali using MobileNetV2 architecture

Pande Putu Ode Juliantara. KW [1]*, I Gede Aris Gunadi [2], I Made Gede Sunarya [3], Ni Nyoman Emang Smrti [4]

[1,2,3] *Computer Science Study Program, Universitas Pendidikan Ganesha, Indonesia*
[4] *Informatics Engineering Study Program, STMIK Bandung Bali, Indonesia*

*Corresponding Author: *odepande21@gmail.com*

Abstract: *Rerajahan Ulap-Ulap* is a visual expression in Balinese Hindu culture that functions as a sacred element in the *mlaspas* ritual of buildings, containing sacred scripts and specific ornaments. The visual complexity of these motifs presents a challenge in accurately recognizing their types, necessitating a technological approach to support the documentation of this cultural heritage. This study aims to develop an image classification model using the MobileNetV2 architecture with a transfer learning method. The research utilized a primary dataset of 810 original images across nine motif classes, expanded through augmentation to 3,888 images (432 per class) to address data limitations and prevent overfitting. Model performance was evaluated using a 5-Fold Cross Validation method across three optimization scenarios: AdamW, Nadam, and Adagrad. Experimental results demonstrate that Scenario 1 (AdamW) achieved superior performance with an average accuracy of 98.64%, followed by Nadam (98.52%) and Adagrad (71.23%). These findings indicate that the combination of MobileNetV2 and AdamW optimization provides a reliable solution for automated classification of *rerajahan*, serving as a digital instrument to support Balinese cultural preservation efforts.

Keywords: *Rerajahan Ulap-Ulap* Bali, Image Classification, MobileNetV2, Data Augmentation, Cultural Heritage.

History Article: Submitted 22 March 2026 | Revised 5 May 2026 | Accepted 12 May 2026

How to Cite: P. P. O. Juliantara. KW, I. G. A. Gunadi, I. M. G. Sunarya, and N. N. E. Smrti, *"Image classification of rerajahan ulap-ulap Bali using MobileNetV2 architecture," Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 16, no. 2, pp. 101–116, 2026, doi: 10.31940/matrix.v16i2.101-116

Introduction

Rerajahan Ulap-Ulap is a distinctive visual expression within Balinese Hindu culture that functions as a sacred element in the *mlaspas* ritual (the purification of both temples and residential buildings). In general, *ulap-ulap* takes the form of a white cloth measuring approximately 20–30 cm in length and about 15–20 cm in width [1]. On the surface of the cloth, symbols are inscribed in the form of sacred scripts (*aksara nyasa pranawa/modre*) and specific ornaments using black ink, which carry profound religious and philosophical meanings. The creation of *ulap-ulap* is not conducted arbitrarily but is performed by individuals who have undergone a process of self-purification and possess spiritual authority, such as *Pemangku/Pinandita* or *Pandita*, thereby ensuring the preservation of its sacred legitimacy within religious practices. In addition to functioning as a symbol of spiritual protection, *rerajahan ulap-ulap* is also believed to possess metaphysical power, serving as a medium to repel negative energies and to maintain the spiritual balance of a building or sacred space. In Balinese Hindu tradition, *rerajahan* is understood as symbolic imagery or inscriptions that contain religious and magical power, used as a ritual medium and as a means of warding off misfortune in the religious practices of Balinese society [2]. Each symbol inscribed on the *ulap-ulap* cloth carries specific meanings related to Balinese Hindu cosmology, such as protection from *bhuta kala*, spatial purification, and the harmonization of relationships between humans, nature, and God, as conceptualized in the *Tri Hita Karana* philosophy [3].

In the current digital era, advancements in computer vision and deep learning have catalyzed the transition from manual expert interpretation of traditional visual imagery toward automated recognition through computational approaches [4]. However, applying such

technological interventions to *rerajahan ulap-ulap* presents unique challenges, primarily due to its profoundly sacred and esoteric (*kadyatmikan*) nature. This preservation effort is further complicated by deep-seated community beliefs that the indiscriminate study or digitization of these symbols may invoke misfortune or spiritual disturbances. These restrictive conditions have led to a scarcity of experts who possess a comprehensive understanding of the formal rules (*pakem*) and philosophical meanings embedded within these sacred scripts. Consequently, this specialized knowledge is experiencing a gradual decline and faces a significant risk of extinction [5]. In addition, there is a gap in scientific studies connecting the visual cultural heritage of rerajahan with automated image classification methods. In fact, the visual complexity and similarities between patterns often make it difficult for the general public to recognize the types accurately [6]. Developing classification models based on computer vision offers significant potential as an innovative cultural preservation tool. Beyond its ability to identify visual features of profound religious significance, this technology facilitates digital documentation and a more secure, structured dissemination of cultural knowledge for upcoming generations [7].

One of the significant developments in the field of computer vision and deep learning is the emergence of Convolutional Neural Networks (CNN) [8]. CNN is a method within artificial neural networks that utilizes multiple layers to perform the learning process. Its development concept is inspired by the mechanism of neural networks in mammals in processing visual perception. In the context of image recognition, CNN operates by learning patterns present in images through training data, which are then used as a model to produce optimal outputs in the process of image identification or classification [9]. In the classification process, CNN generally requires a relatively large amount of training data in order for the model to learn patterns effectively. However, the application of various data augmentation techniques can serve as a solution to improve model performance, particularly in enhancing its generalization capability toward previously unseen images. Through data augmentation, variations within the dataset can be artificially expanded, enabling the model to become more adaptive and less prone to overfitting [10]. In CNN, various architectures are utilized to perform identification and classification of digital images. One of the widely used architectures that demonstrates high performance is MobileNetV2 [11]. MobileNet is a CNN architecture designed to reduce the high computational resource requirements in deep learning processes, as deep learning methods generally require devices with substantial computational capabilities. This architecture optimizes the use of convolutional layers, making the computation process more efficient compared to conventional CNNs. The improved version of this architecture, MobileNetV2, was introduced in April 2017 by incorporating two main components, namely linear bottleneck and shortcut connections between bottlenecks, to enhance both efficiency and model performance [12].

Previous studies have applied the MobileNetV2 architecture based on transfer learning for the classification of traditional weapons from Central Java. The dataset consisted of five classes of traditional weapons with a total of 785 images. The testing results showed that the model achieved an accuracy of 98.64% with a loss value of 0.16 [13]. Another study focused on classifying Jumputan fabric motifs from Palembang using a dataset of 800 images across four motif classes. The model was trained using several optimizer scenarios, namely AdamW, Adagrad, Nadam, and SGD, with training parameters including 100 epochs, a batch size of 32, and a learning rate of 0.001. Based on the evaluation results, the AdamW optimizer achieved the best performance with a testing accuracy of 97%, followed by Nadam at 96%, Adagrad at 95%, and SGD at 90%. These findings indicate that the use of the MobileNetV2 architecture with a transfer learning approach can produce high classification performance in recognizing traditional fabric motifs [4]. Another study also developed a classification system for Songket Jinengdalem fabric motifs using the MobileNetV2 architecture to support cultural preservation through image recognition technology. The dataset consisted of various Balinese Songket motifs obtained directly from artisans in Jinengdalem Village. The evaluation results showed that the MobileNetV2 model achieved an average training accuracy of 99.98% [7].

Although various studies have demonstrated the successful application of the MobileNetV2 architecture in the classification of cultural image objects such as traditional fabric motifs and other visual artifacts, studies that specifically address the classification of religious symbolic images based on *rerajahan* remain very limited. Most previous research has focused on visual objects that are decorative in nature or have relatively structured patterns, such as batik, songket, and other traditional fabrics, which visually exhibit texture characteristics and repetitive patterns

that are easier for deep learning models to learn. In contrast, *rerajahan ulap-ulap* has distinct visual characteristics, as it consists of a combination of sacred *modre* scripts and complex line forms that do not always exhibit consistent repetitive patterns. In addition, to date, there are still very few studies that utilize computer vision approaches to identify or classify *rerajahan* as an object of visual cultural study. To the best of our knowledge, this is the first study to apply a CNN-based deep learning approach specifically to the automated classification of sacred *Rerajahan Ulap-Ulap* imagery in Bali. This condition identifies a critical research gap, as previous studies have predominantly focused on decorative textiles with repetitive patterns rather than esoteric religious symbols. The selection of the MobileNetV2 architecture in this research is justified by its optimal balance between computational efficiency and high classification accuracy. Unlike more complex and resource-heavy architectures such as EfficientNetV2, ResNet50V2, or Vision Transformers, MobileNetV2 utilizes depthwise separable convolutions that significantly reduce the number of parameters without sacrificing feature extraction capability. This makes it particularly suitable for future implementation on mobile devices or edge computing platforms, which is a key priority for the practical and accessible digitalization of Balinese cultural heritage. Therefore, this study is important to address the existing gap by developing a specialized classification model, thereby contributing not only to the advancement of image processing technology but also to the practical preservation of traditional Balinese cultural knowledge.

Methodology

This study adopts a systematic approach to develop an image classification model for *rerajahan ulap-ulap* using deep learning techniques, specifically leveraging a transfer learning methodology. In the transfer learning process, a large-scale source dataset is utilized to train a model to learn general features such as visual patterns, which are then repurposed to solve new, similar problems. This research utilizes the MobileNetV2 architecture, initialized with pre-trained weights from the ImageNet-1k dataset, to significantly reduce the requirement for vast amounts of labeled data while accelerating the learning process. To adapt the model to the specific visual domain of *rerajahan ulap-ulap*, the entire base model (154 layers) was frozen, and a custom classification head was integrated, consisting of a Global Average Pooling layer, a Dropout layer (0.5) to mitigate overfitting, and a Dense layer with 512 neurons.

To identify the most effective optimization strategy, the training process was conducted through three distinct optimizer scenarios: AdamW, Nadam, and Adagrad. All computational processes and experiments were performed using the Google Colab platform. This study utilizes primary data obtained directly to ensure the relevance and authenticity of the visual assets used. The selection of hyperparameters, including the 0.001 learning rate, was established based on configurations from previous studies, with specific modifications to better suit the unique visual complexities of the *rerajahan ulap-ulap* dataset [4], [7], [14]. These adjustments were further validated through empirical manual tuning to ensure stable convergence across all scenarios. By combining transfer learning with robust data augmentation techniques, the model achieves sufficient feature extraction capability despite the specialized and relatively compact nature of the dataset. To ensure statistical validity and overcome potential biases from limited sample sizes, this study implemented a 5-Fold Cross Validation procedure for each optimizer scenario to produce a more generalized, robust, and stable performance estimation. The methodology consists of several integrated stages, including data acquisition, dataset splitting, pre-processing,

and augmentation, all of which are designed to ensure rigorous data preparation and effective model training. The complete framework of this research methodology is illustrated in Figure 1.

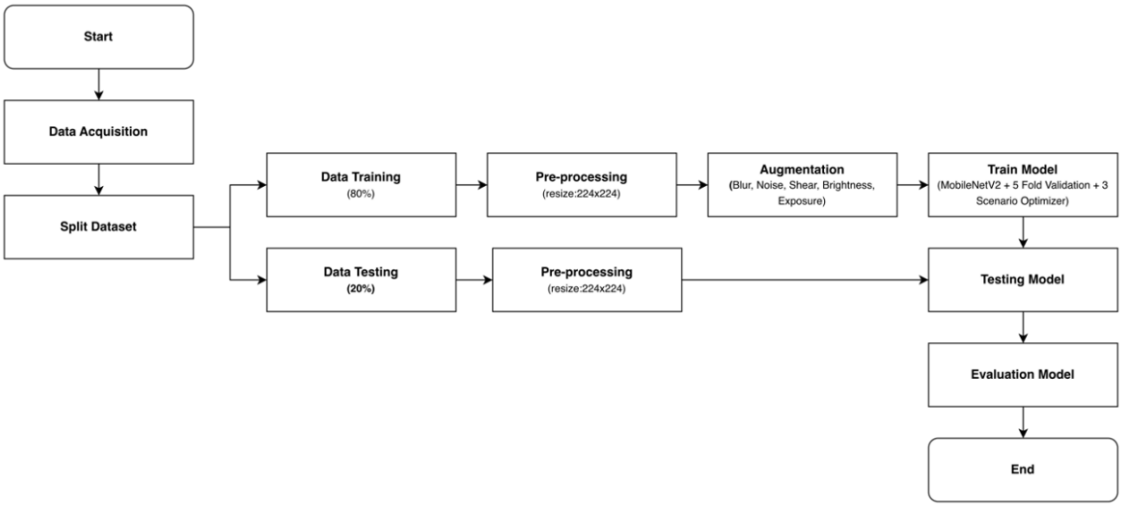


Figure 1. Research methodology

Data Acquisition

The dataset was constructed through manual drawing by five different individuals to introduce variations in form and drawing style. Each participant created *rerajahan* on a white cloth measuring 20 × 30 cm using a permanent marker to maintain consistent line clarity while preserving the authentic texture of the original medium. All drawings referred to *Makarya Ulap-Ulap Palinggih lan Wewangunan* [1]. The dataset includes nine commonly used types of *rerajahan* in traditional Balinese architecture. An example of the dataset distribution is presented in Table 1. Image acquisition was conducted using a Canon 700D DSLR camera (18 MP, 50 mm lens, f/5) at a distance of 100–150 cm under natural lighting conditions between 09:00 and 14:00 WITA to ensure clear and undistorted image quality. To enhance data diversity and model robustness, four acquisition scenarios were applied, including flat cloth, wrinkled cloth, tilted angles (15°–30°), and folded edges, simulating real-world conditions.

Figure 2. The process of drawing *Rerajahan Ulap-Ulap*

Data Augmentation

Data augmentation was applied exclusively to the training dataset to increase data variability by modifying the original images without altering their underlying labels or class identities [15]. The applied techniques include brightness and exposure adjustments to simulate lighting variations, the addition of random noise to represent environmental disturbances, blur to simulate out-of-focus conditions, and shear transformation to introduce slight geometric variations. This process effectively enhances the model's performance by expanding the diversity of the training data without the need for manual data collection.

Training Model

The model training stage is a crucial phase in which the MobileNetV2 architecture learns the visual features of *rerajahan ulap-ulap* images hierarchically through a transfer learning strategy by freezing 154 base layers and optimizing custom layers consisting of Global Average

Pooling, a 256-neuron Dense layer, and 0.6 Dropout to prevent overfitting. This experiment was conducted through three comparative optimization scenarios using the AdamW, Nadam, and Adagrad algorithms with a learning rate of 0.0001, integrated with a 5-Fold Cross Validation method to objectively evaluate the model's performance while ensuring performance stability and minimizing bias in the data samples [16]. Through this technique, a total of 3,888 image data (432 per class) were systematically divided into five balanced folds to allow each subset of data to be used as test data in rotation. In each iteration, 4 folds (3,110 data) were used as training data and 1 fold (778 data) as validation data to maintain class distribution proportions consistent with the original dataset. This process, which took place over five iterations as illustrated in Figure 3, was regulated by an Early Stopping mechanism (patience 10) to produce a final performance estimate that is robust, stable, and capable of generalizing well to unseen data.

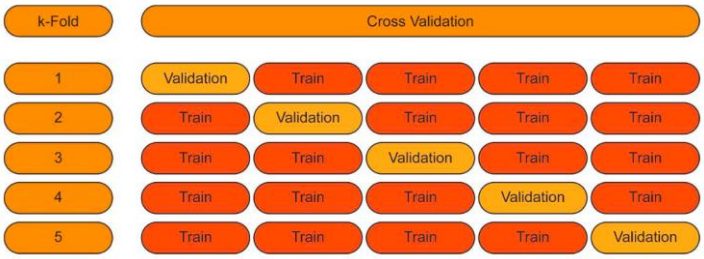


Figure 3. Illustration of 5 fold cross validation process

Further details regarding the parameter settings used during the experimental process and model training can be found in the hyperparameter configuration Table 1.

Table 1. Model training configuration

Parameter	Scenario 1	Scenario 2	Scenario 3
Pre-trained Weights	ImageNet	ImageNet	ImageNet
Input Size	224 x 224 x 3	224 x 224 x 3	224 x 224 x 3
Base Layer Status	Frozen (154 layers)	Frozen (154 layers)	Frozen (154 layers)
Custom Top Layers	GAP2D, Dense 256, Dropout 0.6, Softmax 9	GAP2D, Dense 256, Dropout 0.6, Softmax 9	GAP2D, Dense 256, Dropout 0.6, Softmax 9
Optimizer Type	AdamW	Nadam	Adagrad
Learning Rate	0.0001	0.0001	0.0001
Batch Size	32	32	32
Max Epochs	100	100	100
Early Stopping	Patience = 10	Patience = 10	Patience = 10
Validation Method	5-Fold Validation	5-Fold Validation	5-Fold Validation

Evaluation Model

Model evaluation is a systematic stage aimed at assessing the performance, quality, or effectiveness of a developed and trained algorithm or model [17]. This process is conducted after the training phase is complete, where the model is tested using a test dataset to determine the program's proficiency in performing classification or prediction on new, unseen data [18]. The model evaluation was conducted using a confusion matrix to analyze classification performance in detail. Based on this matrix, several evaluation metrics were calculated, including accuracy, precision, recall, and F1-score. Accuracy is defined as the ratio between the number of correctly predicted labels and the total number of labels, including both predicted and actual labels. The overall accuracy is obtained by averaging the accuracy across all evaluated instance [19]. The accuracy can be calculated using the formula [9]:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

Precision represents the ratio between correctly predicted positive labels and the total predicted positive labels, which is then averaged across all instances [19]. The precision formula is [9]:

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall is defined as the ratio between correctly predicted positive labels and the total actual positive labels, indicating the model's ability to correctly identify all relevant instances within a specific class [20]. The recall formula is expressed as [9]:

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

F1-score is a metric that combines precision and recall to provide a balanced evaluation of both. It is particularly important when there is an imbalance between precision and recall, as it uses the harmonic mean to reflect the overall classification performance of the model [20] The F1-score can be calculated using the formula [9]:

$$F1\ Score = 2\ x\ \frac{Recall\ x\ Precision}{Recall +\ Precision} \tag{4}$$

Results and Discussions

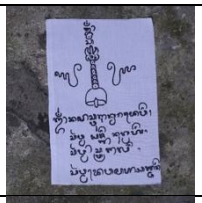
Results

Dataset

In this study, the results of model training and evaluation are presented based on predefined experimental schemes. The dataset consists of 90 image samples for each class, divided into training and testing datasets using an 80:20 ratio, resulting in 72 images per class for training and 18 images per class for testing. Data distribution was balanced across all classes to ensure fair representation and prevent bias during evaluation. All images underwent a pre-processing stage, including resizing to 224 × 224 pixels to match the input requirements of the MobileNetV2 architecture and normalization to standardize pixel values and improve training stability. Data augmentation was then applied to the training dataset, increasing the total number of training images to 432 images per class. This process enhances dataset diversity and improves the model's ability to generalize across various visual conditions. Details regarding the data distribution after the acquisition and augmentation process are presented in Table 2.

Table 2. Dataset distribution per class

Class	Image Sample	Data Train (80%)	Data Test (20%)	Total Data Train After Augmentation
angkul_angkul		72	18	432
bale_daje		72	18	432

bale_dangin		72	18	432
bale_dauh		72	18	432
bale_delod		72	18	432
kemulan		72	18	432
padmasana		72	18	432
paon		72	18	432
tugu_karang		72	18	432
	Total		**162**	**3.888**

Training

The training and validation results for this optimization scenario are comprehensively detailed in Table 3. The data reveals a high level of stability across the 5 fold cross validation process, characterized by a narrow margin of variance in performance metrics. The model achieved a robust average Test Accuracy of 97.11% and a remarkably low average Test Loss of 0.0811, while maintaining a high average Training Accuracy of 99.63%. A deeper analysis of the individual iterations shows that Fold 3 yielded the most optimal results, achieving a peak Test Accuracy of 97.86% and the minimum Test Loss of 0.0656. Furthermore, the consistency between the Training Accuracy (99.63%) and Test Accuracy (97.11%) suggests that the model possesses strong generalization capabilities. The marginal discrepancy of approximately 2.52% between the two metrics confirms that the integration of a 0.6 Dropout ratio and the Early Stopping mechanism effectively regularized the network.

Table 3. Recapitulation of loss value and accuracy of each fold in scenario 1

Fold	Train Loss	Train Acc	Test Lost	Test Acc
1	0.0217	0.9971	0.0948	0.9653
2	0.0256	0.9964	0.0909	0.9697
3	0.0247	0.9958	0.0656	0.9786
4	0.0236	0.9960	0.0772	0.9697
5	0.0256	0.9960	0.0771	0.9724
AVG	**0.02424**	**0.99626**	**0.08112**	**0.97114**

The visualization of the training progress for the Scenario 1 is illustrated in Figure 4, which displays the average accuracy and loss curves across 5 fold cross validation. The accuracy plot (left) shows a steady upward trend for both training and validation sets, reaching a state of convergence beyond the 60 epoch. Notably, the validation accuracy remains closely aligned with the training accuracy, exhibiting minimal fluctuations, which indicates stable learning behavior. Complementing the accuracy trend, the loss plot (right) demonstrates a consistent decay in both average training and validation loss. The curves maintain a parallel trajectory without significant divergence, further substantiating the effectiveness of the 0.6 Dropout ratio in preventing overfitting. As shown in the graph, the training process concluded before reaching the maximum 100 epochs due to the Early Stopping mechanism, which triggered once the validation loss ceased to improve for 10 consecutive epochs. This ensures that the model weights used in Table 3 represent the most optimal state of convergence, balancing high recognition accuracy with robust generalization.

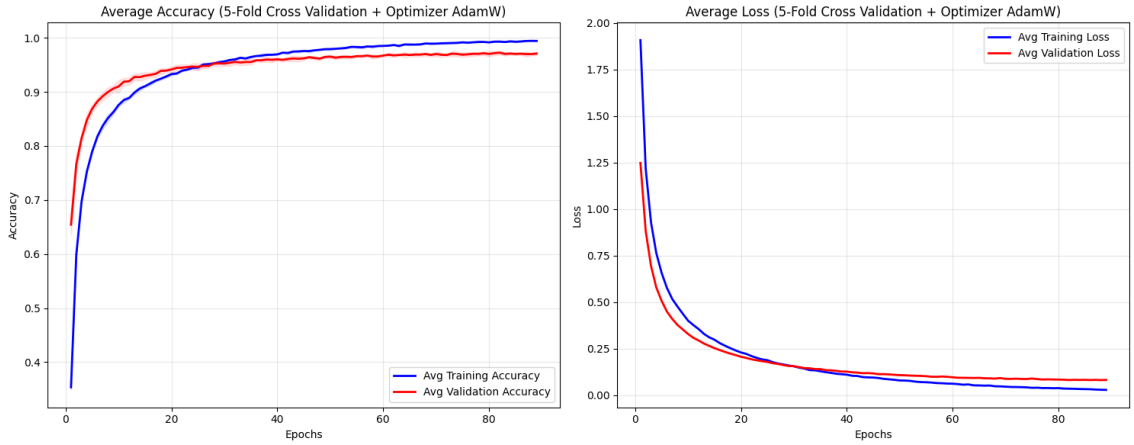


Figure 4. Average accuracy and loss curves scenario 1

The training and validation results for the Scenario 1 are comprehensively detailed in Table 4. The data reveals a high level of stability across the 5 fold cross validation process, characterized by a narrow margin of variance in performance metrics. The model achieved a robust average Test Accuracy of 97.10% and a remarkably low average Test Loss of 0.0802, while maintaining a high average Training Accuracy of 99.53%. A deeper analysis of the individual iterations shows that Fold 3 yielded the most optimal results, achieving a peak Test Accuracy of 97.68% and the minimum Test Loss of 0.0629. Furthermore, the consistency between the Training Accuracy (99.53%) and Test Accuracy (97.10%) suggests that the model possesses strong generalization capabilities. The marginal discrepancy of approximately 2.43% between the two metrics confirms that the integration of a 0.6 Dropout ratio and the Early Stopping mechanism effectively regularized the network. This balance prevented the model from memorizing the training noise, ensuring instead that it learned the fundamental structural patterns of the symbols.

Table 4. Recapitulation of loss value and accuracy of each fold in scenario 2

Fold	Train Loss	Train Acc	Test Lost	Test Acc
1	0.0321	0.9942	0.0888	0.9662
2	0.0241	0.9951	0.0937	0.9679
3	0.0234	0.9951	0.0629	0.9768
4	0.0221	0.9962	0.0742	0.9724
5	0.0257	0.9960	0.0813	0.9715
AVG	**0.02548**	**0.99532**	**0.08018**	**0.97096**

Figure 5 illustrates the training and validation progress for the Scenario 2 through average accuracy and loss plots. The accuracy curve (left) demonstrates a rapid learning phase in the early epochs, followed by a steady convergence that stabilizes after approximately the 50 epoch. The validation accuracy closely tracks the training accuracy, maintaining a consistent trend with minimal variance, which signifies the model's high stability during the cross validation process. Simultaneously, the loss curve (right) shows a significant and smooth reduction in both average training and validation loss. The absence of a widening gap between these two curves indicates that the model generalizes effectively to the validation sets. Similar to the previous scenario, the training process was automatically terminated at approximately epoch 85 by the Early Stopping mechanism, preventing unnecessary iterations once the validation loss reached its optimum.

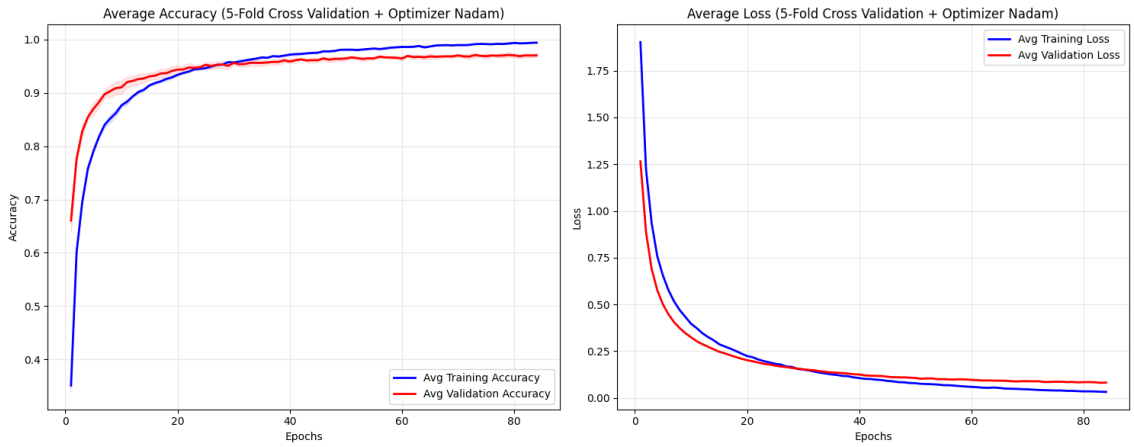


Figure 5. Average accuracy and loss curves scenario 2

The performance metrics for Scenario 3 are presented in Table 5. The data indicates a distinct learning behavior compared to previous scenarios, characterized by higher loss values and lower accuracy across all folds. The model achieved an average Training Accuracy of 59.81% and a Test Accuracy of 71.40%, with an average Test Loss of 1.09198. Despite the lower overall performance, the results remain consistent across the 5 fold cross validation process, showing a stable but slower convergence rate. Interestingly, the Test Accuracy (71.40%) consistently outperformed the Training Accuracy (59.81%) in this scenario. This phenomenon suggests that while the implementation of Scenario 3 struggled to reach a deep minimum during the training phase within the allotted epochs, the model maintained a fair level of generalization on the validation subsets. Among the five iterations, Fold 3 again recorded the highest Test Accuracy at 72.75%, while Fold 5 achieved the lowest Training Loss of 1.2217.

Table 5. Recapitulation of loss value and accuracy of each fold in scenario 3

Fold	Train Loss	Train Acc	Test Lost	Test Acc
1	1.2380	0.5917	1.0731	0.7251
2	1.2516	0.5996	1.1257	0.6963
3	1.2397	0.5998	1.0962	0.7275
4	1.2407	0.5958	1.0845	0.6990
5	1.2217	0.6034	1.0804	0.7222
AVG	**1.23834**	**0.59806**	**1.09198**	**0.71402**

Figure 6 illustrates the training and validation progress for Scenario 3 through average accuracy and loss curves. Unlike the previous scenarios, the accuracy plot (left) shows a significantly slower convergence rate, with both curves continuing to rise steadily even as they approach the 100 epoch. The loss plot (right) complements this observation by showing a gradual and continuous decline in both average training and validation loss without reaching a definitive plateau. The absence of a triggered Early Stopping mechanism as the training proceeded for the full 100 epochs indicates that Scenario 3 required more iterations to reach an optimal state compared to Scenario 1 and Scenario 2.

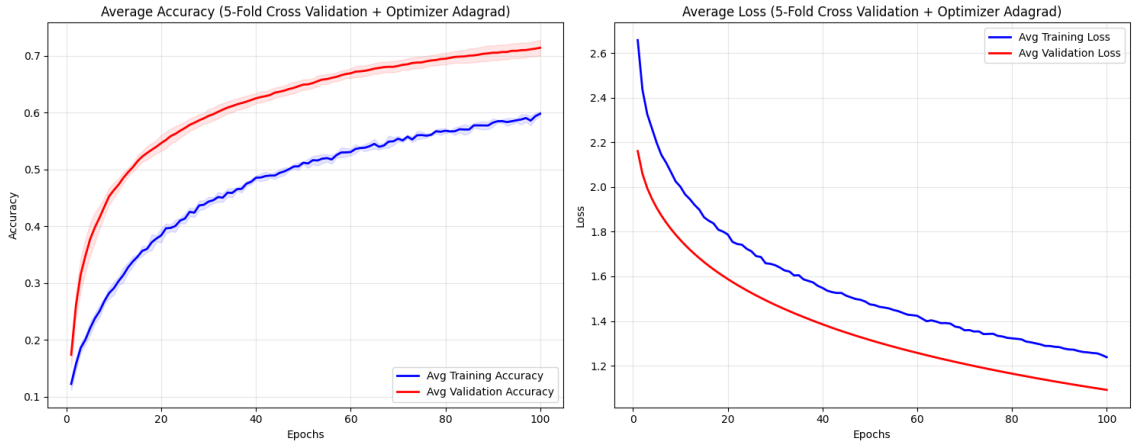


Figure 6. Average accuracy and loss curves scenario 3

Figure 7 presents a comparative analysis of the average validation loss and validation accuracy across the three optimization scenarios: Scenario 1 (AdamW), Scenario 2 (Nadam), and Scenario 3 (Adagrad). The left plot clearly demonstrates that both AdamW (blue) and Nadam (green) achieved a significantly faster and deeper convergence, with loss values dropping sharply below 0.2 within the first 40 epochs. In contrast, the Adagrad optimizer (orange) exhibited a much slower decay, maintaining a substantially higher loss throughout the 100 epoch duration, which suggests a less efficient exploration of the loss landscape for this specific dataset. The accuracy comparison in the right plot further reinforces these findings. AdamW and Nadam show nearly identical performance trajectories, both stabilizing at high accuracy levels above 97%. This indicates that both optimizers are highly effective in refining the MobileNetV2 weights for *Rerajahan Ulap-Ulap* imagery. Conversely, Adagrad reached a peak validation accuracy of only 71.40%, significantly underperforming compared to the other two candidates.

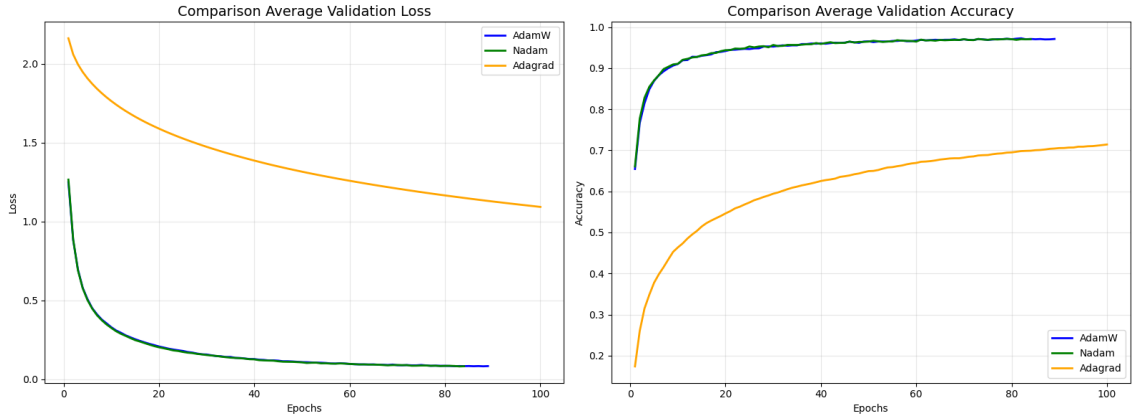


Figure 7. Comparison average accuracy and loss curves

Evaluation

The evaluation phase was conducted to measure the extent to which the trained model could recognize new, unseen data using an independent test dataset. This testing utilizes standard classification metrics to provide a comprehensive overview of the model architecture's accuracy and stability.

Based on comprehensive testing on that test dataset, the MobileNetV2 model using the Scenario 1 demonstrates impressive levels of accuracy and stability. The achieved average accuracy of 98.64% proves that this architecture is capable of recognizing the visual features of *Rerajahan Ulap-Ulap* motifs with a high degree of precision. The closely aligned values for Precision (98.75%), Recall (98.64%), and F1-Score (98.64%) indicate that the model possesses a well-balanced capability in minimizing both false positives and false negatives across all class categories.

A deeper analysis of each testing iteration reveals that Fold 2 successfully achieved the best performance compared to other folds, with an accuracy of 98.77% and the highest Precision value of 98.89%. The robustness of the model in this scenario is further strengthened by the very low Standard Deviation (STD DEV) value of 0.0025, providing statistical validation that the model is highly robust and insensitive to differences in data partitioning during the cross validation process.

Table 6. Recapitulation of evaluation model in scenario 1

Fold	Accuracy	Precision	Recall	F1-Score
1	0.9815	0.9818	0.9815	0.9815
2	0.9877	0.9889	0.9877	0.9876
3	0.9877	0.9889	0.9877	0.9876
4	0.9877	0.9889	0.9877	0.9876
5	0.9877	0.9889	0.9877	0.9876
AVG	**0.9864**	**0.9875**	**0.9864**	**0.9864**
STD DEV	0.0025	0.0028	0.0025	0.0025

A more detailed analysis of the model's prediction distribution can be seen in Figure 8, which presents the Normalized Confusion Matrix for the Scenario 1. This matrix reflects the accumulated results of the 5 fold cross validation, where the values on the main diagonal represent the percentage of successful predictions for each class relative to its actual label. Based on the data in Figure 8, the model demonstrates perfect recognition capability (100.00%) across most classes, including: *angkul-angkul*, *bale_daje*, *bale_dauh*, *kemulan*, *padmasana*, *paon*, and *tugu_karang*. This indicates that the visual features of these categories are highly distinctive, allowing them to be distinguished with exceptional accuracy by the MobileNetV2 architecture.

Nevertheless, as shown in Figure 8, a slight classification ambiguity is observed within the *bale_dangin* and *bale_delod* categories. The *bale_dangin* class achieved an accuracy rate of

88.89%, with 11.11% of the data being misclassified as *bale_delod*. Conversely, the *bale_delod* class reached 98.89% accuracy, with 1.11% of the data predicted as *bale_dangin*. This reciprocal misclassification pattern suggests a high degree of similarity in visual features or building morphology between the *bale_dangin* and *bale_delod* motifs.

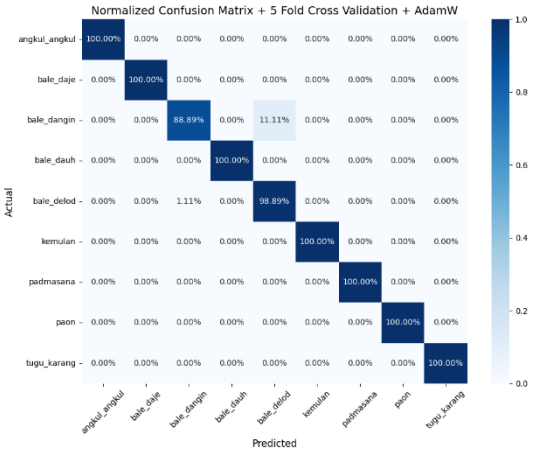


Figure 8. Confusion matrix scenario 1

Based on the comprehensive testing on the test dataset presented in Table 6, Scenario 2 demonstrates a high level of accuracy and stability. The achieved average accuracy of 98.52% proves that this architecture is highly capable of recognizing the visual features of *Rerajahan Ulap-Ulap* motifs. The highly consistent values for Precision (98.58%), Recall (98.52%), and F1-Score (98.52%) indicate that the model possesses a balanced capability in minimizing classification errors across all class categories. A deeper analysis of each testing iteration in Table 6 reveals that Fold 1 successfully achieved the best performance compared to other folds, with an accuracy of 98.77% and a Precision value of 98.89%. The robustness of the model in this scenario is further strengthened by the low Standard Deviation (STD DEV) value of 0.0030, providing statistical validation that the model is robust and insensitive to differences in data partitioning during the cross validation process.

Table 7. Recapitulation of evaluation model in scenario 2

Fold	Accuracy	Precision	Recall	F1-Score
1	0.9877	0.9889	0.9877	0.9876
2	0.9815	0.9818	0.9815	0.9815
3	0.9815	0.9818	0.9815	0.9815
4	0.9877	0.9877	0.9877	0.9877
5	0.9877	0.9889	0.9877	0.9876
AVG	**0.9852**	**0.9858**	**0.9852**	**0.9852**
STD DEV	0.0030	0.0033	0.0030	0.0030

A more detailed analysis of the model's prediction distribution in Scenario 2 can be seen in Figure 9. Based on the data, the model achieved a perfect recognition rate (100.00%) for the majority of classes, including: *angkul-angkul*, *bale_daje*, *bale_dauh*, *kemulan*, *padmasana*, *paon*, and *tugu_karang*. This demonstrates that the optimizer in Scenario 2 is highly effective in helping the MobileNetV2 architecture extract distinctive visual features for these categories.

However, as shown in Figure 9, slight ambiguities remain in the classification of *bale_dangin* and *bale_delod* motifs. The *bale_dangin* class achieved an accuracy of 90.00%, where 10.00% of the samples were incorrectly predicted as *bale_delod*. Conversely, the *bale_delod* class recorded an accuracy of 96.67%, with 3.33% of the data misclassified as *bale_dangin*. This pattern shows a shift in misclassification rates where *bale_dangin* accuracy slightly improved, while *bale_delod* accuracy experienced a slight decrease compared to the

previous scenario. This indicates that Scenario 2 optimization contributes differently to distinguishing building morphological features with high visual similarity.

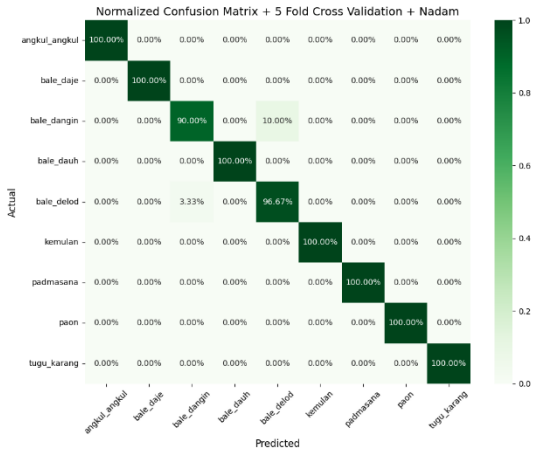


Figure 9. Confusion matrix scenario 2

Based on the comprehensive testing on the test dataset presented in Table 8, the MobileNetV2 model in Scenario 3 shows a significant decrease in performance compared to the previous scenarios. The achieved average accuracy of 71.23% indicates challenges in optimally extracting the visual features of *Rerajahan Ulap-Ulap* motifs within this scenario. The average values for Precision (75.09%), Recall (71.23%), and F1-Score (67.83%) suggest a higher misclassification rate and an imbalance in recognizing the various class categories tested. A deeper analysis of each testing iteration in Table 8 reveals that Fold 2 recorded the best performance with an accuracy of 75.93%. Unlike the previous scenarios which maintained high stability, Scenario 3 exhibits a much larger Standard Deviation (STD DEV), specifically 0.0323 for accuracy and 0.0587 for Precision. This high variance provides statistical validation that the model in this scenario is less robust, with its performance being significantly affected by differences in data partitioning during the cross validation process.

Table 8. Recapitulation of evaluation model in scenario 3

Fold	Accuracy	Precision	Recall	F1-Score
1	0.6728	0.7358	0.6728	0.6363
2	0.7593	0.7974	0.7593	0.7436
3	0.6914	0.6458	0.6914	0.6382
4	0.6975	0.7647	0.6975	0.6667
5	0.7407	0.8110	0.7407	0.7069
AVG	**0.7123**	**0.7509**	**0.7123**	**0.6783**
STD DEV	0.0323	0.0587	0.0323	0.0414

A detailed analysis of the model's prediction distribution in Scenario 3 can be seen in Figure 10, illustrating a significant performance degradation and widespread misclassification across nearly all categories of Balinese *Rerajahan* and *Ulap-Ulap* motifs. Based on the data, the model only maintained perfect performance (100.00%) for two categories, specifically the kemulan and padmasana motifs. Conversely, other categories suffered from massive predictive overlaps, with the *angkul_angkul* motif emerging as the lowest performing category only 21.11% of the data was correctly predicted, while the remaining 64.44% was incorrectly classified as the *padmasana* motif.

Extremely high ambiguity is also clearly observed in the *bale_dangin* motif, which only reached an accuracy of 34.44%, with significant misclassification spreads toward the *bale_delod* (27.78%) and *tugu_karang* (22.22%) motifs. The visualization in Figure 10 confirms that the implementation of Scenario 3 failed to help the model extract distinctive enough visual features

to differentiate the unique characteristics between motifs. The high number of values scattered outside the main diagonal indicates that the model failed to achieve optimal convergence, resulting in an average accuracy that is considerably lower and less stable than that of the previous testing scenarios.

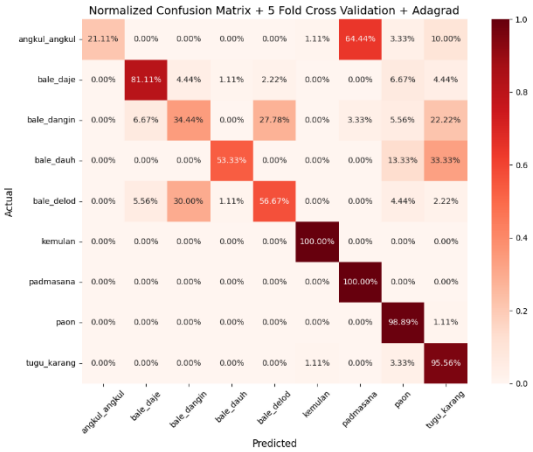


Figure 10. Confusion matrix scenario 3

Based on the comprehensive testing results presented in Table 8, it can be concluded that Scenario 1 delivers the best performance in classifying Balinese *Rerajahan Ulap-Ulap* motifs. This scenario achieved the highest average accuracy of 98.64%, followed by Precision (98.75%), Recall (98.64%), and F1-Score (98.64%). These results demonstrate that the configuration in Scenario 1 possesses the most superior and stable ability to recognize the unique visual characteristics of the dataset. Scenario 2 follows with nearly identical performance, recording an average accuracy of 98.52%. Although there is a slight decrease compared to the first scenario, the metric values remain at a very high level, proving the model's robustness in classification. On the other hand, Scenario 3 shows a significant performance decline, with an average accuracy of only 71.23% and an F1-Score of 67.83%. This indicates that the approach in Scenario 3 is less effective in capturing the complex features of *Rerajahan Ulap-Ulap* motifs compared to the other two scenarios.

Table 9. Final result

Scenario	Accuracy (avg)	Precision (avg)	Recall (avg)	F1-Score (avg)
1	98.64%	98.75%	98.64%	98.64%
2	98.52%	98.58%	98.52%	98.52%
3	71.23%	75.09%	71.23%	67.83%

Discussions

The experimental results demonstrate that the MobileNetV2 architecture, combined with a transfer learning strategy, is highly effective in recognizing the unique visual characteristics of Balinese *Rerajahan Ulap-Ulap* motifs. Based on the comparison of the three tested schemes, Scenario 1 (AdamW) delivered the most superior performance, achieving an average accuracy of 98.64%, followed closely by Scenario 2 (Nadam) with an accuracy of 98.52%. The superiority of AdamW in this study aligns with findings from previous traditional fabric motif classification studies, where the algorithm proved more adaptive in handling complex visual features. The performance stability in Scenario 1 was further reinforced by a very low standard deviation (0.0025), providing statistical validation that the model is robust and consistent despite being tested with different data partitions through the 5 fold cross validation method.

On the other hand, Scenario 3 (Adagrad) showed a significant performance decline, with an average accuracy of only 71.23%. Analysis of the training curves indicates that the approach in this scenario experienced a much slower convergence rate, where the loss values remained

high and failed to reach a definitive plateau even by the 100 epoch. This suggests that the visual features in the primary *Rerajahan Ulap-Ulap* dataset require more dynamic weight adjustments during the training process an aspect that was handled less optimally by Adagrad compared to AdamW or Nadam. The lower performance in Scenario 3 was also accompanied by a higher variance (standard deviation of 0.0323), indicating that the model was less stable in generalizing motif patterns across various data subsets.

Although the models in Scenarios 1 and 2 achieved very high accuracy, analysis through the normalized confusion matrix revealed specific challenges regarding the *bale_dangin* and *bale_delod* motifs. Reciprocal misclassification occurred, where in Scenario 1, 11.11% of *bale_dangin* data was predicted as *bale_delod*, reflecting a very strong visual morphological similarity between the two categories. The characteristics of *rerajahan ulap-ulap*, which involve complex lines and sacred *modre* scripts, indeed present a higher level of feature extraction difficulty compared to decorative objects with repetitive texture patterns. However, the integration of data augmentation techniques, the use of a 0.6 Dropout ratio, and the Early Stopping mechanism proved crucial in preventing overfitting and ensuring the model remained capable of generalizing well on the primary data used. The success in developing this model provides an innovative solution for the digitalization and preservation of Balinese cultural heritage through a computer vision technology approach.

Conclusion

Based on the experiments and analysis conducted, it can be concluded that the MobileNetV2 architecture is highly effective for classifying Balinese *Rerajahan Ulap-Ulap* motifs with a very high degree of accuracy. The test results demonstrate that the choice of optimization scenario is crucial in determining model performance, where Scenario 1 (AdamW) emerged as the best model with an average accuracy of 98.64%, Precision of 98.75%, Recall of 98.64%, and an F1-Score of 98.64%. This performance outperforms Scenario 2 (Nadam), which achieved 98.52% accuracy, and Scenario 3 (Adagrad), which only reached 71.23%. The application of data augmentation techniques and the Early Stopping mechanism successfully addressed the limitations of the primary dataset and prevented overfitting, resulting in a model with strong generalization capabilities.

The successful development of this model provides a significant contribution to the efforts of digitalizing cultural heritage and preserving traditional Balinese knowledge through a computer vision technology approach. However, given that the primary dataset used in this study is limited in size, it is highly recommended that further testing be conducted using external datasets outside the currently available set to ensure the model's reliability and robustness in real-world scenarios. Testing with independent data that includes a wider range of environmental conditions, lighting variations, and drawing styles will validate whether the model can maintain high accuracy in classifying objects more universally and reliably.

References

[1] I. W. Dharmawan and I. P. A. Suryawan, *Makarya Ulap-Ulap Palinggih lan Wewangunan* . Yogyakarta: Pustaka Pranala, 2020.

[2] I. K. D. Pendit, "Studi Pembelajaran Religi Hindu Dalam Gambar Rerajahan Ulap-Ulap," *Social Studies*, vol. 10, no. 2, pp. 1–15, Sep. 2023.

[3] I. G. W. Simrana, I. W. Mudra, I. G. Yudarta, and N. W. Ardini, "Makna Bentuk dan Aksara Rerajahan," *Jurnal Bali Membangun Bali*, vol. 4, no. 2, pp. 101–113, Aug. 2023, doi: 10.51172/jbmb.v4i2.273.

[4] M. Sahpira and Yohannes, "Transfer Learning dengan MobileNetV2 untuk Klasifikasi Motif Jumputan Palembang," *Jurnal Algoritme*, vol. 6, no. 1, pp. 1–10, Oct. 2025.

[5] I. W. Sudana, "EKSISTENSI RERAJAHAN SEBAGAI MANIFESTASI MANUNGGALNYA SENI DENGAN RELIGI," *Imaji*, vol. 7, no. 2, pp. 141–158, Nov. 2015, doi: 10.21831/imaji.v7i2.6631.

[6] N. M. R. A. Permatasari, M. W. A. Kesiman, and I. M. G. Sunarya, "Klasifikasi Jenis Samphyan Banten Upakara Adat Bali Dengan Arsitektur VGG-16 Dan InceptionResnet-V2," *SINTECH (Science and Information Technology) Journal*, vol. 8, no. 3, pp. 220–230, Dec. 2025, doi: 10.31598/sintechjournal.v8i3.2009.

[7] K. Y. Mahendra, L. J. E. Dewi, I. K. Purnamawan, and F. B. Pasaribu, "Songket Motif Classification using MobileNet Architecture for Cultural Heritage Preservation," *International Journal of Informatics and Computation*, vol. 7, no. 2, Dec. 2025.

[8] R. M. M. Komang, J. E. D. Luh, Y. E. A. Kadek, A. S. Ketut, and V. Pariwate, "ANALISIS HYPERPARAMETER PADA KLASIFIKASI JENIS TANAMAN MENGGUNAKAN ALGORITMA RESNET50 DAN MOBILENETV2," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 6, pp. 9921–9928, Nov. 2025, doi: 10.36040/jati.v9i6.15832.

[9] M. W. A. Kesiman, I. M. G. Sunarya, and I. G. L. T. Sumantara, "Comparative Analysis of CNN Methods for Periapical Radiograph Classification," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 13, no. 2, pp. 204–214, Jul. 2024, doi: 10.23887/janapati.v13i2.71664.

[10] N. P. D. A. S. Dewi, M. W. A. Kesiman, I. M. G. Sunarya, G. A. A. D. Indradewi, and I. G. Andika, "Klasifikasi Jenis Daun Tumbuhan Herbal Berdasarkan Lontar Usada Taru Pramana Menggunakan CNN," *Techno.Com*, vol. 23, no. 1, pp. 271–283, Feb. 2024, doi: 10.62411/tc.v23i1.9510.

[11] I. K. P. G. Hartawan and I. M. G. Sunarya, "Tinjauan Sistematis Literatur Tentang Sistem Deteksi Penyakit Berbasis Deep Learning," *Digital Transformation Technology (Digitech)*, vol. 6, no. 1, pp. 112–123, Mar. 2026.

[12] I. Mudzakir and T. Arifin, "Klasifikasi Penggunaan Masker dengan Convolutional Neural Network Menggunakan Arsitektur MobileNetv2," *EXPERT: Jurnal Manajemen Sistem Informasi dan Teknologi*, vol. 12, no. 1, p. 76, Jun. 2022, doi: 10.36448/expert.v12i1.2466.

[13] O. Saputra, D. I. Mulyana, and M. B. Yel, "Implementasi Algoritma Convolutional Neural Network (CNN) Untuk Klasifikasi Senjata Tradisional Di Jawa Tengah Dengan Metode Transfer Learning," *Jurnal SISKOM-KB (Sistem Komputer dan Kecerdasan Buatan)*, vol. 5, no. 2, pp. 45–52, Mar. 2022, doi: 10.47970/siskom-kb.v5i2.282.

[14] Y. N. FUADAH, I. D. UBAIDULLAH, N. IBRAHIM, F. F. TALININGSING, N. K. SY, and M. A. PRAMUDITHO, "Optimasi Convolutional Neural Network dan K-Fold Cross Validation pada Sistem Klasifikasi Glaukoma," *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 10, no. 3, p. 728, Jul. 2022, doi: 10.26760/elkomika.v10i3.728.

[15] M. F. A. Maulana, N. M. Anggadimas, and D. A. Sani, "Klasifikasi Citra Penyakit Daun Padi Dengan Metode CNN Menggunakan Arsitektur ResNet50V2," *CESS (Journal of Computer Engineering, System and Science)*, vol. 10, no. 2, pp. 517–529, Jul. 2025, doi: 10.24114/cess.v10i2.66960.

[16] G. W. Purnama, A. A. J. Permana, and N. K. Kertiasih, "KLASIFIKASI CITRA MEDIS MAMOGRAFI BERBASIS MULTI-VIEW CONVOLUTIONAL NEURAL NETWORKS," *Jurnal Pendidikan Teknologi dan Kejuruan*, vol. 22, no. 2, pp. 162–171, 2025.

[17] I. K. Y. Prayoga, I. M. G. Sunarya, and P. H. Suputra, "Klasifikasi Penyakit Demam Berdarah Menggunakan Algoritma Stacking Ensemble Learning," *SINTECH Journal*, vol. 8, no. 2, pp. 104–116, Aug. 2025.

[18] I. P. W. Prasetia and I Made Gede Sunarya, "Image Classification of Balinese Seasoning Base Genep Based on Deep Learning," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 13, no. 1, Mar. 2024, doi: 10.23887/janapati.v13i1.67967.

[19] T. Bariyah, M. Arif Rasyidi, and Ngatini, "Convolutional Neural Network Untuk Metode Klasifikasi Multi-Label Pada Motif Batik Convolutional Neural Network for Multi-Label Batik Pattern Classification Method," 2021.

[20] P. P. Vedanty, M. W. A. Kesiman, and I. M. G. Sunarya, "PENGARUH DATA AUGMENTASI PADA IDENTIFIKASI PENYAKIT DAUN TANAMAN OBAT MENGGUNAKAN K-NEAREST NEIGHBOR," *JATI (Jurnal Mahasiswa Teknik Informatika)*, vol. 9, no. 2, pp. 2094–2100, Mar. 2025, doi: 10.36040/jati.v9i2.12961.

Matrix: Jurnal Manajemen Teknologi dan Informatika
Volume 16 Issue 2 Year 2026 Pages 117-128
DOI: http://dx.doi.org/10.31940/matrix.v16i2.117-128
URL: http://ojs2.pnb.ac.id/index.php/matrix

Multi-class skin disease classification using transfer learning architectures

Muhammad Akhdaan [1]*, Majid Rahardi [2]

[1,2] *Informatics Study Program, Universitas Amikom Yogyakarta, Indonesia*

*Corresponding Author: *akhdaan@students.amikom.ac.id*

Abstract: Dermatological conditions are among the most common health problems worldwide, where delayed identification may increase disease severity and complicate treatment procedures. However, restricted access to dermatological expertise and insufficient public awareness often contribute to delayed diagnosis. This study proposes a multi-class skin disease classification approach using deep learning and transfer learning architectures based on digital skin images. The dataset, obtained from the *babaruzair/kaggle-skin-disease* repository, consists of 1,157 images categorized into eight skin disease classes. Image preprocessing techniques, including resizing, normalization, and data augmentation, were applied to improve data quality and model generalization. Three models were evaluated in this study, namely a baseline Convolutional Neural Network (CNN), MobileNetV2, and EfficientNetB0. Both transfer learning models utilized ImageNet pre-trained weights, followed by customized classification layers and fine-tuning of selected upper layers. Model performance was evaluated using accuracy, precision, recall, F1-score, and confusion matrix analysis. Experimental results show that the baseline CNN achieved an accuracy of 52.36%, while MobileNetV2 and EfficientNetB0 achieved accuracies of 88.84% and 95.28%, respectively. The findings demonstrate that transfer learning architectures significantly outperform conventional CNN models for limited medical image datasets, with EfficientNetB0 providing the best classification performance. These results indicate the potential of deep learning-based approaches to support early skin disease diagnosis and assist clinical decision-making.

Keywords: Skin diseases, CNN, MobileNetV2, EfficientNetB0

History Article: Submitted 2 February 2026 | Revised 13 April 2026 | Accepted 16 May 2026

How to Cite: M. Akhdaan and M. Rahardi, "*Multi-class skin disease classification using transfer learning architectures,*" *Matrix: Jurnal Manajemen Teknologi dan Informatika*, vol. 16, no. 2, pp. 117–128, 2026, doi: 10.31940/matrix.v16i2.117-128

Introduction

Dermatological disorders remain one of the most frequently encountered health problems worldwide, impacting approximately 1.8 billion individuals worldwide. According to the World Health Organization (WHO) in 2023, infections attributable to bacteria, fungi, and viruses constitute primary causes of morbidity in numerous countries [1]. In Indonesia, skin diseases remain among the most common public health problems and frequently require long-term medical treatment [2].

A common issue is the limited public understanding of skin health, causing skin diseases to often be diagnosed and treated only at advanced stages. This reliance on dermatologists, who are often limited in availability, especially in certain regions, leads to long waiting times for consultations and delayed diagnosis [3].

As digital technologies advance, artificial intelligence approaches in medical image analysis, particularly for medical image classification, offer promising opportunities for faster and more efficient diagnosis. Skin diseases exhibit visual characteristics that can be identified through digital images, making them suitable for analysis using image-processing approaches. Compared to other medical imaging methods such as histology, Magnetic Resonance Imaging (MRI), and Computer Tomography (CT), skin imaging is generally more accessible and cost-effective to obtain [4].

Convolutional Neural Networks (CNNs) have become dominant architectures for automated image analysis tasks in image processing because of their capability to learn hierarchical visual

representations directly from image data without manual intervention [5]. However, the performance of CNN models is highly influenced by the quantity and quality of training data. In practical applications, skin disease datasets are often limited and moderately imbalanced, which may negatively affect model generalization performance [6].

To address these challenges, transfer learning has emerged as an effective approach for medical image classification, particularly when dealing with limited datasets [7]. MobileNetV2 has been widely utilized in image classification tasks due to its lightweight architecture, computational efficiency, and competitive performance in transfer learning applications [8]. In addition, The EfficientNetB0 architecture has demonstrated superior classification capability through compound scaling strategies that optimize network depth, width, and image resolution simultaneously, enabling more effective feature extraction while maintaining parameter efficiency [8]. Several representative studies on deep learning-based skin disease classification are summarized in Table 1.

Therefore, this study presents a comparative analysis of three deep learning architectures, namely a baseline Convolutional Neural Network (CNN), MobileNetV2, and EfficientNetB0, for multi-class skin disease classification. Unlike previous studies that mainly focused on a single transfer learning architecture or specific skin lesion datasets, this study provides a comprehensive comparison between conventional CNN and two transfer learning models under identical preprocessing, training, and evaluation settings. Furthermore, customized classification layers, dropout regularization, and fine-tuning strategies are incorporated to improve feature adaptation for relatively limited and moderately imbalanced skin disease datasets. The scientific contributions of this study are threefold. First, it provides a fair comparative evaluation of conventional CNN, MobileNetV2, and EfficientNetB0 under identical experimental settings. Second, it demonstrates the effectiveness of transfer learning for multiclass skin disease classification using a relatively limited and moderately imbalanced dataset. Third, it provides benchmark results that can serve as a reference for future research on automated dermatological diagnosis.

Table 1. Comparison of previous studies on deep learning-based skin disease classification

Authors	Dataset	Model	Main Findings
Virupakshappa et al.	DermNet, PH2, HAM10000, ISIC	Enhanced MobileNetV2	Proposed an enhanced MobileNetV2 integrated with attention mechanisms and multi-scale feature extraction, demonstrating high performance across multiple public skin disease datasets.
Taşar	HAM10000	MobileNet, DenseNet121, ResNet50, VGG16, Xception	Compared several transfer learning architectures for multiclass skin lesion classification and showed that transfer learning models consistently outperformed conventional CNN models.
Harahap and Zulkifli	HAM10000	InceptionResNetV2	Developed a transfer learning-based skin lesion classification system and demonstrated that pretrained models can effectively improve multiclass skin lesion classification.
Dhuhita et al.	HAM10000	MobileNetV2	Demonstrated that MobileNetV2 can be applied for multiclass skin cancer classification using transfer learning, although further improvements are needed to enhance performance.

As presented in Table 1, previous studies have demonstrated that transfer learning architectures achieve promising performance in skin disease classification. Nevertheless, most studies focused on evaluating a single transfer learning architecture or specific skin lesion datasets

such as HAM10000, ISIC, PH2, and DermNet. Comprehensive comparisons between conventional CNN and multiple transfer learning architectures under identical preprocessing, training, and evaluation settings for multiclass skin disease classification remain limited. Therefore, a comparative evaluation is still required to better understand the effectiveness of different deep learning architectures for this task.

Methodology

This research was carried out in several phases, carefully structured to achieve its aims. It began with data collection and ended with evaluating the classification model's performance, as shown in Figure 1.

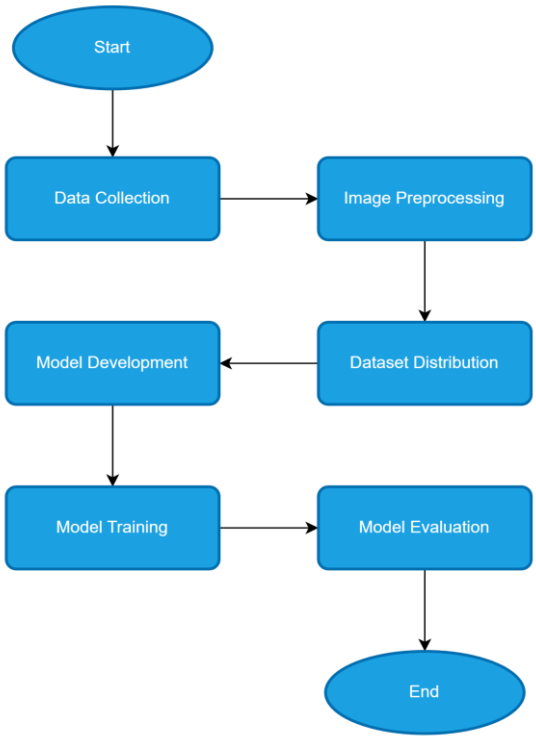


Figure 1. Research procedure

Data Collection

This study utilizes a publicly accessible dataset from Kaggle, titled 'kaggle-skin-disease.' This dataset comprises 1,157 images depicting various skin conditions, categorized into eight disease groups (Cellulitis, Impetigo, Athlete's Foot, Nail Fungus, Ringworm, Cutaneous Larva Migrans, Chickenpox, and Shingles). The distribution of images across each category in this study is outlined in Table 2.

Table 2. Distribution of skin disease images per class

No	Disease Category	Number of Images
1	Cellulitis	168
2	Impetigo	100
3	Athlete's Foot	156
4	Nail Fungus	162
5	Ringworm	113
6	Cutaneous Larva Migrans	125
7	Chickenpox	170
8	Shingles	163
	Total	1,157

Figure 2 shows examples of images in each class to provide a visual representation of the data used.

Figure 2. Image dataset sample

Image Preprocessing

Prior to model training, all skin disease images underwent preprocessing procedures to ensure consistent input representation and improve data quality. In this stage, all images were transformed into a uniform spatial resolution of 224×224 pixels to match the standard input dimensions required by the transfer learning architectures used in this study, namely MobileNetV2 and EfficientNetB0. In addition, pixel intensity values were normalized into a numerical range between 0 and 1 to support more stable optimization during the learning process [9].

Considering the relatively limited dataset size, data augmentation techniques were employed to increase data variability and improve the model's generalization capability [10]. Several augmentation techniques, including rotation, horizontal flipping, zooming, and width and height shifting, were applied to generate more diverse training samples. This approach was intended to reduce overfitting risk and enable the models to learn more robust image features from various skin disease patterns. Figure 3 illustrates an example of the image preprocessing and augmentation results applied in this study.

Figure 3. Image preprocessing results on one image

Model Development

In this study, three deep learning architectures were employed for multi-class skin disease classification, namely a baseline Convolutional Neural Network (CNN), MobileNetV2, and EfficientNetB0. The baseline CNN was utilized as a conventional deep learning benchmark without transfer learning, while MobileNetV2 and EfficientNetB0 were implemented using transfer learning approaches with ImageNet pre-trained weights. These architectures were selected to evaluate the effectiveness of transfer learning techniques on relatively limited and moderately imbalanced skin disease datasets. Both MobileNetV2 and EfficientNetB0 utilized ImageNet pre-trained weights, enabling the models to extract representative visual features such as textures, edges, and lesion patterns from skin disease images [11].

In the initial stage of model development, the base layers of the transfer learning architectures were frozen to preserve previously learned low-level visual features and prevent excessive modification of pre-trained weights during early training. This strategy was also intended to reduce overfitting risk considering the relatively limited dataset used in this study [12]. Subsequently, customized classification layers were added on top of the pre-trained base models to adapt the architectures for multi-class skin disease classification tasks, consisting of:

1. Input Layer
   RGB images with dimensions of 224×224 pixels were provided as input to the transfer learning architectures.
2. Base Model
   The pre-trained base models were initially frozen during the early training phase to preserve previously learned low-level visual features and reduce excessive modification of pre-trained weights.
3. Global Average Pooling 2D
   A Global Average Pooling layer was applied to reduce feature dimensionality before classification.
4. Dropout (0.5)
   Dropout regularization with a rate of 0.5 was introduced to reduce excessive neuron dependency during training.
5. Dense Layer (128)
   A fully connected dense layer with 128 neurons and ReLU activation was utilized to learn discriminative feature representations.
6. Dropout (0.3)
   A second dropout layer with a rate of 0.3 was added before the output layer to further improve regularization performance.
7. Output Layer
   The final classification layer employed Softmax activation to generate probability distributions across all skin disease categories.

After the initial feature extraction stage, selected upper layers of the pre-trained architectures were unfrozen and fine-tuned using a lower learning rate.

Model Training

Model optimization was performed using the Adam optimizer with an initial learning rate of 0.001 to support stable parameter updates during the training process. Preprocessing and augmentation procedures were exclusively applied to the training subset to improve model robustness and generalization capability while preventing data leakage. The validation subset was used to evaluate learning performance throughout the training stage, as presented in Table 3. Training was conducted using mini-batches consisting of 32 images, meaning that model weight updates were performed after processing every 32 images. In addition, Adaptive training callbacks, including EarlyStopping and ReduceLROnPlateau, were incorporated to reduce overfitting risk and improve training stability.

For the transfer learning architectures, the training procedure was conducted in two phases. Transfer learning optimization was performed in two sequential stages. Initially, the pre-trained backbone networks remained frozen while only the newly added classification layers were updated. Subsequently, selected upper layers of the pre-trained architectures were unfrozen and

fine-tuned using a lower learning rate of 0.0001 to improve adaptation toward dermatological image characteristics while retaining previously learned low-level visual representations [13].

Table 3. Division of train data, validation data, test data

No	Disease Category	Train (68.1%)	Validation (11.8%)	Test (20.1%)	Number of Images
1	Cellulitis	115	20	33	168
2	Impetigo	68	12	20	100
3	Athlete's Foot	105	19	32	156
4	Nail Fungus	110	19	33	162
5	Ringworm	77	13	23	113
6	Cutaneous Larva Migrans	85	15	25	125
7	Chickenpox	116	20	34	170
8	Shingles	111	19	33	163
	Total	788	136	233	1,157

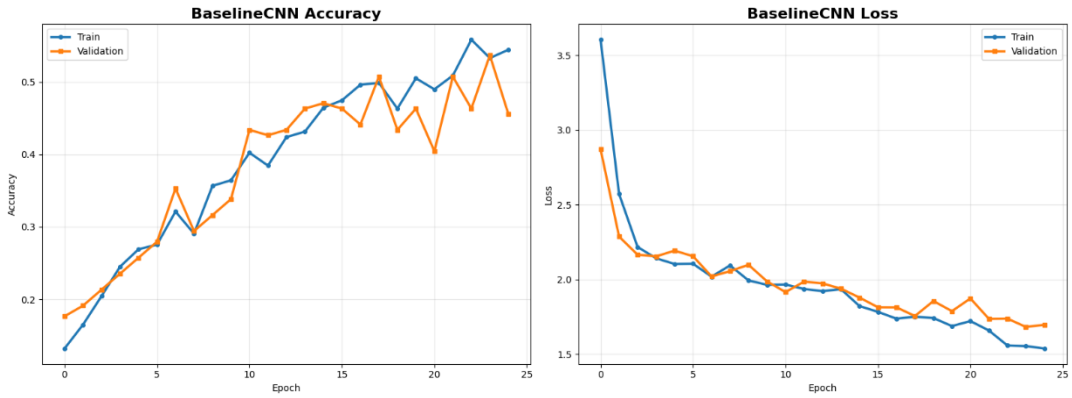
The dataset was divided into training, validation, and testing subsets with proportions of 68.1%, 11.8%, and 20.1%, respectively. Although these proportions differ slightly from commonly used split configurations such as 70:15:15 or 80:10:10, the distribution strategy was intentionally designed using a stratified approach to preserve class representation across all subsets [14]. Considering that the dataset used in this study was relatively limited and moderately imbalanced, maintaining proportional representation for each skin disease category was considered more important than strictly following conventional split ratios. In addition, allocating approximately 20% of the dataset for testing provided a sufficiently representative evaluation of the model's generalization performance on previously unseen data. This strategy was expected to reduce sampling bias and provide a more reliable and balanced assessment of classification performance across all skin disease categories [15].

Evaluation

Model evaluation was conducted using test data that had been separated from the training and validation datasets. The performance of each proposed architecture was evaluated using several classification metrics, including accuracy, precision, recall, and F1-score. In addition, confusion matrix analysis and classification reports were utilized to examine classification performance across each skin disease category. The evaluation results from the baseline CNN, MobileNetV2, and EfficientNetB0 models were further compared to analyze the effectiveness of transfer learning approaches for multi-class skin disease classification.

Results and Discussions

Results



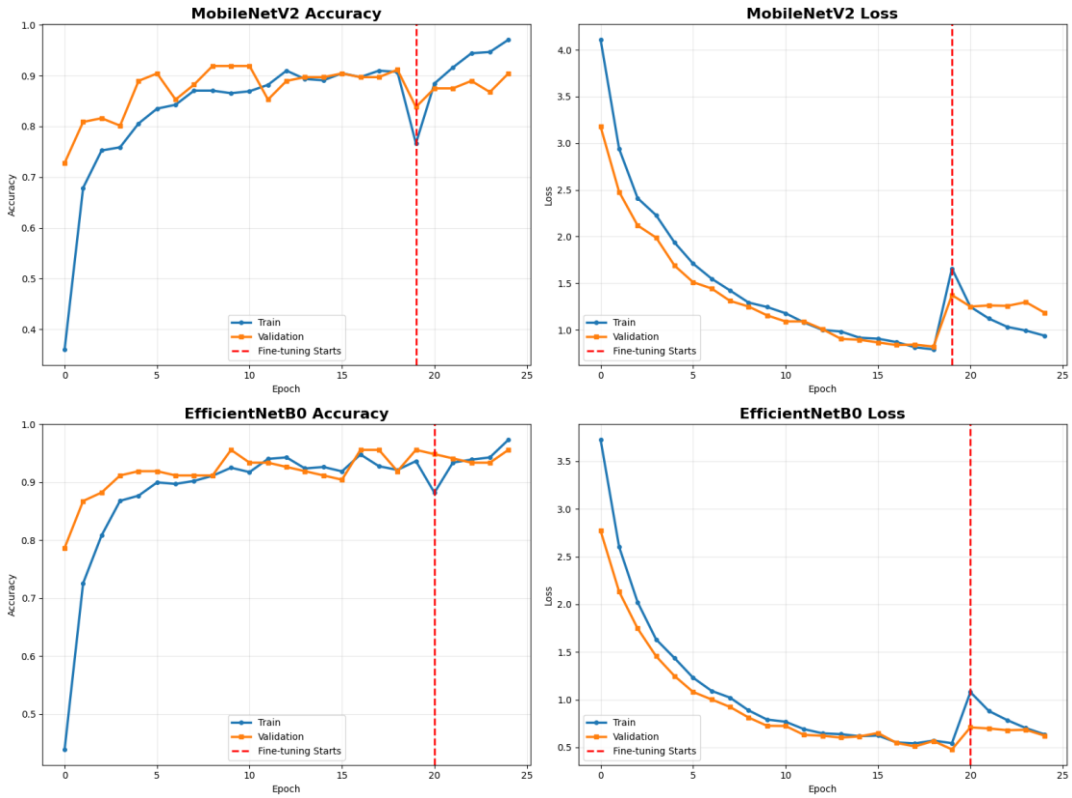


Figure 4. Model training result graph

Figure 4 illustrates the training and validation performance of the Baseline CNN, MobileNetV2, and EfficientNetB0 models during the learning process. The training and validation accuracy values showed a consistent increase during the early epochs before stabilizing in the later training stages. At the same time, both training and validation loss values gradually decreased, indicating that the model was able to effectively learn discriminative features from skin disease images. The relatively small gap between training and validation performance suggests that the implemented regularization techniques, data augmentation, and fine-tuning strategy contributed to reducing overfitting risk and improving model generalization capability. Furthermore, the stable convergence pattern indicates that EfficientNetB0 was able to extract representative image features efficiently, which may be attributed to its compound scaling strategy that balances network depth, width, and input resolution.

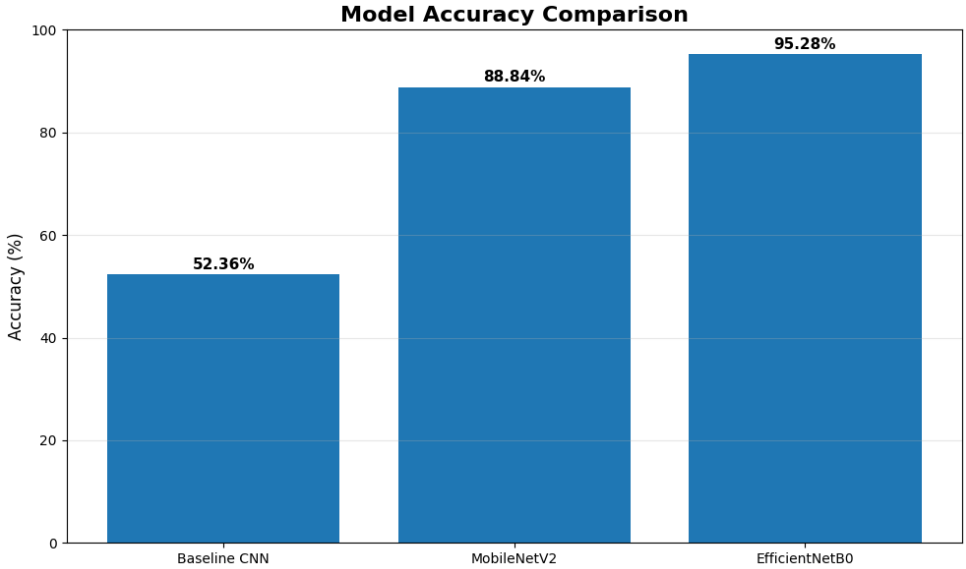


Figure 5. Model accuracy comparison

Figure 5 presents the comparison results of the proposed deep learning architectures for multi-class skin disease classification. The baseline CNN achieved an accuracy of 52.36%, while MobileNetV2 and EfficientNetB0 obtained accuracies of 88.84% and 95.28%, respectively. These results demonstrate that transfer learning architectures significantly outperformed the conventional CNN model on the limited skin disease dataset used in this study.

The relatively low performance of the baseline CNN indicates that conventional convolutional architectures without transfer learning experience difficulties in extracting robust discriminative features from limited and moderately imbalanced medical image datasets. In contrast, MobileNetV2 and EfficientNetB0 benefited from ImageNet pre-trained weights, enabling the models to utilize previously learned visual representations such as textures, shapes, and image patterns.

Among all evaluated architectures, EfficientNetB0 achieved the highest classification performance. This superior performance may be attributed to the compound scaling strategy employed by EfficientNetB0, which optimizes network depth, width, and image resolution simultaneously, allowing the model to extract more representative features from skin disease images. Although MobileNetV2 produced slightly lower accuracy compared to EfficientNetB0, it still demonstrated competitive performance while maintaining a relatively lightweight architecture suitable for resource-constrained applications.

The VI-chickenpox class achieved the highest performance with an F1 score of 90%, while FU-ringworm and VI-shingles scored 83%. This performance may still be considered reasonable considering the limited and moderately imbalanced dataset used in this study.

Table 4. Per-class F1-score comparison of Baseline CNN, MobileNetV2, and EfficientNetB0 models

Disesase Category	Baseline CNN	MobileNetV2	EfficientNetB0
Cellulitis	0.68	0.95	0.94
Impetigo	0.41	0.90	0.98
Athlete's Foot	0.60	0.90	0.98
Nail Fungus	0.52	0.95	1.00
Ringworm	0.38	0.82	0.88
Cutaneous Larva Migrans	0.44	0.76	0.93
Chickenpox	0.78	0.94	0.97
Shingles	0.12	0.88	0.93
Macro Avg	0.49	0.89	0.95
Weighted Avg	0.53	0.89	0.95

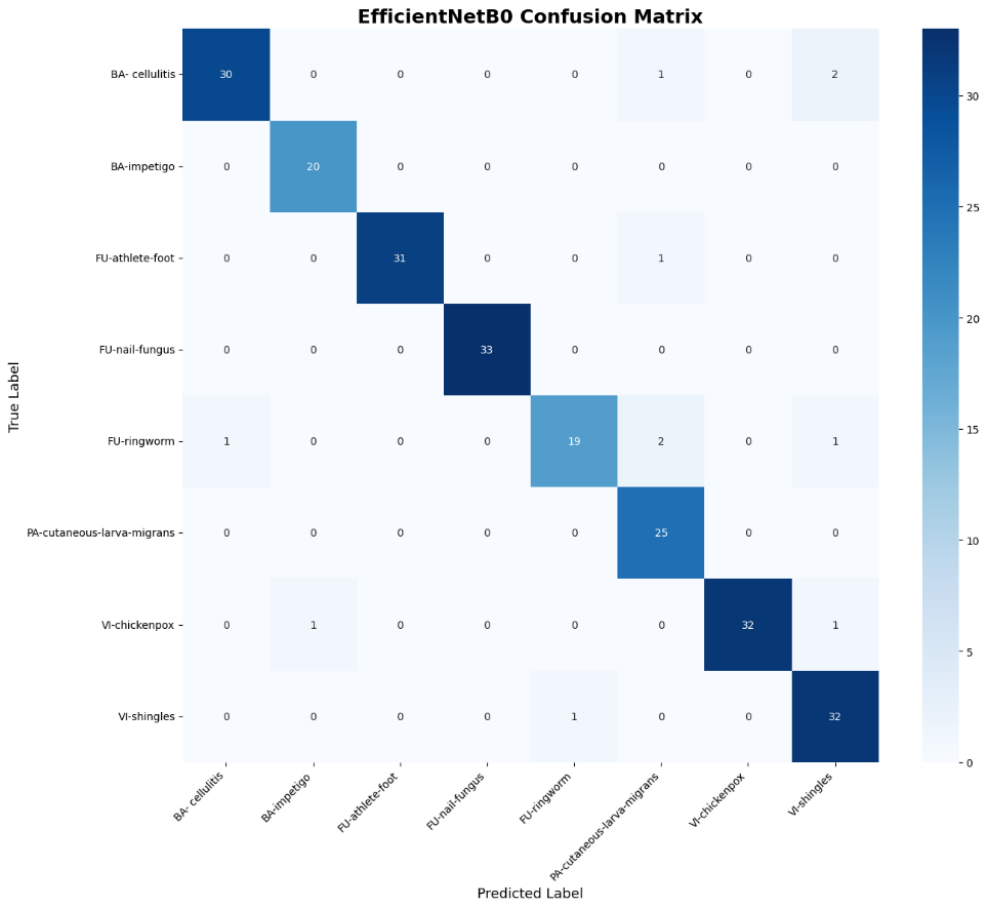


Figure 6. Confusion matrix

Table 4 presents the per-class F1-score comparison of the Baseline CNN, MobileNetV2, and EfficientNetB0 models, while Figure 6 illustrates the confusion matrix obtained from the EfficientNetB0 model. Based on the evaluation results, EfficientNetB0 achieved the highest classification performance across most skin disease categories, with weighted average precision, recall, and F1-score values of 0.96, 0.95, and 0.95, respectively. In contrast, the baseline CNN demonstrated substantially lower performance, particularly for visually similar disease categories, indicating limitations in extracting robust discriminative features from the relatively limited dataset. Meanwhile, MobileNetV2 showed considerable performance improvement compared to the baseline CNN, demonstrating the effectiveness of transfer learning approaches for skin disease classification.

The confusion matrix demonstrates that most predictions were concentrated along the diagonal region, indicating that the majority of skin disease images were classified correctly. Several classes, such as FU-nail-fungus, VI-chickenpox, and VI-shingles, achieved particularly high classification performance with minimal misclassification. These results indicate that the transfer learning approach combined with fine-tuning and customized classification layers was effective in extracting representative visual features from skin disease images.

Nevertheless, a small number of misclassifications were still observed between visually similar disease categories. This issue may have occurred due to similarities in skin texture, lesion shape, or color patterns among certain classes. In addition, although EfficientNetB0 achieved high classification accuracy, the relatively limited dataset size may still introduce potential overfitting and dataset bias. Therefore, further evaluation using larger and more diverse clinical datasets is necessary to validate the robustness and generalization capability of the proposed models in real-world applications.

Discussions

The experimental results demonstrate that transfer learning architectures significantly improved skin disease classification performance compared to the conventional baseline CNN model. The baseline CNN achieved relatively low accuracy due to its limited capability in extracting robust feature representations from a relatively small and moderately imbalanced dataset. In contrast, MobileNetV2 and EfficientNetB0 benefited from ImageNet pre-trained weights, enabling the models to utilize previously learned visual representations such as textures, edges, shapes, and lesion patterns. This transfer learning strategy substantially improved feature extraction capability and accelerated convergence during training.

Among the evaluated architectures, EfficientNetB0 achieved the highest classification performance with an accuracy of 95.28%, outperforming MobileNetV2 and the baseline CNN. The superior performance of EfficientNetB0 may be attributed to its compound scaling approach, which balances network depth, width, and image resolution simultaneously [16]. This mechanism enables the model to capture more representative and discriminative image features while maintaining efficient parameter utilization. Meanwhile, MobileNetV2 also demonstrated competitive performance with an accuracy of 88.84%, indicating that lightweight transfer learning architectures remain effective for skin disease classification tasks, particularly in resource-constrained environments [17].

The confusion matrix analysis further indicates that most skin disease categories were classified correctly, although several misclassifications were still observed between visually similar classes. Such classification errors may occur because certain skin diseases share similar visual characteristics, including lesion texture, color distribution, and skin surface patterns [18].

Despite the promising results obtained in this study, several limitations should be acknowledged. The dataset consisted of only 1,157 images distributed across eight classes, resulting in relatively limited training data for deep learning models. Although data augmentation, dropout regularization, and fine-tuning strategies were implemented to reduce overfitting risk, the possibility of dataset bias and limited generalization capability may still exist. Furthermore, the dataset used in this study originated from publicly available image collections and may not fully represent real-world clinical variations. Therefore, future studies are encouraged to utilize larger and more diverse clinical datasets, explore additional transfer learning architectures, and investigate explainable artificial intelligence (XAI) techniques to improve model interpretability and reliability in practical healthcare applications.

Conclusion

This research evaluated the comparative performance of conventional CNN and transfer learning architectures for multi-class skin disease classification. The experimental results demonstrated that transfer learning architectures significantly outperformed the conventional CNN model on a relatively limited and moderately imbalanced skin disease dataset. Among the evaluated models, EfficientNetB0 achieved the highest classification performance with an accuracy of 95.28%, followed by MobileNetV2 with 88.84% accuracy.

The findings indicate that transfer learning, combined with image preprocessing, data augmentation, dropout regularization, and fine-tuning strategies, effectively improved feature extraction capability and model generalization performance for skin disease classification tasks. Furthermore, the results demonstrate the effectiveness of transfer learning architectures in extracting representative dermatological features from limited and moderately imbalanced datasets. In addition, confusion matrix analysis showed that most skin disease categories were classified correctly, although several misclassifications still occurred among visually similar classes.

Despite the strong classification performance achieved in this study, several limitations remain. The dataset consisted of only 1,157 images distributed across eight disease categories, which may still introduce overfitting risk and limit model generalization capability in real-world clinical scenarios. Therefore, future studies are encouraged to utilize larger and more diverse clinical datasets, investigate additional transfer learning architectures, and explore explainable artificial intelligence (XAI) approaches to improve model interpretability and reliability for practical healthcare applications [19].

References

[1] World Health Organization, "WHO's first global meeting on skin NTDs calls for greater efforts to address their burden," 2023. [Online]. Available: https://www.who.int/news/item/31-03-2023-who-first-global-meeting-on-skin-ntds-calls-for-greater-efforts-to-address-their-burden

[2] N. I. Khani and S. Rakasiwi, "Penerapan Convolutional Neural Network dengan ResNet-50 untuk Klasifikasi Penyakit Kulit Wajah Efektif," *Edumatic*, vol. 9, no. 1, pp. 217–225, Apr. 2025, doi: 10.29408/edumatic.v9i1.29572.

[3] D. Gunawan and H. Setiawan, "Convolutional Neural Network dalam Citra Medis," *KONSTELASI*, vol. 2, no. 2, May 2022, doi: 10.24002/konstelasi.v2i2.5367.

[4] C. Chen, N. A. Mat Isa, and X. Liu, "A review of convolutional neural network based methods for medical image classification," *Computers in Biology and Medicine*, vol. 185, p. 109507, Feb. 2025, doi: 10.1016/j.compbiomed.2024.109507.

[5] A. W. Salehi *et al.*, "A Study of CNN and Transfer Learning in Medical Imaging: Advantages, Challenges, Future Scope," *Sustainability*, vol. 15, no. 7, p. 5930, Mar. 2023, doi: 10.3390/su15075930.

[6] I. Isewon, E. Alagbe, and J. Oyelade, "Optimizing machine learning performance for medical imaging analyses in low-resource environments: The prospects of CNN-based Feature Extractors," *F1000Res*, vol. 14, p. 100, Jan. 2025, doi: 10.12688/f1000research.156122.1.

[7] J. Chae and J. Kim, "An Investigation of Transfer Learning Approaches to Overcome Limited Labeled Data in Medical Image Analysis," *Applied Sciences*, vol. 13, no. 15, p. 8671, Jul. 2023, doi: 10.3390/app13158671.

[8] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in International Conference on Machine Learning (ICML), 2019, pp. 6105–6114.

[9] J. B. Ruhland, I. Masoudian, and D. Heider, "Enhancing deep neural network training through learnable adaptive normalization," *Knowledge-Based Systems*, vol. 326, p. 113968, Sep. 2025, doi: 10.1016/j.knosys.2025.113968.

[10] T. Kumar, R. Brennan, A. Mileo, and M. Bendechache, "Image Data Augmentation Approaches: A Comprehensive Survey and Future Directions," *IEEE Access*, vol. 12, pp. 187536–187571, 2024, doi: 10.1109/ACCESS.2024.3470122.

[11] C. Gu and M. Lee, "Deep Transfer Learning Using Real-World Image Features for Medical Image Classification, with a Case Study on Pneumonia X-ray Images," *Bioengineering*, vol. 11, no. 4, p. 406, Apr. 2024, doi: 10.3390/bioengineering11040406.

[12] A. Davila, J. Colan, and Y. Hasegawa, "Comparison of fine-tuning strategies for transfer learning in medical image classification," *Image and Vision Computing*, vol. 146, p. 105012, Jun. 2024, doi: 10.1016/j.imavis.2024.105012.

[13] H. E. Kim, A. Cosa-Linan, N. Santhanam, M. Jannesari, M. E. Maros, and T. Ganslandt, "Transfer learning for medical image classification: a literature review," *BMC Med Imaging*, vol. 22, no. 1, p. 69, Dec. 2022, doi: 10.1186/s12880-022-00793-7.

[14] J. Sadaiyandi, P. Arumugam, A. K. Sangaiah, and C. Zhang, "Stratified Sampling-Based Deep Learning Approach to Increase Prediction Accuracy of Unbalanced Dataset," *Electronics*, vol. 12, no. 21, p. 4423, Oct. 2023, doi: 10.3390/electronics12214423.

[15] J.-X. Zhuang, J. Cai, J. Zhang, W. Zheng, and R. Wang, "Class attention to regions of lesion for imbalanced medical image recognition," *Neurocomputing*, vol. 555, p. 126577, Oct. 2023, doi: 10.1016/j.neucom.2023.126577.

[16] C. Lin, P. Yang, Q. Wang, Z. Qiu, W. Lv, and Z. Wang, "Efficient and accurate compound scaling for convolutional neural networks," *Neural Networks*, vol. 167, pp. 787–797, Oct. 2023, doi: 10.1016/j.neunet.2023.08.053.

[17] J. P. Nyakuri, C. Nkundineza, O. Gatera, and K. Nkurikiyeyezu, "Tiny-MobileNet-SE: A Hybrid Lightweight CNN Architecture for Resource-Constrained IoT Devices," *IEEE Access*, vol. 13, pp. 108389–108401, 2025, doi: 10.1109/ACCESS.2025.3582055.

[18] M. O. Sarı and K. Keser, "Classification of skin diseases with deep learning based approaches," *Sci Rep*, vol. 15, no. 1, p. 27506, Jul. 2025, doi: 10.1038/s41598-025-13275-x.

[19] J. Gupta and K. R. Seeja, "A Comparative Study and Systematic Analysis of XAI Models and their Applications in Healthcare," *Arch Computat Methods Eng*, Apr. 2024, doi: 10.1007/s11831-024-10103-9.

POLITEKNIK NEGERI BALI

9 772580 563008

Redaksi Jurnal Matrix
Gedung P3M, Politeknik Negeri Bali
Bukit Jimbaran, PO BOX 1064 Tuban, Badung, Bali.
Phone: +62 361 701981, Fax: +62 361 701128
e-mail: p3mpoltekbali@pnb.ac.id
https://ojs2.pnb.ac.id/index.php/MATRIX