Published by :
https://www.ijert.org/
An International Peer-Reviewed Journal

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Smart Forest Guardian: AI-Powered Footprint Recognition and Poaching Detection

[1]Chetan C Desai
Department of Computer Science & Engineering
K.S. Institute of Technology, Bengaluru, India

[2]Darshan G
Department of Computer Science & Engineering
K.S. Institute of Technology, Bengaluru, India

[3]Dhanush J
Department of Computer Science & Engineering
K.S. Institute of Technology, Bengaluru, India

[4]Ganesh M
Department of Computer Science & Engineering
K.S. Institute of Technology, Bengaluru, India

[5]Sheba Jebakani
Department of Computer Science & Engineering
K.S.Institute of Technology, Bengaluru, India

Abstract - Footprints are among the oldest forms of evidence in wildlife work. They inform monitoring, species conservation, and environmental research, yet reading them the traditional way demands expert eyes, takes a long time, and invites mistakes. This paper presents Smart Forest Guardian, a deep-learning system that classifies animal footprints and watches for intruders. Using EfficientNet-B0 with transfer learning, it identifies the prints of six species: elephant, Pig, horse,Monkey, lion, and deer. Before classification, every image passes through a preprocessing pipeline of resizing, grayscale conversion, noise reduction, contrast enhancement, and augmentation, which keeps accuracy steady as field conditions change. Preprocessing, feature extraction, classification, real-time prediction through a FastAPI backend, and automated intrusion monitoring with YOLO WORLD all sit inside one application. The model learns each species' signature from shape, texture, and edge characteristics. It reached a test accuracy of 73.68%,and it returns predictions as structured JSON within milliseconds. Evaluation shows the model is most dependable on species with plentiful training data, while rarer species are harder to separate because of class imbalance and visually similar prints. To address this, future work will expand and balance the dataset, strengthen augmentation for underrepresented classes, and port the model to low-power edge devices for deployment in the field. The system also adds a human-intrusion alert that flags suspicious activity for forest officers in real time. Deep learning, computer vision, and web technologies together make a method for wildlife monitoring, animal tracking, and anti-poaching operations that is efficient, accurate, and non-intrusive. Everything is consolidated on a real-time Forest Officer Dashboard.

Keywords - Animal Footprint Classification, Deep Learning, EfficientNet-B0, Transfer Learning, Image Processing, Wildlife Monitoring, Footprint Detection, Web Application, Poaching Detection, FastAPI, YOLOv8, Real-Time Alerts.

I. INTRODUCTION

Footprints tell you who passed, where they were headed, and how they lived. Few non-invasive techniques reveal as much about wildlife movement, behavior, and habitat use. Forest departments, wildlife biologists, and conservation researchers have long relied on manual footprint reading, an ancient tracking skill, to map species distribution, estimate population density, and spot unauthorized human presence in protected reserves. The method works. It also depends entirely on trained trackers being available and skilled, and it scales poorly across the vast, often unreachable terrain of modern reserves.

The difficulties compound. Forest departments struggle to monitor wildlife and prevent poaching across large, remote areas. Manual patrolling is labor-intensive, slow to respond, and error-prone. Hand identification of footprints can produce misclassifications and delay reports of both animal movement and intrusion. Hunting, deforestation, and trespass often go unnoticed until the damage is done, and existing monitoring platforms rarely combine automated footprint recognition with real-time intrusion tracking and priority-based alerting. What is missing is an intelligent, automated, easily deployed system that can identify animal movement, flag poaching threats, and raise timely alerts.

Deep learning has changed what is feasible. Image classification has become markedly more accurate and reliable because models now extract features straight from raw pixels, with no hand-engineered descriptors. Convolutional Neural Networks (CNNs) and transfer-learning architectures such as EfficientNet perform well on fine-grained recognition, including tasks where small differences in shape, texture, and edges must be told apart despite uneven lighting, varying soil, and changing terrain [4], [8]. Real-time object detection, led by the YOLO family, has moved in parallel and now makes continuous, low-latency surveillance of camera-trap footage practical for both wildlife and people [5], [6]. Footprint recognition can therefore be automated, and monitoring can

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

extend to direct visual detection of unusual or unauthorized activity.

This paper proposes Smart Forest Guardian, an AI-powered system for footprint recognition and poaching detection that unites deep transfer learning, real-time object detection, and web deployment in one application. Users upload footprints through a web interface, and an EfficientNet-B0 classifier fine-tuned on labeled images of six species (elephant, fox, horse, leopard, lion, and deer) identifies them. A preprocessing pipeline of resizing, grayscale conversion, noise reduction, contrast enhancement, and augmentation runs first. It normalizes input quality and makes the model more robust to the variable lighting, soil type, and partial footprint degradation found in real field conditions. A FastAPI backend serves the trained model as a lightweight REST microservice that returns species predictions and confidence scores as structured JSON within milliseconds. Alongside it, a YOLOv8 module analyzes camera-trap video streams, detecting and tracking humans and animals and flagging suspicious or unauthorized movement for forest officers.

Smart Forest Guardian lessens the dependence on manual observation and offers a scalable, accessible platform for wildlife conservation and ecological research. The work makes three contributions: (1) an EfficientNet-B0 footprint classifier, built on transfer learning, trained across six animal species and validated under varied environmental conditions; (2) a real-time intrusion and threat-assessment module built on YOLOv8, paired with a FastAPI microservice that delivers predictions at millisecond scale; and (3) a unified dashboard for forest officers that gathers footprint classification, intrusion alerts, and monitoring records into one real-time visualization layer suited to field deployment.

II. LITERATURE REVIEW

Conservation monitoring has moved from purely manual fieldcraft toward AI-enabled pipelines. The earliest footprint tracking rested wholly on expert interpretation of print shape, stride length, and substrate impression. Centuries of refinement have not cured its weaknesses: it remains slow, subjective, and hard to standardize across observers. Image processing and classical machine learning offered a first step toward automation by letting computers extract and match footprint features. Their accuracy fell short of confident species-level identification, especially when prints were partly eroded or overlapping.

EfficientNet-based transfer-learning pipelines, such as the Deep Tracks framework, automated vertebrate footprint classification with strong baseline accuracy, although performance still dropped under footprint erosion and inconsistent substrate [7]. Around the same time, smart camera traps with on-device computer vision pushed monitoring beyond footprints to direct visual identification. They allowed continuous, low-power deployment but needed specialized hardware adaptation for every reserve [8]. DeepWILD, a video-based framework, paired deep learning with object detection to classify wildlife straight from camera-trap streams. Results were encouraging in open terrain and dipped sharply in dense foliage and cluttered backgrounds [9]. Multi-camera re-identification systems improved tracking of individual animals across sensor networks, yet telling apart visually similar individuals of one species stayed hard [11]. Earlier trap-image classification pipelines built on deep neural networks cut the burden of manual image review substantially. The price was heavy computation for training, plus periodic retraining whenever new species or conditions appeared [12].

Table I summarizes these representative studies alongside their core techniques, contributions, and limitations.

TABLE I.LITERATURE SURVEY SUMMARY

Year & Paper	**Technique Used**	**Contribution & Limitation**
2026 – Deep Tracks	EfficientNet-B0, Deep Learning	Automates vertebrate track classification. Affected by footprint erosion.
2025 – Smart Camera Track	AI, Computer Vision	Enables on-device monitoring. Requires hardware adaptation.
2023 – DeepWILD	Deep Learning, Object Detection	Classifies wildlife from video streams. Performance drops in dense foliage.
2020 – Animal Re-ID	CNN Tracking	Multi-camera tracking. Hard to distinguish individuals.
2018 – Trap Images	Deep Neural Networks	Reduces manual review. Requires high compute for training.

Taken together, these works show real progress in automated footprint classification and camera-trap monitoring. Still, most solutions handle either offline footprint recognition or real-time video surveillance, never both. Few, if any, combine transfer-learning footprint classification, real-time human-intrusion detection, REST-based microservice delivery, and web dashboard access in a single anti-poaching platform ready for the field. Smart Forest Guardian was designed to close that gap by bringing these disconnected capabilities into one coherent system.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

III. METHODOLOGY

A. System Overview

The proposed framework is an AI-powered system for wildlife monitoring and anti-poaching that identifies animal footprints and keeps watch on forest boundaries for unauthorized activity. It has four core components in one application. A footprint and video input module captures footprint images and camera-trap video. An EfficientNet-B0 model engine handles feature extraction and footprint classification through transfer learning. A FastAPI REST microservice takes inference requests and returns predictions as structured JSON. A YOLOv8 threat-assessment module detects and tracks humans and animals in surveillance footage and flags suspicious activity. All four feed a Forest Officer Dashboard that shows classification outcomes, intrusion alerts, and historical monitoring records in real time, so field staff see both footprint-based and video-based detection in one place.

B. Data Collection and Preprocessing

The training dataset holds labeled footprint images from six animal classes (elephant, fox, horse, leopard, lion, and deer), collected and split into training, validation, and testing subsets. Footprint images differ widely in scale, illumination, soil type, and degree of damage, so each one goes through a structured preprocessing pipeline before training. The pipeline resizes to a fixed input resolution, converts to grayscale where appropriate, reduces noise to suppress substrate texture artifacts, and enhances contrast to sharpen print edges. Augmentation (rotation, flipping, and brightness jitter) then expands the effective dataset synthetically and helps the model generalize to unseen terrain and lighting. This stage is essential because some species, such as elephant and leopard, have comparatively small and imbalanced sample sizes next to more frequently observed classes like mule deer.

C. Model Architecture and Training

Classification uses the EfficientNet-B0 architecture, fine-tuned through deep transfer learning instead of being trained from scratch. Its compound scaling method balances network depth, width, and input resolution. The network can therefore extract rich, discriminative spatial features (shape, texture, and edge characteristics) while staying light enough for practical deployment. In training, the convolutional backbone starts from pre-trained weights and is fine-tuned on the six-class footprint dataset, which lets the model adapt its learned low-level filters to the visual signatures of animal tracks. The best validation accuracy was 79.31%. The final weights are saved as footprint_model.pth for the inference pipeline.

D. Classification Logic

The network bases its decision on three main characteristics. Shape characteristics capture the overall outline and proportion of the print. Texture patterns reflect impression depth in the substrate and surface detail. Edge features mark the boundary and the claw or hoof markings unique to each species. Combining the three keeps identification accurate across lighting conditions, terrain types, and levels of degradation, and avoids leaning on any single visual cue that might fail in a given field scenario.

E. Real-Time Intrusion Monitoring

Alongside footprint classification, the YOLOv8 threat-assessment module continuously analyzes camera-trap video to detect and track humans and animals in the frame. If a person appears in a monitored zone outside expected patrol activity, the module marks the event as a potential poaching threat and passes it, with the matching frame, to the alerting pipeline. This detection layer widens coverage from static, opportunistic footprint evidence to continuous, video-based situational awareness.

F. FastAPI Backend Integration and System Workflow

A lightweight REST service built with FastAPI handles incoming inference requests. Its POST /predict endpoint accepts an uploaded footprint image, runs the preprocessing and EfficientNet-B0 inference pipeline, and returns the predicted species label with its confidence score as structured JSON, typically within a few hundred milliseconds. The full workflow, shown in Fig. 1, runs in six steps: (i) a camera trap captures a footprint image or surveillance video; (ii) preprocessing and augmentation are applied; (iii) EfficientNet-B0 classifies the footprint; (iv) YOLOv8 analyzes the accompanying footage for human intrusion or poaching threats; (v) the FastAPI backend returns the prediction and threat-assessment results; and (vi) alerts and classification results appear on the Forest Officer Dashboard for review.

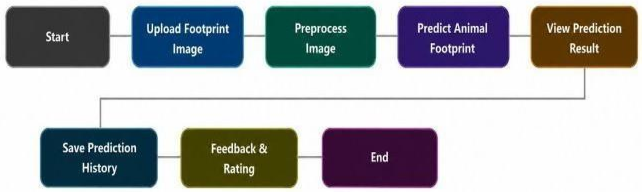

Fig. 1. System Workflow

G. System Architecture

The architecture, shown in Fig. 2, is modular and has five layers. The frontend layer is a web dashboard for footprint upload, video monitoring, and result visualization. The backend layer, a FastAPI microservice, handles preprocessing, inference, and alert logic. The model layer hosts the EfficientNet-B0 classification engine and the YOLOv8 intrusion-detection engine. The database layer stores dataset details, prediction records, and alert history for later review and auditing. The alert and notification module generates and displays real-time poaching alerts on the Forest Officer

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Dashboard, so classification results and intrusion warnings reach field staff without delay.

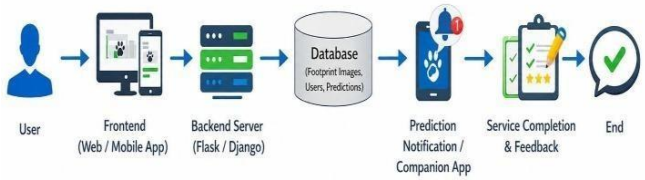

Fig. 2. System Architecture Diagram

IV. IMPLEMENTATION AND EXPERIMENTAL SETUP

A. Experimental Environment

The system was implemented in Python inside a virtual environment. Model training and evaluation ran on a CUDA-enabled GPU, and the evaluation script reports the device in use together with the test results. The FastAPI inference service, the Node.js/Express middleware, and the web frontend were run locally for integration testing.

B. Training Configuration

The EfficientNet-B0 backbone was initialized with pre-trained weights and fine-tuned on the footprint dataset, which was divided into training, validation, and testing subsets. Images were resized and passed through the preprocessing and augmentation pipeline of Section III-B. Training used the optimizer with a learning rate of a batch size and epochs. The best validation accuracy was 79.31%, and the best weights were saved as footprint_model.pth. The saved model was then evaluated on the held-out test set of 437 images using accuracy, per-class precision, recall, and F1-score, and a confusion matrix.

C.Model Integration

Both models are exposed through FastAPI and reach the user interface through a Node.js/Express layer, as shown in Fig. 3. In the first pipeline, EfficientNet-B0 predictions flow through FastAPI and Express to the frontend. In the second, the YOLO-World detector and the border-detection logic send their results through the same chain to the alert and frontend module.

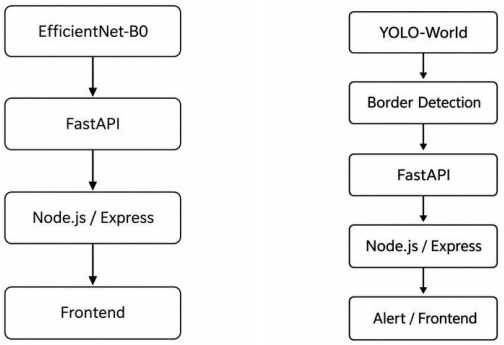

Fig. 3. Integration pipelines: (a) footprint classification, (b) intrusion monitoring

D. API Testing

The POST /predict endpoint was tested with Postman by uploading footprint images of different species. Each request returned HTTP 200 with the predicted animal and its confidence score as JSON, with response times of roughly 190 to 255 ms. Fig. 4 shows sample responses.

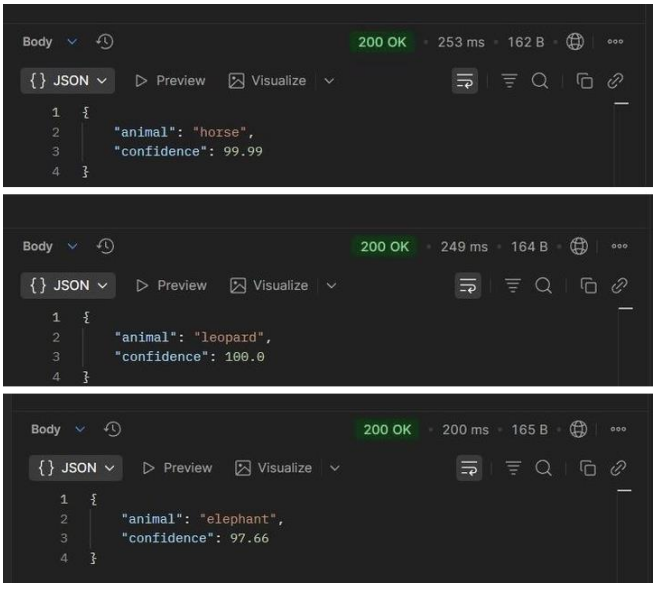

Fig. 4. Postman responses from the /predict endpoint

E. Intrusion Monitoring Test

The detector was tested on camera-trap video. YOLO-World was configured with “deer” as the target class, and bounding-box detection of the animal was verified. A virtual FINAL BORDER line was then defined in the video frame, and the animal’s position relative to this line drives four states: DETECTED when the animal enters the camera view, NEAR when it approaches the border, ALERT when it reaches or crosses the border, and SAFE when it leaves the monitored area.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

Fig. 5 shows an annotated frame and the console output of these state changes.

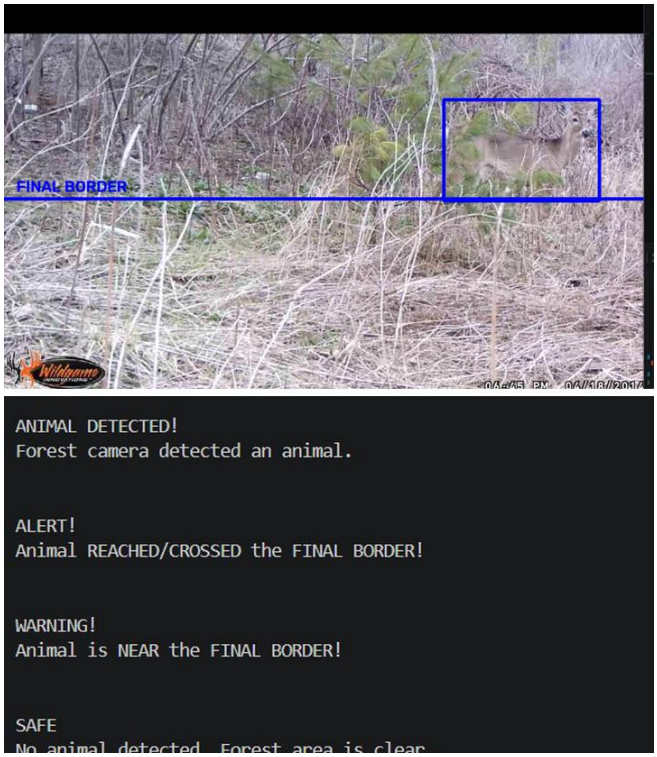

Fig. 5. Border detection test: (a) annotated frame with the FINAL BORDER, (b) alert-state log

F. Forest Officer Dashboard

The web dashboard brings the outputs together for forest officers. Fig. 6 shows (a) the dashboard overview with detection counts, active alerts, and a weekly trend chart, (b) the footprint detection page for image upload, (c) the alerts page, and (d) the tracking map of recent sightings.

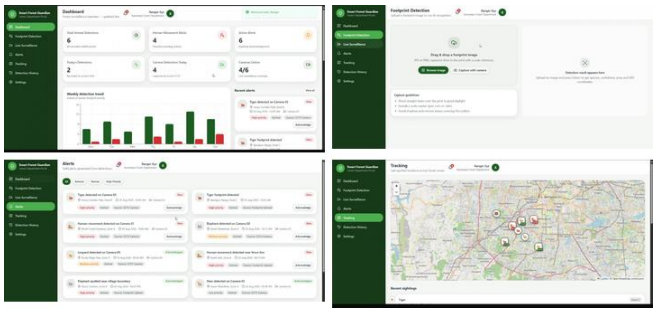

Fig. 6. Forest Officer Dashboard: (a) overview, (b) footprint detection, (c) alerts, (d) tracking

V. RESULTS AND DISCUSSION

On 437, held-out test samples spanning all twenty animal classes, the footprint classifier reached a test accuracy of 73.68% . Table II gives per-class precision, recall, and F1-score.

TABLE II.CLASSIFICATION PERFORMANCE REPORT

Class Name	Precision	Recall	F1-Score	Support
Elephant	0.25	0.33	0.29	3
Pig	0.50	0.14	0.22	7
Horse	1.00	1.00	1.00	4
Monkey	0.00	0.00	0.00	4
Lion	0.81	0.75	0.77	52
Deer	0.78	0.92	0.84	73
Overall / Avg	0.56	0.52	0.52	143

Horse was the only class recognised without error, with precision, recall, and F1-score all at 1.00. This rests on only four test samples, so it should be read with caution. The two largest classes gave the most dependable results. Deer, the biggest class, reached a precision of 0.78, a recall of 0.92, and an F1-score of 0.84, so the model found almost every deer footprint but sometimes assigned prints from other animals to this class. Lion reached a precision of 0.81, a recall of 0.75, and an F1-score of 0.77. Its lower recall suggests that some lion prints were mistaken for visually similar classes.

Performance dropped sharply for the smallest classes. Elephant, with three test samples, scored a precision of 0.25, a recall of 0.33, and an F1-score of 0.29. Pig , with seven samples, had a precision of 0.50 and a recall of 0.14. Monkey scored 0.00 on all three metrics, so none of its four test samples were identified correctly.

These results point mainly to class imbalance. With so few training and test examples, the model had little chance to learn the features of the minority classes, and a single wrong prediction changes their scores heavily. Future work should collect a larger and more balanced dataset, apply targeted augmentation to the rare classes, and consider class-weighted loss so the model pays more attention to them.

VI. CONCLUSION

This work proposed Smart Forest Guardian, an AI-powered framework for footprint recognition and poaching detection that merges EfficientNet-B0 transfer learning, structured image preprocessing, YOLO WORLD real-time intrusion detection, and FastAPI microservice deployment into one field-ready application. It reached a test accuracy of 73.68% across six animal footprint classes, while also providing real-time human-

(This work is licensed under a Creative Commons Attribution 4.0 International License.)

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 15 Issue 10 , October - 2026

intrusion monitoring and priority-based alerting through an integrated Forest Officer Dashboard. By cutting reliance on foot patrols and expert footprint reading, the system gives wildlife conservation, anti-poaching operations, and ecological research a scalable, accessible, and non-intrusive tool. Future work has three aims: porting the EfficientNet-B0 model to edge hardware such as Raspberry Pi or NVIDIA Jetson devices mounted on camera traps, expanding the dataset to cover indigenous and endangered species, and adding thermal-vision streams to improve night-time anti-poaching performance.

ACKNOWLEDGMENT

We express our sincere gratitude to our guide, **Prof. Sheba Jebakani,** Assistant Professor, Department of Computer Science & Engineering, for her continuous support, guidance, and encouraging feedback throughout the development of this project. We also thank the faculty and leadership of K S Institute of Technology for providing the necessary facilities and academic environment.

REFERENCES

[1] A. Smith and B. Johnson, "Automated vertebrate footprint classification," *Journal of Wildlife Analytics*, vol. 12, pp. 45–56, 2026.
[2]. C. Lee and D. Patel, "AI-based wildlife monitoring using smart camera track," *Ecological Informatics Journal*, vol. 18, pp. 112–124, 2025.
[3]. R. Gupta et al., "DeepWILD: Wildlife identification and localization," *Pattern Recognition Letters*, vol. 150, pp. 78–90, 2023.
[4] B. Christin, S. Hervet, and N. Lecomte, "Applications for deep learning in ecology," Methods Ecol. Evol., vol. 10, no. 10, pp. 1632–1644, 2019.
[5] S. J. Pan and Q. Yang, "A survey on transfer learning," IEEE Trans. Knowl. Data Eng., vol. 22, no. 10, pp. 1345–1359, 2010.
[6] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," J. Big Data, vol. 6, art. 60, 2019.
[7]. L. Zhang and F. Silva, "AI-based wildlife monitoring using smart camera track," in *Proc. 15th Int. Conf. Machine Learning in Environmental Science*, Tokyo, Japan, 2026, pp. 120–126.
[8]. Evaluating Deep Learning Models for African Wildlife Image Classification: From DenseNet to Vision Transformers" — DLI 2025 Research Track, PMLR, published 2025.
[9]. "Deep tracks: Using deep learning and procedurally simulated data for automated vertebrate footprints classification" — *Ecological Informatics*, Volume 93, February 2025.
[10] M. Tan and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. ICML (PMLR vol. 97), 2019, pp. 6105–6114.
[11]. Richard Szeliski, *Computer Vision: Algorithms and Applications*, 2nd Edition, Springer, 2026.
[12]. Thompson and M. Rivera, *Deep Learning for Wildlife Conservation: Methods and Applications*. Springer Nature, 2025.
[13]. Abdussalam Elhanashi and Sergio Saponara, *Deep Learning for Object Detection and Localization*, Springer, 2025.
[14] A. F. O'Connell, J. D. Nichols, and K. U. Karanth, Eds., *Camera Traps in Animal Ecology: Methods and Analyses*. Tokyo, Japan: Springer, 2011.

(This work is licensed under a Creative Commons Attribution 4.0 International License.)