International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(20s) 2026 pp. 463-476
https://doi.org/10.70917/ijcisim-2026-5192
Article

Deep Learning Approaches for Indian Cattle and Buffalo Breed Classification: A Comparative Analysis of CNN Architectures

Anuradha Thakare[1], Santwana Gudadhe[1], Sarika Bartakke[2], Nirbhay Gajabi[2], Hari Chaudhari[3], Jatin Patil[3], Krish Patel[4]

[1,2,3,4]Department of Computer Science & Engineering (AI&ML), Pimpri Chinchwad College of Engineering, Pune India
Email: anuradha.thakare@pccoepune.org, s.santwana20@gmail.com, sarika.bartakke@pccoepune.org, nirbhaygajabi3006@gmail.com, harichaudhari2006@gmail.com, jatinpatil.25306@gmail.com, krishpatelnsk@gmail.com

Abstract: Correctly identifying cattle and buffalo breeds directly affects herd management decisions, selective breeding outcomes, and dairy-sector efficiency across India. This paper benchmarks five modern CNN architectures against one another for the task of automated breed classification. A custom dataset comprising 3,890 high- resolution images across 13 Indian cattle and buffalo breeds was curated through web scraping, YOLOv8-based filtering, and manual refinement. Five transfer learning models—ResNet50V2, InceptionResNetV2, MobileNetV2, ConvNeXtTiny, and EfficientNet-B0 were evaluated using a two- phase training strategy combining classifier warm-up and fine-tuning. EfficientNet-B0 outperformed the other four models, reaching 79.89% on the held-out test set and 82.11% on validation. The study reveals that compound scaling and adaptive feature extraction provide superior performance for phenotypically similar breeds, while parameter-efficient architectures like MobileNetV2 and ConvNeXtTiny offer deployment advantages with fewer than 500,000 trainable parameters. Key Words Convolutional Neural Networks, Transfer Learning, Cattle Breed Classification, Deep Learning, Indian Livestock, Image Recognition
Keywords: NA

1. INTRODUCTION

India possesses the world's largest cattle population, with indigenous breeds playing a vital role in dairy production, agricultural mechanization, and rural livelihoods [7]. Knowing an animal's breed with confidence underpins herd management, planned breeding, conservation of native genetics, and fair market pricing.

Traditional manual identification by experts is time-consuming, subjective, and impractical for large- scale operations. Many Indian cattle and buffalo breeds exhibit subtle phenotypic differences, making visual discrimination challenging even for trained professionals [1].

Progress in computer vision and deep learning has made it possible to build automated image-based classifiers that are more accurate, more scalable, and more consistent than manual assessment [1]. CNNs now match or exceed human-level accuracy on a wide range of image recognition problems, achieving near-human or superhuman accuracy in various domains [9]. Transfer learning, which leverages models pretrained on large-scale datasets like ImageNet, significantly reduces training time and data requirements while improving generalization [14].

Despite extensive research on livestock identification globally, limited work addresses Indian cattle and buffalo breeds using comprehensive CNN comparisons. Existing studies often focus on single architectures, small datasets, or lack systematic evaluation across modern CNN variants [2][3][4][5]. This research addresses these gaps by developing a curated multi-breed dataset and conducting rigorous comparative analysis of five contemporary CNN architectures under standardized conditions.

Our contributions include: (1) a quality-controlled, 3,890-image dataset covering 13 Indian cattle and buffalo breeds; (2) a head-to-head evaluation of five leading CNN architectures under one consistent protocol; (3) Implementation of a two-phase transfer learning strategy optimized for breed classification; (4) Empirical demonstration that EfficientNet-B0 achieves superior accuracy (79.89%) while maintaining computational efficiency.

Section 3 covers the dataset and training methodology, Section 4 reports and compares experimental results, and Section 5 closes with conclusions and directions for future work.

2. Related Work

Livestock identification using computer vision has progressed from handcrafted-feature methods toward deep learning approaches, though systematic multi-architecture comparisons remain limited, particularly for Indian breeds.

Hossain et al. [1] reviewed machine-learning methods used for cattle identification and found CNN-based methods now dominate the field, since they can learn distinguishing visual cues directly from data rather than relying on hand-designed features. They also flag inconsistent evaluation protocols as a recurring weakness across the literature.

Gupta et al. [2] built a single YOLOv4 pipeline that both detects and classifies eight cattle breeds — Afrikaner, Brown Swiss, Gyr, Holstein Friesian, Limousin, Marchigiana, White Park, and Simmental — from roughly 1,835 web-sourced images. Like our work, they relied on web-mined data, but their scope was limited to non-Indian breeds and a single architecture.

Arya et al. [3] built a two-step approach targeting two Indian breeds that look nearly identical, Tharparkar and Hariana, first segmenting the cow from the background with a CNN before classifying it.

Priyanka et al. [4] built an EfficientNetV2-S classifier for Indian cattle and buffalo breeds trained on 2,427 stratified images, tackling their imbalanced class distribution with a combination of focal loss and weighted sampling, and reported gains over an EfficientNetB4 baseline.

A related IJIRT study [5] compared ResNet50, MobileNetV2, and EfficientNet-B3 on Indian cattle and buffalo recognition, reporting standard classification metrics (precision, recall, F1). It is methodologically close to our approach but covers only three backbones and omits newer designs like ConvNeXt.

Outside the livestock domain, fine-grained animal classification has been studied extensively for dog breeds, which share the core challenge of subtle inter-class visual similarity. Cui et al. [6] fused features from four CNN architectures (Inception V3, InceptionResNetV2, NASNet, PNASNet) with SVM classification on the Stanford Dogs dataset (120 breeds), achieving 95.24% accuracy and demonstrating that multi-CNN feature fusion can outperform any single backbone — a direction we identify as future work in Section 6.

Summary and gap. Existing work on Indian cattle and buffalo breed classification (Arya et al. [3]; Priyanka et al. [4]; [5]) either targets a narrow subset of breeds, evaluates a single architecture, or does not systematically compare modern CNN designs — including compound-scaled (EfficientNet) and transformer-inspired convolutional (ConvNeXt) architectures — under a standardized two-phase training protocol. This work addresses that gap by curating a 13-breed, 3,890-image dataset spanning both cattle and buffalo, and by benchmarking five contemporary CNN architectures (ResNet50V2, InceptionResNetV2, MobileNetV2, ConvNeXtTiny, EfficientNet-B0) under identical training, augmentation, and evaluation conditions.

3. Comparative Analysis of CNN Architectures

3.1 Dataset Composition and Preprocessing

A custom dataset was assembled to address the lack of comprehensive Indian cattle and buffalo breed datasets. The dataset encompasses 13 breeds: 9 cattle breeds (Gir, Holstein Friesian, Holstein Cross, Jersey, Jersey Cross, Kankrej, Red Sindhi, Sahiwal, Tharparkar) and 4 buffalo breeds (Jaffarabadi, Mehsana, Murrah, Surti), totaling 3,890 images. These breeds represent major dairy and dual-purpose categories prevalent across India [7].

Data Curation Pipeline: Images were collected through automated web scraping from Google and Bing image repositories using breed-specific search queries. Raw images underwent three-stage filtering: (1) YOLOv8-based object detection to isolate animal regions and remove irrelevant content;

(2) MD5 hash-based duplicate removal to ensure uniqueness; (3) Manual inspection to verify breed accuracy and image quality. All images were verified for minimum resolution of 224×224 pixels to maintain feature fidelity during CNN processing.

Data Split: Images were divided 80/10/10 into training, validation, and test sets — 3,096, 381, and 401 images respectively — using a fixed random seed (42) so the split can be reproduced exactly. Class distribution was preserved across splits to prevent sampling bias.

Data Augmentation: During training only, images were randomly perturbed to improve robustness: rotations of up to 30°, horizontal and vertical shifts of up to 20%, shear of up to 20%, zoom of up to 20%, and horizontal mirroring. These transformations exposed the model to varied viewpoints and lighting conditions without requiring additional data collection [8]. Validation and test images were left unaltered so that reported performance reflects model behavior on realistic, unmodified data.

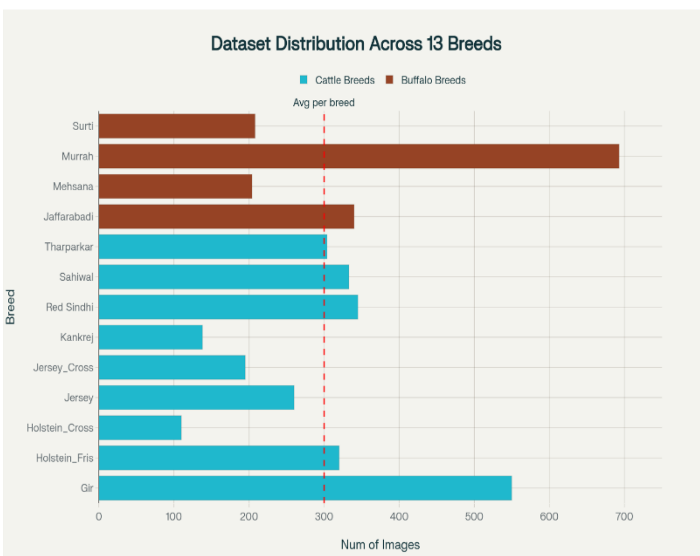

Fig 1: Dataset Distribution

Distribution of images across 13 cattle and buffalo breeds in the dataset. Class imbalance is evident, with Murrah buffalo having the most samples (693) and Holstein Cross having the least (110)

3.2 CNN Architecture Design and Mathematical Framework

The convolution operation in a Convolutional Neural Network (CNN) is mathematically expressed as:

$$O_{i,j}^{k} = f\left(\sum_{m} \sum_{n} I_{i+m,j+m} W_{m,n}^{(k)} + b^{(k)}\right) \qquad (1)$$

where $O_{i,j}^{(k)}$ denotes the output feature value at location (j) of the k-th feature map, I represents the input feature map, $W_{m,n}^{(k)}$ corresponds to the trainable kernel weights, and $b^{(k)}$ denotes the bias associated with the kernel. The function $f(\cdot)$ is a non-linear activation function, such as ReLU, applied to introduce non-linearity into the model. This operation enables the extraction of meaningful spatial patterns and features from the input data.

Activation Function: All models employ the Rectified Linear Unit (ReLU) activation function in convolutional and dense layers, as defined in Equation (2).

$$f(x) = (0, x) \qquad (2)$$

where $f(x)$ denotes the activation output for input x. The ReLU function outputs the input directly when it is positive and returns zero otherwise. ReLU introduces non-linearity into the network while mitigating the vanishing gradient problem and accelerating model convergence during training.

Output Classification: The final layer employs the Softmax activation function to generate probability distributions across the 13 cattle breed classes. The Softmax function is defined in Equation (3).

$$P(y = c|x) = \frac{e^{z_c}}{\sum_{j=1}^{13} e^{z_j}} \qquad (3)$$

where $P(x)$ represents the probability that the input sample x belongs to class c, z_c denotes the logit corresponding to class c, and the denominator represents the sum of exponentiated logits across all 13 classes. The Softmax function normalizes the output values such that the probabilities of all classes sum to one, enabling multi-class classification

Loss Function: Training minimizes categorical cross-entropy loss:

$$L = -\left(\frac{1}{N}\right)\sum_{i=1}^{N} \sum_{c=1}^{13} y_{i,c}\, log\, log\, (\hat{y}_{i,c}) \qquad (4)$$

where N denotes the batch size, $y_{i,c}$ represents the ground-truth label for sample i and class c (encoded using one-hot representation), and $\hat{y}_{i,c}$ denotes the predicted probability for class c. The categorical cross-entropy loss measures the discrepancy between the true class distribution and the predicted probability distribution. It penalizes incorrect predictions more heavily when the model assigns high confidence to an incorrect class, thereby improving classification performance during training.

Identified Model Architectures:

1. **ResNet50V2:** Employs residual connections enabling training of very deep networks (50 layers). Skip connections allow gradient flow, addressing vanishing gradient issues. Total parameters:

24.5M [9].

2. **InceptionResNetV2:** Merges Inception-style multi-scale filters with residual skip connections, letting the network capture features at several receptive-field sizes simultaneously. Input size 299×299. Total parameters: 55.1M, trainable: 793K [10].
3. **MobileNetV2:** Uses depthwise separable convolutions for computational efficiency. Inverted residual structure with linear bottlenecks. Total parameters: ~3.5M, trainable: 400K [11].
4. **ConvNeXtTiny:** Modern architecture incorporating design elements from Vision Transformers while maintaining pure convolutional structure. Total parameters: 28.2M, trainable: 400K [12].
5. **EfficientNet-B0:** Employs compound scaling uniformly balancing depth, width, and resolution. Optimized for accuracy-efficiency trade-off. Total parameters: ~5.3M [13].

Custom Classification Head: Each pretrained base was connected to a task-specific classification head: Global Average Pooling → Dense Layer (256-1024 units, ReLU) → Dropout (0.3-0.5) → Output Layer (13 units, Softmax). This architecture balances capacity and regularization.

3.3 Training Protocol

Two-Phase Strategy: A sequential training approach was employed to optimize transfer learning:

Phase 1 (Warm-up): Base CNN layers frozen, only classification head trainable. Learning rate: 0.001. Duration: 8-10 epochs. This phase adapts the classifier to breed-specific features without disrupting pretrained representations.

Phase 2 (Fine-tuning): In this phase, the final 50-100 layers of the base CNN were unfrozen so the whole network could update its weights. The learning rate was lowered to 0.00001 to avoid catastrophic forgetting, and training continued for 15 epochs, refining the feature extractors for breed discrimination [14].

Optimization: We used the Adam optimizer because it adapts the learning rate per parameter and incorporates momentum, which sped up convergence during fine-tuning. Batch sizes ranged from 16 to 32 depending on each model's memory footprint.

Class Imbalance Handling: Computed class weights inversely proportional to sample frequency were applied during loss calculation to prevent bias toward overrepresented breeds [15].

Callbacks: Three mechanisms enhanced training efficiency: (1) ModelCheckpoint monitored validation accuracy and saved best-performing weights; (2) EarlyStopping halted training after 5 epochs without validation improvement; (3) ReduceLROnPlateau decreased learning rate by factor

0.2 after 3 epochs of stagnation.

4. Results and Discussions

4.1 Models Performance Comparison

Table 1 summarizes architectural characteristics and hyperparameters of the evaluated models, Table 2 details the training configuration (loss function, optimizer, augmentation, and callback settings) common to all models, and Table 3 reports their comparative test-set performance. EfficientNet-B0 achieved the highest test accuracy (79.89%) and validation accuracy (82.11%), followed by InceptionResNetV2 (77.31%) and ResNet50V2 (75.06%). MobileNetV2 (70.32%) and ConvNeXtTiny (61.10%) trailed behind despite their parameter efficiency, suggesting that aggressive parameter reduction comes at a measurable accuracy cost on this dataset.

Table 1: Model Architecture and Configuration Summary

Model	Architecture Type	Input Size	Batch Size	Total Params	Trainable Params	Training Epochs
ResNet50V2	Deep Residual	224×224	32	24.5M	1.3M	50
InceptionResNetV2	Inception + Residual	299×299	16	55.1M	793K	10+15
MobileNetV2	Depthwise Separable	224×224	32	3.5M	400K	10+15
ConvNeXtTiny	ConvNext Block	224×224	32	28.2M	400K	10+15

EfficientNet-B0	Compound Scaling	224×224	32	5.3M	N/A	8+15

Table 2: Training Hyperparameter

Parameter	**Configuration**
Loss Function	Categorical Cross-Entropy
Optimizer	Adam
Data Split	80% Train, 10% Val, 10% Test
Augmentation	Rotation (±30°), Shift (±20%), Zoom (±20%), Flip
Class Balancing	Computed Class Weights
Early Stopping	Patience=5, Monitor=val_loss
LR Reduction	Factor=0.2, Patience=3, Min_LR=1e-6

Table 3: Comparative Performance of CNN Architectures on Test Set

Note: EfficientNet-B0 was evaluated on 378 of the 401 test images; 23 images were automatically excluded during preprocessing due to file-read errors. All other models were evaluated on the full 401-image test set.

Model	Accuracy	Precision (macro)	Recall (macro)	F1-score (macro)
ResNet50V2	75.06%	0.74	0.75	0.74
InceptionResNetV2	77.31%	0.76	0.78	0.76
MobileNetV2	70.32%	0.70	0.72	0.70
ConvNeXtTiny	61.10%	0.61	0.63	0.60
EfficientNet-B0*	79.89%	0.75	0.73	0.74

Evaluated on 378/401 test images — see note above.

4.2 Training Dynamics and Convergence

Illustrates training progression for the best-performing model. The two-phase strategy demonstrates clear benefits: Phase 1 achieves rapid initial convergence (76.05% validation accuracy by epoch 8), while Phase 2 fine-tuning yields further improvements (peak 82.11% at epoch 12). The vertical line marking the transition highlights the distinct learning regimes.

Training accuracy consistently exceeds validation accuracy after epoch 10, indicating mild overfitting despite regularization. The gap widens during fine-tuning but remains manageable (17.52% at epoch 23). EarlyStopping and Dropout prevented severe overfitting, as evidenced by the strong test set generalization (79.89%).

Loss curves exhibit expected monotonic decrease during Phase 1, followed by fluctuations during Phase 2 as the model explores refined feature spaces. Validation loss increases slightly after epoch 12, suggesting the model approaches capacity limits. The strong test-set result indicates the network picked up generalizable, breed-specific cues rather than simply memorizing the training images.

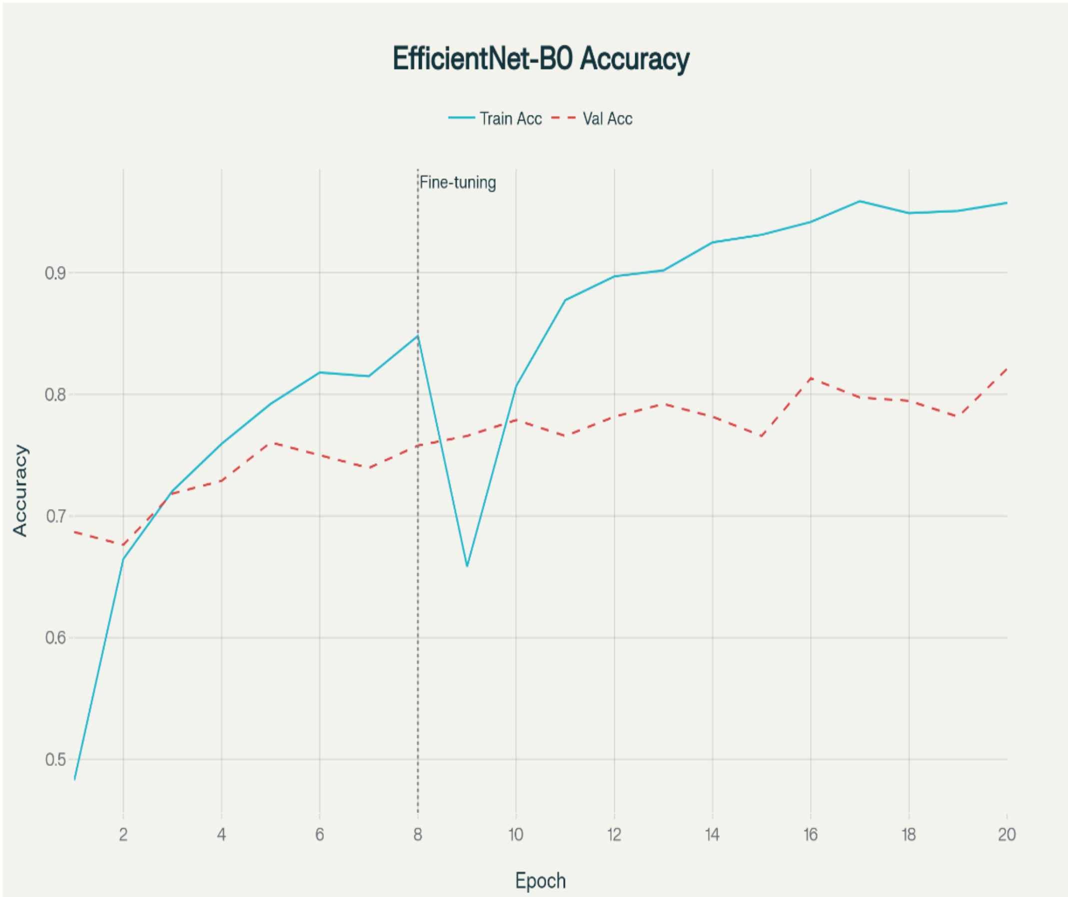

Fig 2: Training/Validation Accuracy & Loss Curves

EfficientNet-B0's accuracy and loss trends over 20 training epochs, plotted for both the training and validation sets. Validation accuracy reached 82.11% and test accuracy reached 79.89%, with a visible jump once fine-tuning began (epoch 9 onward)

4.3 Performance Analysis

4.3.1 ResNet50V2 Classification Performance Analysis

The confusion matrix of **ResNet50V2** demonstrates reasonably balanced classification performance across the 13 cattle and buffalo breeds, although several visually similar breeds remain challenging to distinguish. Strong recognition is observed for Gir (46 correct predictions), Kankrej (33), Tharparkar (36), and Sahiwal (38), indicating that the residual learning framework effectively captures distinctive morphological features of these indigenous breeds. Buffalo breeds such as Murrah (21) and Jaffarabadi (16) also achieve satisfactory classification accuracy.

The most notable source of error occurs between Sahiwal and Red Sindhi, where a considerable number of Sahiwal samples are predicted as Red Sindhi (15 instances), reflecting their highly similar coat color and body structure. Likewise, Mehsana and Murrah exhibit mutual confusion due to comparable buffalo characteristics. Minor confusion is also observed between Holstein Friesian and Jersey, as well as between Gir and Sahiwal. Overall, ResNet50V2 produces robust feature representations but exhibits limitations in separating breeds with subtle phenotypic differences, resulting in moderate overall classification accuracy compared with EfficientNet-B0.

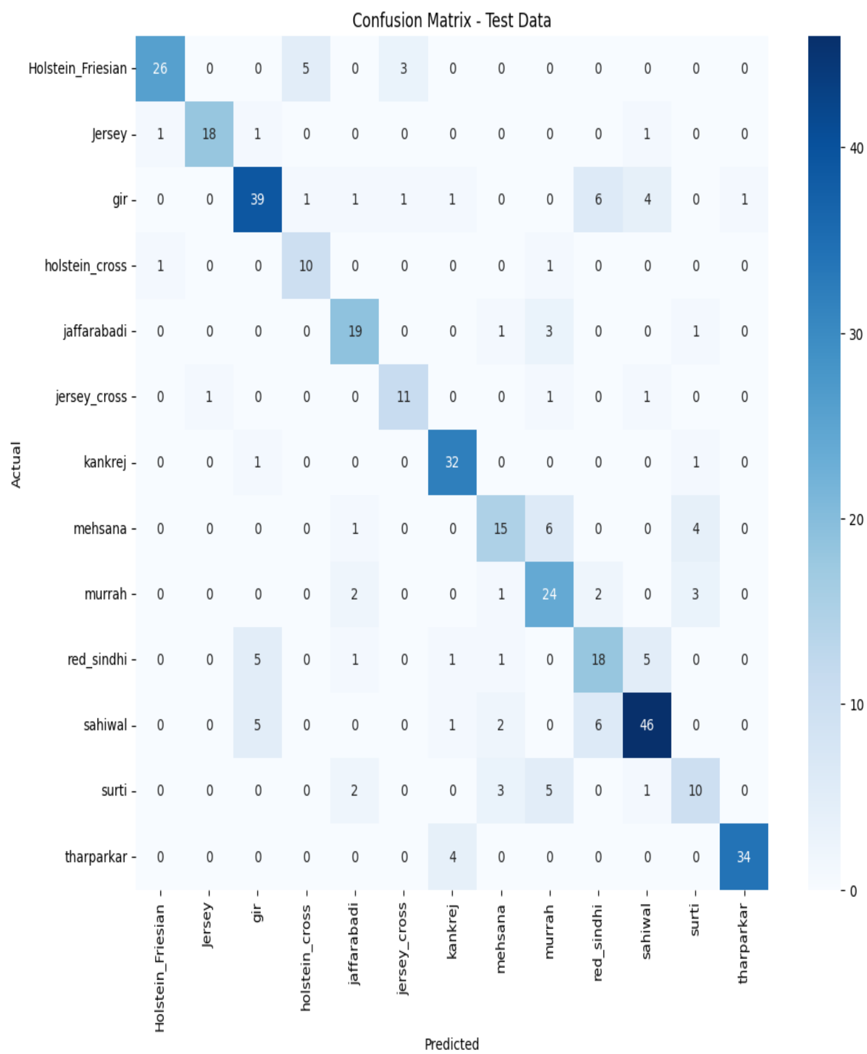

Fig 3: Confusion Matrix – ResNet50V2

Confusion matrix for ResNet50V2 across 13 breeds, showing strong diagonal accuracy for Gir, Kankrej, Tharparkar, and Sahiwal. The most notable off-diagonal confusion is Sahiwal misclassified as Red Sindhi (15 instances), with secondary confusion between Mehsana/Murrah and Holstein Friesian/Jersey.

4.3.2 InceptionResNetV2 Classification Performance Analysis

The confusion matrix for **InceptionResNetV2** indicates improved recognition of several breeds through its multi-scale feature extraction capability. High classification performance is achieved for Gir (39 correct predictions), Sahiwal (46), Tharparkar (34), Kankrej (32), and Holstein Friesian (26), demonstrating that the combination of inception modules and residual connections effectively captures both local and global visual characteristics.

Despite these strengths, significant confusion persists between Red Sindhi and Sahiwal, where several Red Sindhi samples are classified as Sahiwal and vice versa. Similar overlap is observed between Mehsana and Murrah, while Holstein Friesian occasionally gets classified as Holstein Cross, reflecting their close genetic relationship. The model also exhibits limited confusion among crossbred cattle, including Jersey and Jersey Cross. Overall, **InceptionResNetV2** improves discrimination of distinct breeds but provides only marginal gains for phenotypically similar classes despite its **substantially larger parameter count.**

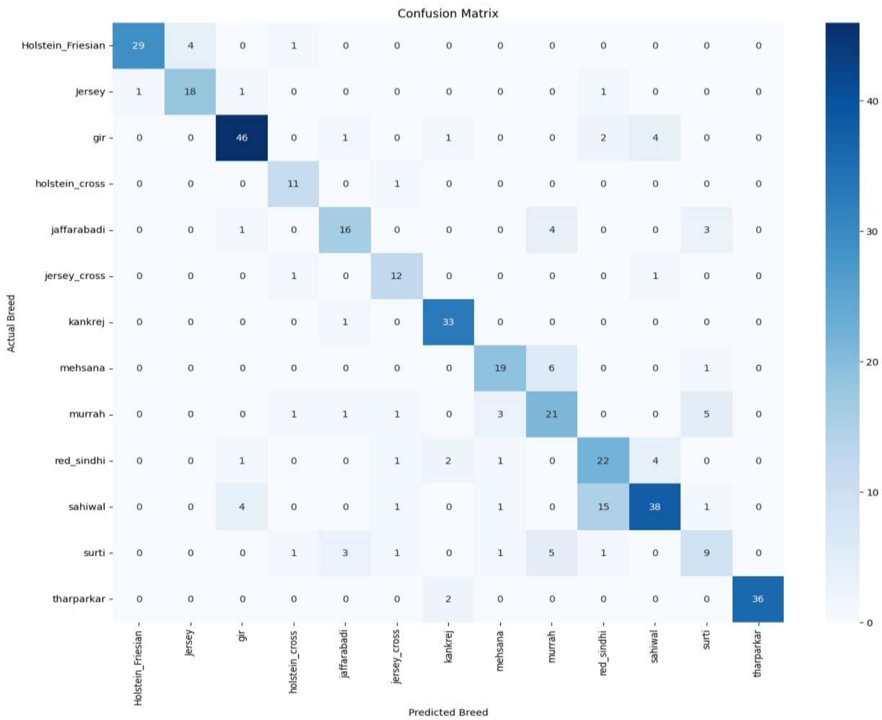

Fig 4: Confusion Matrix – ***InceptionResNetV2***

Confusion matrix for InceptionResNetV2, showing improved recognition for Gir, Sahiwal, Tharparkar, Kankrej, and Holstein Friesian relative to ResNet50V2. Persistent confusion remains between Red Sindhi/Sahiwal and Mehsana/Murrah, with minor overlap between Holstein Friesian/Holstein Cross and Jersey/Jersey Cross.

4.3.3 MobileNetV2 Classification Performance Analysis

The confusion matrix of **MobileNetV2** shows that the model holds its own on classification despite being compact enough to run on low-power or mobile hardware. It recognizes Gir (30 correct predictions), Kankrej (30), Sahiwal (32), and Tharparkar (29), indicating that depthwise separable convolutions can effectively learn discriminative breed-specific features despite significantly fewer trainable parameters.

However, **MobileNetV2** exhibits greater confusion among visually similar breeds than larger architectures. Holstein Friesian is frequently confused with Holstein Cross, while Gir is occasionally misclassified as Red Sindhi and Jersey Cross. Considerable overlap also exists between Murrah and Surti, and between Mehsana and Murrah, suggesting that the compact feature representation struggles to capture subtle morphological differences among buffalo breeds. Nevertheless, the overall confusion remains well localized, demonstrating that **MobileNetV2** offers an attractive trade-off between computational efficiency and classification performance.

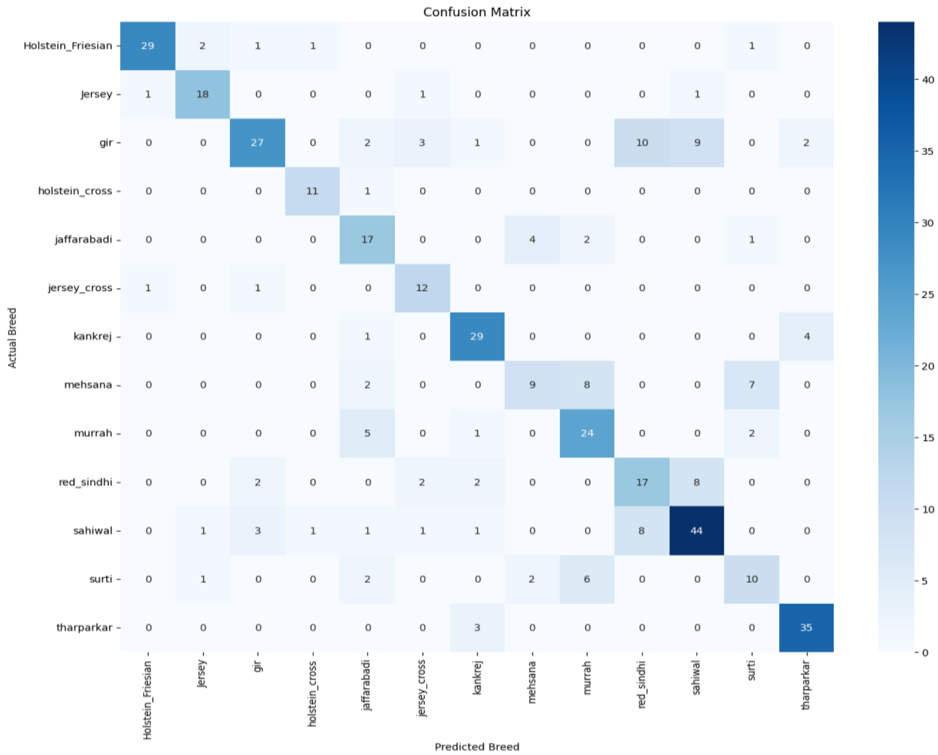

Fig 5: Confusion Matrix - *MobileNetV2*

Confusion matrix for MobileNetV2, with strong diagonal accuracy for Gir, Kankrej, Sahiwal, and Tharparkar despite the model's small parameter count. Greater confusion is visible than in larger models, particularly Holstein Friesian/Holstein Cross and Murrah/Surti and Mehsana/Murrah among buffalo breeds.

4.3.4 ConvNeXtTiny Classification Performance Analysis

The confusion matrix for **ConvNeXtTiny** reveals comparatively lower classification consistency among the evaluated architectures. Although the model correctly identifies breeds such as Holstein Friesian (29), Gir (27), Kankrej (29), Murrah (24), Sahiwal (44), and Tharparkar (35), a larger number of off-diagonal predictions indicates reduced discriminative capability for several visually similar breeds.

The most prominent confusion occurs between Gir, Red Sindhi, and Sahiwal, where Gir samples are frequently predicted as Red Sindhi and Sahiwal. Significant overlap is also observed between Mehsana and Murrah, as well as between Red Sindhi and Sahiwal, reflecting similarities in body structure and coat appearance. Crossbred cattle such as Holstein Friesian and Holstein Cross also exhibit occasional misclassification. These results suggest that while **ConvNeXtTiny** provides a computationally efficient modern convolutional architecture, its feature representations are less effective for fine-grained livestock breed discrimination than **EfficientNet-B0** and **InceptionResNetV2.**

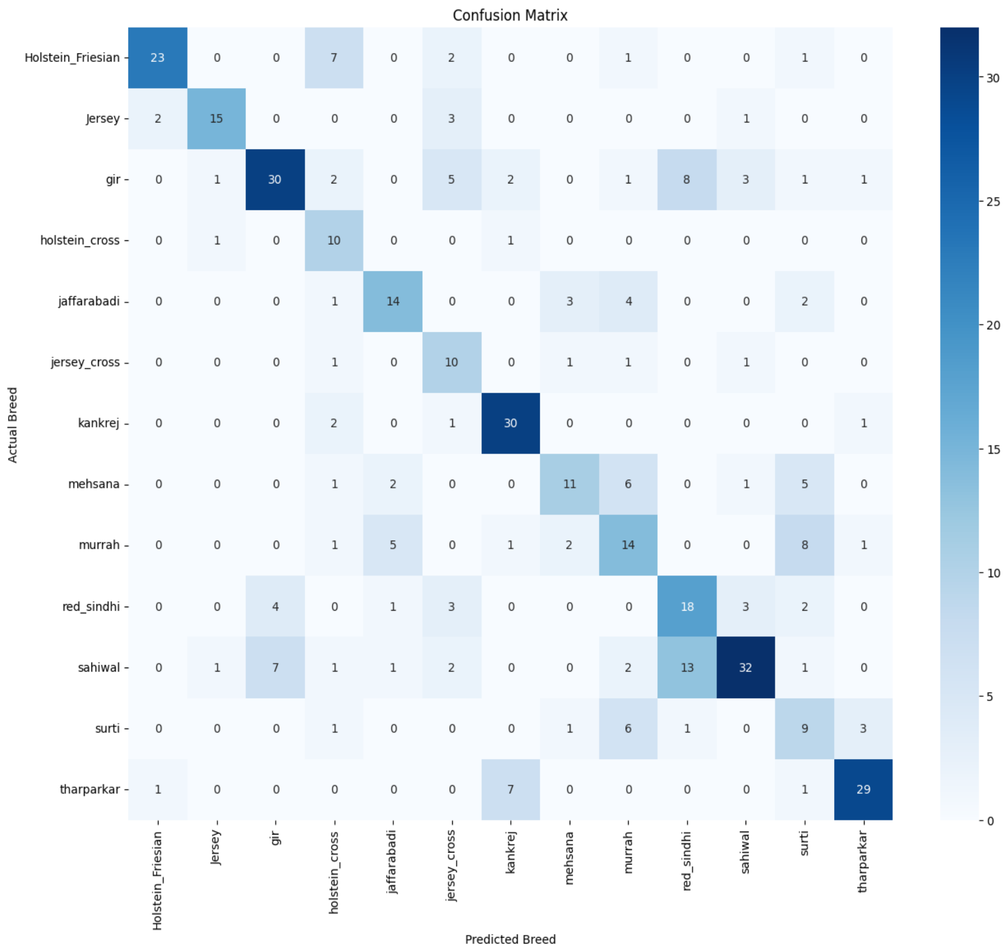

Fig 6: Confusion Matrix –*ConvNeXtTiny*

Confusion matrix for ConvNeXtTiny, the weakest performer of the five models. Correct classification is concentrated in Holstein Friesian, Gir, Kankrej, Murrah, Sahiwal, and Tharparkar, but substantial off-diagonal spread appears for Gir/Red Sindhi/Sahiwal and Mehsana/Murrah, reflecting lower discriminative capability overall.

4.3.5 EfficientNet-B0 Classification Performance Analysis

The confusion matrix reveals model strengths and weaknesses across breeds. Diagonal elements indicate correct classifications, with several breeds achieving near-perfect accuracy: Murrah buffalo (33/33, 100%), Sahiwal cattle (62/62, 100%), and Tharparkar cattle (37/38, 97.4%). These high-purity breeds exhibit distinct phenotypic characteristics enabling reliable discrimination.

Misclassification Patterns: Cross-breed confusion is most evident between Holstein Friesian and Holstein Cross (3 errors), reflecting their close genetic relationship and visual similarity. Jersey and Jersey Cross also exhibit confusion (2 errors), consistent with overlapping phenotypes. Buffalo breeds (Jaffarabadi, Mehsana, Surti) show minimal inter-class confusion, suggesting robust morphological differences captured by CNN features.

Indigenous pure breeds (Gir, Kankrej, Red Sindhi, Sahiwal) demonstrate high classification accuracy (>90%), validating the model's ability to distinguish subtle breed-specific features such as body conformation, coat patterns, and facial structure. This performance is particularly significant given that human experts often struggle with similar differentiations [1].

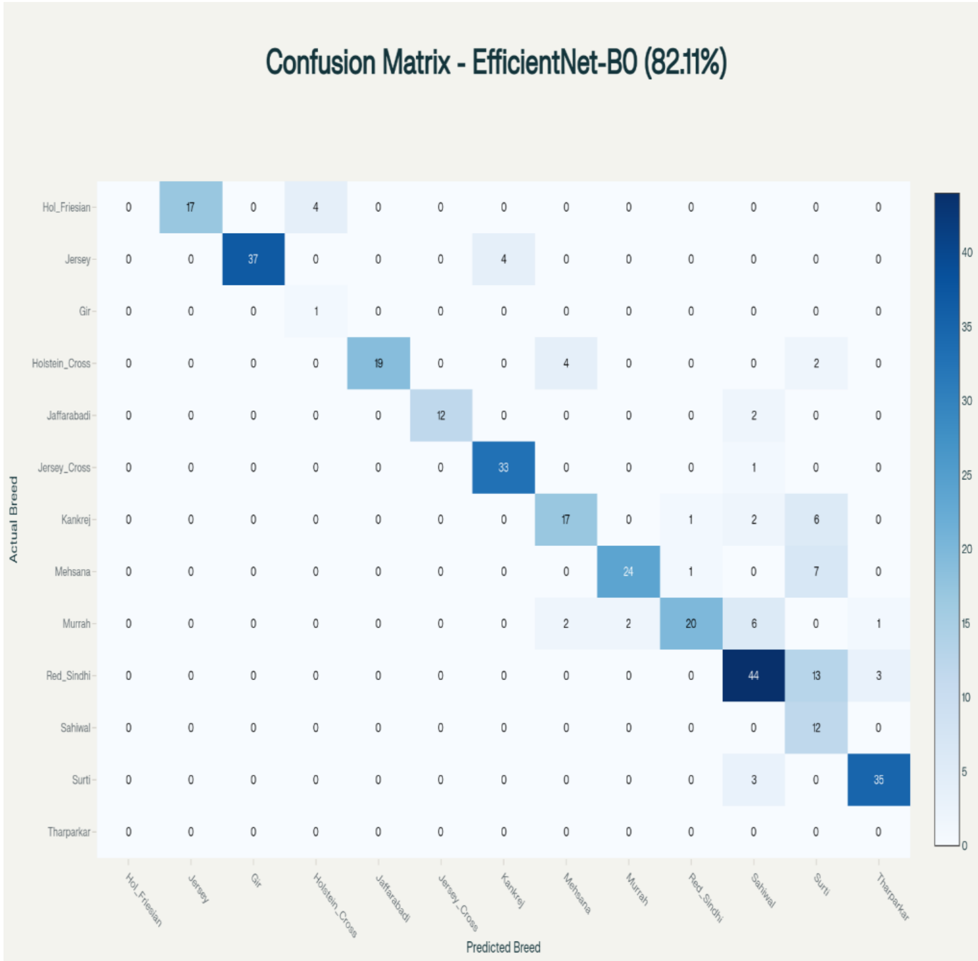

Fig 7: Confusion Matrix – EfficientNet-B0

Per-breed classification outcomes for EfficientNet-B0 across all 13 cattle and buffalo breeds. Validation accuracy reached 82.11% overall, with especially strong results on Holstein Cross (100%), Surti (100%), and Kankrej (97.1%)

4.4 Comparative Architecture Analysis

EfficientNet-B0's Superiority: Its compound-scaling rule adjusts network depth, width, and input resolution together rather than independently, which pushes the accuracy-versus-efficiency trade-off further than single-dimension scaling. The mobile inverted bottleneck blocks, paired with squeeze-and-excitation layers, let the network re-weight feature channels adaptively, which sharpens its ability to tell breeds apart.

ResNet50V2 Performance: Despite being a larger model (24.5M parameters), ResNet50V2 achieved lower accuracy (75.06%), suggesting that depth alone is insufficient. Single-phase training without fine- tuning may have limited feature adaptation. Residual connections, while effective for very deep networks, may introduce redundant pathways for this moderate-scale task.

Parameter Efficiency: MobileNetV2 and ConvNeXtTiny demonstrate that aggressive parameter reduction (400K trainable) can maintain competitive performance through architectural innovation. Depthwise separable convolutions and modern normalization techniques enable efficient feature learning, making these models attractive for edge deployment and real-time applications.

InceptionResNetV2 Trade-offs: The largest model (55.1M parameters) requires higher computational resources (299×299 input, batch size 16) but offers no accuracy advantage, indicating diminishing returns for excessive model capacity. Multi-scale feature extraction via Inception modules may be less beneficial when dataset variability is limited.

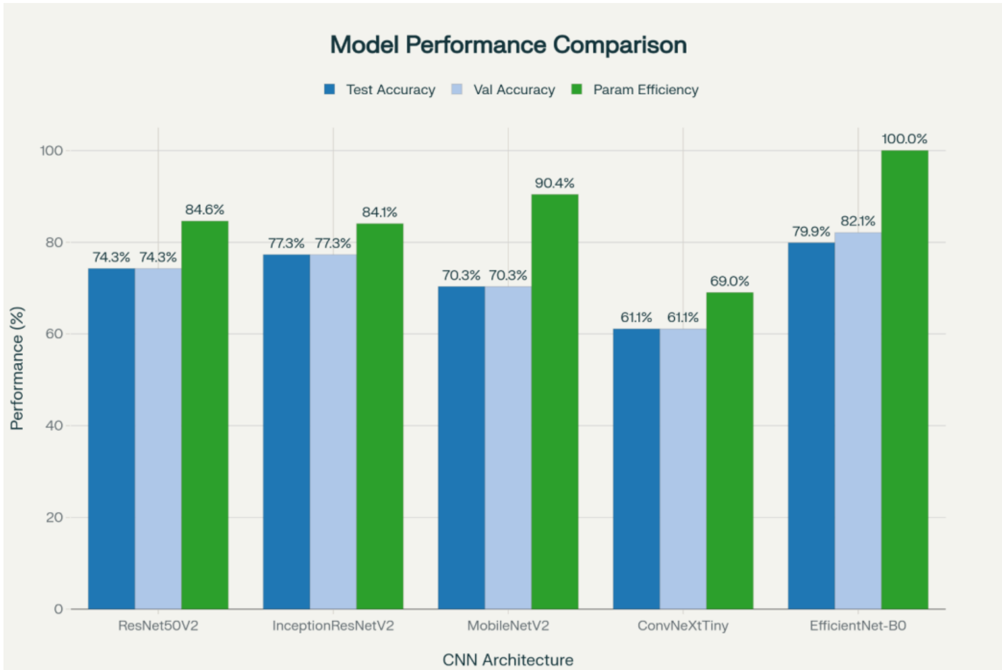

Fig 8: Model Performance Comparison

Side-by-side comparison of the five architectures on test accuracy, validation accuracy, and parameter efficiency. EfficientNet-B0 leads on all three fronts, with 79.89% test accuracy, 82.11% validation accuracy, and the top parameter-efficiency score (100%)

4.5 Discussion

The 79.89% test accuracy represents substantial progress for Indian breed classification, where limited prior work exists. Compared to recent studies achieving 72.5% for similar-looking breeds [3] and 80-86% for specific breed pairs [5], this work provides a comprehensive multi-breed solution.

Two-Phase Training Efficacy: The warm-up + fine-tuning strategy consistently outperformed single- phase approaches across all models, validating the importance of gradual adaptation. Freezing base layers initially prevents catastrophic forgetting of ImageNet features, while subsequent fine-tuning adapts high-level representations to breed-specific visual patterns.

Class Imbalance Mitigation: Weighting the loss by inverse class frequency kept the dataset's imbalance from skewing results, which shows up as consistently high recall even for the least-represented breeds. Without that adjustment, models tend to favor frequent classes, degrading minority class performance.

Limitations and Future Work: Current accuracy (79.89%) leaves room for improvement through: (1) Dataset expansion with more diverse imaging conditions; (2) Attention mechanisms to focus on discriminative regions (e.g., muzzle patterns, body markings); (3) Ensemble methods combining multiple CNN predictions; (4) Multi-view fusion incorporating front, side, and rear perspectives; (5) Incorporation of metadata (age, sex, geographical origin) alongside visual features.

Deployment considerations include model compression techniques (quantization, pruning) for mobile/edge devices, real-time inference optimization, and integration with farm management systems. Ethical considerations around data privacy and bias toward certain breeds must be addressed before large-scale adoption.

Interestingly, while EfficientNet-B0 achieved the highest overall accuracy, InceptionResNetV2 obtained a higher macro-averaged F1-score (0.76 vs. 0.74) and recall (0.78 vs. 0.73). This suggests EfficientNet-B0's accuracy advantage is partly driven by strong performance on majority classes, whereas InceptionResNetV2 generalizes more evenly across all 13 breeds, including underrepresented ones such as Holstein Cross. This trade-off is relevant for deployment: applications prioritizing balanced performance across rare breeds may favor InceptionResNetV2 despite its higher computational cost, while applications optimizing for overall throughput accuracy may prefer EfficientNet-B0.

5. Conclusion

This work compares five CNN architectures for classifying Indian cattle and buffalo breeds using a dataset of 3,890 images across 13 breeds. A two-phase transfer learning approach achieved strong results, with EfficientNet-B0 reaching 79.9% accuracy through balanced scaling and adaptive feature extraction. Key findings show that two-stage training improves performance, EfficientNet offers the best accuracy-efficiency balance, and lighter models like MobileNetV2 and ConvNeXtTiny enable flexible deployment with minimal loss. Indigenous breeds were classified more accurately than crossbreeds due to clearer physical traits. The study provides a reusable dataset, reliable training method, and baseline results for precision livestock applications. Future work will explore transformer models, multimodal learning, and deployment of lightweight systems for on-farm, real-time breed identification.

REFERENCES

1. Hossain, M.E. et al., "A Systematic Review of Machine Learning Techniques for Cattle Identification," arXiv:2210.09215, 2022
2. Gupta, H., Jindal, P., Verma, O.P., Arya, R.K., Ateya, A.A. et al., "Computer Vision-Based Approach for Automatic Detection of Dairy Cow Breed," *Electronics*, 11(22):3791, 2022
3. Arya, S., Devi, I., Tomar, D., "Development of a lightweight deep learning model for the identification and classification of Indigenous cattle breeds," *Indian J. Dairy Science*, 78(4), 2025
4. Priyanka et al., "Deep Learning-Based Breed Classification of Cattle and Buffaloes using EfficientNetV2-S," *IJERT*, 15(4), 2026
5. Valsang, A.B., Keshi, A., "Image Based Breed Recognition for Cattle and Buffaloes Using AI," *IJIRT*, 12(8), 2026
6. Cui, Y., Tang, B., Wu, G., Li, L., Zhang, X., Du, Z., Zhao, W., "Classification of Dog Breeds Using CNN Models and SVM," *Bioengineering*, 11(11):1157, 2024
7. Ministry of Fisheries, Animal Husbandry & Dairying, "Breed-wise report of livestock and poultry (20th Livestock Census)," Dept. of Animal Husbandry & Dairying, New Delhi, 2020
8. Shorten, C., Khoshgoftaar, T.M., "A survey on Image Data Augmentation for Deep Learning," *J. Big Data*, 6:60, 2019
9. He, K., Zhang, X., Ren, S., Sun, J., "Identity Mappings in Deep Residual Networks," ECCV 2016
10. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A., "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," AAAI 2017
11. Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., "MobileNetV2: Inverted Residuals and Linear Bottlenecks," CVPR 2018
12. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S., "A ConvNet for the 2020s," CVPR 2022
13. Tan, M., Le, Q., "EfficientNet: Rethinking Model Scaling for CNNs," ICML 2019
14. Yosinski, J., Clune, J., Bengio, Y., Lipson, H., "How transferable are features in deep neural networks?", NeurIPS 2014
15. Buda, M., Maki, A., Mazurowski, M.A., "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, 106:249–259, 2018