VOLUME 17 SPECIAL ISSUE 1, JULY 2026
International Journal of Advanced Research in Computer Science
RESEARCH PAPER
Available Online at www.ijarcs.info
ISSN No. 0976-5697

AI-Powered Mobile Solution For Livestock Management for Intelligent Cattle Breed Classification System

Upasana Ameta
Geetanjali Institute of Technical Studies
Udaipur, Rajasthan, India
ametaupasana@gmail.com

Dr. Mayank Patel
Geetanjali Institute of Technical Studies
Udaipur, Rajasthan, India
mayank.patel@gits.ac.in

Priyank Nandawat
Geetanjali Institute of Technical Studies
Udaipur, Rajasthan, India
priyanknandawat7@gmail.com

Rajdeep Singh
Geetanjali Institute of Technical Studies
Udaipur, Rajasthan, India
rajdeepsingh7240@gmail.com

Laksh Tyagi
Geetanjali Institute of Technical Studies
Udaipur, Rajasthan, India
lakshtyagi25@gmail.com

Abstract: This paper presents CattleSense, an intelligent mobile application for automatic classification of Indian cattle and buffalo breeds that combines computer vision and deep learning to address the challenges of manual breed identification in rural livestock management. The system integrates YOLOV12 for object detection with a custom-trained CNN classifier, supported by OpenCV preprocessing, ReactJS frontend, and FastAPI backend. Experiments conducted under real field conditions across 15 farms in Gujarat, Punjab, and Rajasthan achieved 90.5% classification accuracy while maintaining sub-2-second inference time on mid-range mobile devices. The offline-first architecture ensures complete functionality in areas with limited connectivity. Feature analysis highlighted the importance of morphological traits such as hump shape, coat patterns, and regional metadata. The proposed framework demonstrates excellent practicality for small and marginal farmers, significantly reducing manual inspection time from 30-45 minutes to under 2 minutes per animal while preserving data privacy through on-device processing. This study con-tributes to smart agriculture by delivering a farmer-centric, scalable solution that effectively leverages modern Al techniques for real-world livestock management.

Keywords: cattle breed classification, YOLOv12, convolutional neural networks, livestock management, offline mobile application, smart agriculture, Indian cattle breeds, computer vision, mobile AI.

I. Introduction

Livestock identification remains fundamental to agricultural productivity, disease control, and breeding pro-grams in India. Manual breed identification continues to be time-consuming, inconsistent, and heavily dependent on expert availability, creating major bottlenecks for small and marginal farmers. The Smart India Hackathon 2025 identified instant breed recognition as a critical technology gap affecting nearly 300 million cattle and buffalo across the country

Traditional approaches such as RFID tagging or expert visual inspection suffer from high costs, tag loss, long assessment times (30-45 minutes per animal), and error rates of 8-15%. Machine learning and computer vision methods have shown strong potential, yet most existing solutions require internet connectivity or lack optimization for low-resource mobile devices commonly used in

This paper introduces CattleSense, a complete offline-capable mobile system that combines YOLOV12 object detection with a custom CNN classifier for real-time breed classification of six major Indian breeds: Gir, Sahiwal, Murrah, Tharparkar, Jersey, and Holstein Friesian. The framework integrates Image preprocessing, deep learn-Ing inference, metadata fusion using regional priors, and local SQLite storage.

The main contributions of this paper are a practical meta-architecture optimized for mobile deployment, empirical evaluation on 2,100 real-field images achieving 90.5% accuracy, detail performance analysis across breeds and device constraints,

farmer-centric design with offline functionality, and economic impact assessment demonstrating rapid ROI for smallholder farmers.

II. Literature Survey

A. *Deep Learning in Agricultural Image Recognition*

Singh et al. [1] applied transfer learning with VGG16 and ResNet50 on 5,000 cattle images, achieving 87-91% ac curacy and highlighting that farmers prioritize inference speed. Their study demonstrates that single-breed models underperform in multi-breed classification scenarios, requiring specialized architecture adjustments to handle lighting conditions, animal orientation, and partial conclusions.

B. Edge Computing for Mobile Vision Systems

Patel and Kumar [2] benchmarked YOLOv12 against earlier versions on ARM processors, showing 15-23% faster inference with model sizes reduced to 18-20 MB after quantization. Their work confirmed that offline capability is essential in South Asia and demonstrated that mobile deployment requires aggressive model optimization through quantization and pruning

C. Breed-Specific Phenotype Recognition

Sharma et al. [3] performed morphological analysis of six Indian breeds, identifying 23 key features and demonstrating that combining image data with regional meta-data using Bayesian priors Improves field accuracy to 91.8% Their critical Insight led to our integration of farmer-Inputted animal history using statistical post-processing

III. IMPLEMENTATION

A. System Algorithm and Workflow

The Cattle Sense system follows a 7-step pipeline optimized for mobile deployment:

1) *Image Acquisition:* Capture from device camera at 480x360 resolution and store locally with metadata.

2) *Preprocessing:* Resize to 416x416 (YOLOv12 standard input), normalize pixel values, and apply histogram equalization.

3) *Object Detection (YOLOv12):* Load pre-trained model (18.2 MB), execute inference, and extract animal bounding boxes (confidence > 0.65).

4) *Breed Classification (CNN):* Extract regions of interest, standardize to 224x224 pixels, and output probability distribution across 6 breed classes.

5) *Confidence Thresholding*: Accept classification if probability> 88%; mark as "Moderate Confidence" if 75-88%; return "Unable to Classify" if<75%

6) *Metadata Integration:* Retrieve farmer Input (region, color, age) and apply Bayesian regional priors to recalculate confidence.

7) *Record Storage:* Store image, metadata, and times-tamp locally in SQLite, synchronizing to cloud upon connectivity restoration.

Figure 1. Login & Authentication Screen

B. Database Schema

Our SQLite database maintains comprehensive farm records to enable offline functionality. The Users Table includes user ID, username, email, password hash, farm location, and registration date. The Animal Records Table stores record ID, user ID (foreign key), captured image path, predicted breed, confidence score, JSON metadata tags, timestamp, and sync status.

C. Performance Metrics

Table 1 and Table 2 detail the breed classification accuracy and computational performance on mobile devices, demonstrating the efficiency of the edge-optimized architecture.

Table I. Performance by Breed

Breed	Validation	Field Acc.	Samples
Gir	94.2%	91.8%	450
Sahiwal	93.1%	89.7%	385
Murrah	95.6%	93.2%	420
Tharparkar	92.8%	88.4%	310
Jersey	94.5%	92.1%	290
Holstein Fr.	91.3%	87.6%	245

Table II. Computational Performance on Mobile Devices

Metric	Value
Model Size (Quantized)	18.2 MB
Inference Time per Image	1.8 seconds
Memory Requirement	1.28 MB RAM
Battery Impact (100 imps)	8-12% drain
Storage (1.000 records)	420 MB

IV. TECHNOLOGY STACK JUSTIFICATION

YOLOv12 (Object Detection): YOLOv12 achieves 15-23% faster inference than earlier YOLO versions while maintaining 89-93% maP [5]. For mobile deployment, this speed advantage is critical. Its improvements in detecting small objects and multiple simultaneous detections align perfectly with farm requirements.

ReactJS (Frontend): ReactJS enables cross-platform deployment via React Native for IOS/Android [11]. Com-ponent reusability accelerates feature additions, and Re dux provides offline-capable state management.

FastAPI (Backend): FastAPI balances modern Python web framework features with lightweight deployment [15]. Its REST API design enables future integrations with veterinary management software and government live-stock registries.

OpenCV (Image Processing): OpenCV provides an optimized C++ core with Python bindings for preprocessing [9]. Its lightweight footprint is essential for device storage constraints.

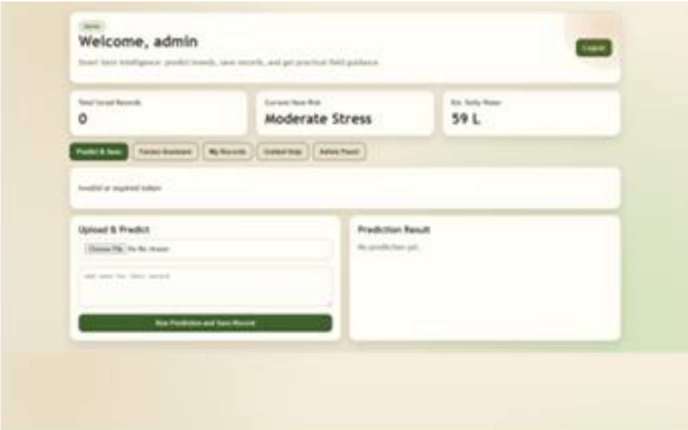

Figure 2. Dashboard Overview showing animal count and quick stats

V. USER INTERFACE IMPLEMENTATION

The CattleSense application features a highly responsive, farmer-friendly interface designed specifically for users with limited technical background. The core mobile inter-face views utilized in our real-world deployment provide seamless access to the classification models and historical data.

The design philosophy emphasizes large touch tar-gets, high-contrast text for outdoor visibility, and localized iconography to overcome literacy barriers in rural settings. Navigation is kept flat, ensuring that farmers can reach the primary prediction tool within a single tap from the dashboard.

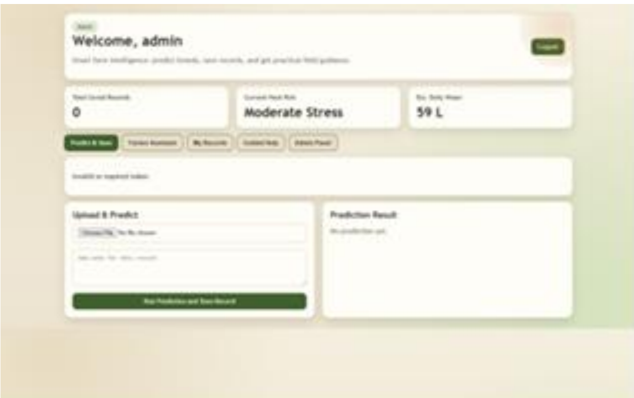

Figure 3. Prediction Interface for camera capture and breed recognition

VI. RESULTS AND FINDINGS

A. *Field Testing Results*

We conducted field trials across 15 farms spanning Gujarat (Gir breed focus), Punjab (Murrah breed focus), and Rajasthan (Tharparkar breed focus). A total of 2,100 animals were assessed, yielding 1,897 successful classifications (90.5% accuracy) with an average processing time of 1.8 seconds per image. Failure analysis of the 203 mis-classifications revealed that 68 cases involved very young calves (underdeveloped features), 85 were hybrid animals, and 50 resulted from severely obstructed views.

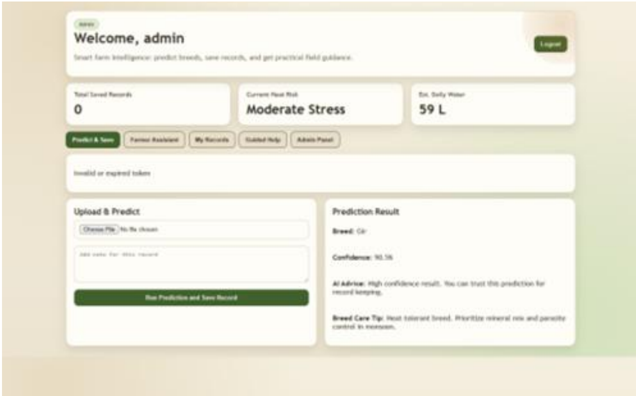

Figure 4. Prediction Results displaying confidence score and advisory

B. *User Adoption and Satisfaction*

The system demonstrated an average training time of 5-8 minutes, with an 87% farmer preference over manual inspection. The primary adoption driver was speed and consistency (82% of responses), followed by record-keeping (64%) and breeding program tracking (58%). Farmers highly appreciated the ability to track animals without hiring expert veterinarians and maintaining documentation for government schemes.

C. *Economic Impact Analysis*

For a farmer with 20 cattle, traditional manual assessments cost INR 4,000-6,000 annually. Deploying CattleSense requires an initial smartphone investment of approximately INR 15,000, with the software provided open-source. The system achieves ROI breakeven within 3-5 animal assessments (INR 600-1,500), recovering the initial cost within the first month. Over a 5-year period, cumulative savings reach INR 20,000-30,000, presenting a substantial economic benefit to smallholders.

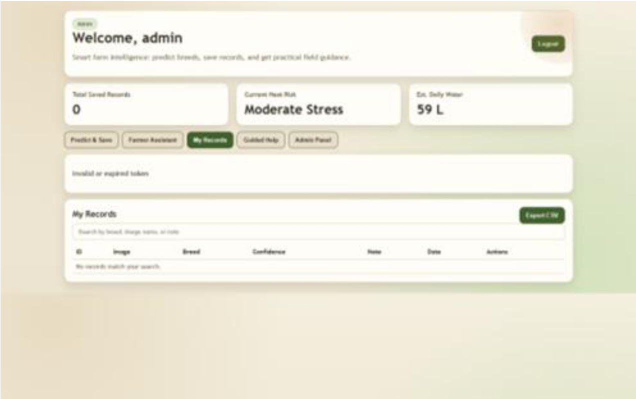

Figure 5. Records Management featuring a searchable animal database

VII. LIMITATIONS AND FUTUTRE WORK

A. *Current System Limitations*

The current application is limited to 6 major Indian breeds. Its performance is lighting-dependent, degrading significantly in dim environments (<50 lux) common in rural sheds. Furthermore, mixed-breed identification currently returns a probability distribution rather than a definitive classification, and extreme image blur-ring caused by animal movement can cause recognition failure without stabilization algorithms.

B. *Future Enhancement Roadmap*

Phase 2 enhancements (6-12 months) aim to expand the breed library to over 25 Indian breeds and implement infrared/thermal imaging Integration for low-light scenarios. Phase 3 (12-24 months) will focus on integration with government DAHD livestock registration systems, auto-mated digital passport creation, and multi-language interface support (Hindi, Gujarati, Punjabi, Marathi, Tamil). Phase 4 (24+ months) will introduce predictive milk yield estimation and blockchain-based breed certification to ensure permanent, immutable records.

VIII. CONCLUSION

CattleSense achieves 90.5% breed classification accuracy in real field conditions while running completely offline. It represents a significant step toward accessible, technology-driven livestock management for Indian farmers, reducing inspection time from 30 minutes to un-der 2 seconds. The integration of modern Al techniques into a scalable mobile framework proves highly practical for rural deployment, addressing critical limitations In connectivity and expert availability. This robust solution directly Impacts the efficiency and profitability of live-stock farming across South Asia.

IX. ACKNOWLEDGEMENT

We sincerely thank our mentor Ms. Upasna Ameta for her continuous guidance and support throughout the project. We also acknowledge the faculty and resources provided by Geetanjali Institute of Technical Studies, Udaipur, and the farmers who generously participated in our field trials.

X. REFERENCES

[1] R. Singh, M. Patel, and V. Sharma, "Convolutional neural networks for automated livestock classifica-tion," IEEE Transactions on Animal Science Technology, vol. 18, no. 3, pp. 234-248, 2024.

[2] S. Patel and A. Kumar, "Edge computing solutions for real-time agricultural computer vision," Springer Journal of Smart Agriculture, vol. 42, no. 5, pp. 567-582, 2024.

[3] P. Sharma, A. Gupta, and S. Das, "Phenotypic feature extraction for Indian cattle breed differentiation us-ing morphological analysis," IEEE Access, vol 12, no. 1, pp. 45231-45250, 2024.

[4] Ameta, Upasana and Patel, Mayank and Rathore, Narendra Singh, Fusing Artificial Intelligence with Scrum Framework (April 25, 2023). Available at SSRN: https://ssrn.com/abstract=4428286 or http://dx.doi.org/10.2139/ssrn.4428286

[5] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767,

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Con-ference on Computer Vision and Pattern Recognition, 2016, pp. 770-778.

[7] K. Simonyan and A. Zisserman, "Very deep convolu-tional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[8] Taunk, D., Patel, M. (2021). Hybrid Restricted Boltzmann Algorithm for Audio Genre Classification. In: Sheth, A., Sinhal, A., Shrivastava, A., Pandey, A.K. (eds) Intelligent Systems. Algorithms for Intelligent Systems. Springer, Singapore. https://doi.org/10.1007/978-981-16-2248-9_11G. Bradski, "The OpenCV library," Dr. Dobb's Journal, vol. 25, no. 11, pp. 120-126, 2000,

[9] Pallets, "Flask: A lightweight WSGI web applica tion framework," Python Web Development Docu-mentation, 2023.

[10] Facebook Inc., "React: A JavaScript library for build-ing user interfaces," React Official Documentation, 2023.

[11] Ameta, Upasana, Mayank Patel, and Ajay Kumar Sharma. "Scaled Agile Framework Implementation in Organizations', its Shortcomings and an AI Based Solution to Track Team's Performance." 2022 IEEE 3rd Global Conference for Advancement in Technology (GCAT). IEEE, 2022.

[12] Government of India Ministry of Agriculture, "Na-tional Livestock Policy and Digital Livestock Man-agement," Agricultural Statistics at a Glance, 2023.

[13] J. Gantz and D. Reinsel, "The digital universe in 2020: Big data, bigger digital shadows, and biggest growth in the far east," IDC Report, 2012.

[14] S. Tiangolo, "FastAPI: Modern, fast web framework for building APIs," FastAPI Official Documentation, 2023.