Indian Journal of Animal Sciences **96** (7): 598–605, July 2026/Article
https://doi.org/10.56093/ijans.v96i7.180966

Scalable and automated Zebu cattle breed identification system using low-complexity AI models

INDU DEVI[1], SAMARTH GODARA[2]✉ SANJEET K VERMA[1], DIVYANSHU SINGH TOMAR[3], KULDEEP DUDI[4] and TANIYA QAMAR[1]

ICAR–National Dairy Research Institute, Karnal 132 001 Haryana India

Received: 01 July 2026; Accepted: 22 July 2026

ABSTRACT

The objective of study was to investigate computationally efficient, low-complexity ML models for automated image-based identification of important milch zebu cattle breeds, i.e. Sahiwal, Red Sindhi, Hariana and Tharparkar. For each breed, images were captured from frontal and left-side views to ensure a comprehensive visual representation of significant features. The dataset comprised images across eight classes of each breed. Four classification models were studied and compared: CNN, Multi-Layer Perceptron (MLP), Random Forest (RF), and K-Nearest Neighbour (KNN). To enhance computational efficiency, dimensionality reduction techniques were applied where appropriate. The results demonstrated that light-weight CNN model, was top performer, achieving 95% accuracy with high precision, recall, and F1-scores across eight breed-view classes, closely followed by 90% accuracy for MLP and RF and 88% accuracy for KNN, demonstrating robust feature handling post-PCA dimensionality reduction while maintaining low computational demands suitable for field deployment. The study indicated that low-complexity machine learning models can serve as efficient and deployable alternatives to deep neural networks for cattle breed classification, enabling scalable implementation in smart dairy monitoring systems.

Keywords: Automated breed Identification, Convolutional neural network, Image processing, Machine learning, Phenotypic traits

Accurate phenotypic characterisation is a cornerstone for the identification, classification, and conservation of livestock genetic resources (FAO 2012). India has 193.46 million cattle population, out of which 142.10 million (73.5%) are indigenous cattle (Livestock Census 2019). This indigenous population encompasses substantial genetic variability, represented by 55 officially recognised zebu breeds (NBAGR Database 2025). Prominent indigenous dairy breeds such as Sahiwal, Red Sindhi, Tharparkar, and Hariana play a critical role in sustaining milk production systems under tropical climatic conditions. However, accurate breed identification under field conditions remains a significant challenge for farmers due to high phenotypic similarity among several indigenous breeds. Misclassification of animals can adversely impact genetic improvement programs, breed conservation strategies, and herd management decisions. Traditionally, breed identification relies on expert visual assessment, which is often subjective, time-intensive, and not readily accessible in rural and resource-limited settings. Although molecular characterisation techniques provide high accuracy, their dependence on sophisticated laboratory infrastructure and associated costs limits their applicability for routine field-level deployment. These limitations underscore the need for automated, rapid, and cost-effective digital solutions for breed identification.

Recent advancements in computer vision technologies have opened new possibilities for automated livestock phenotyping. The AI-enabled system offers a robust, non-invasive, and scalable approach for cattle breed assessment as well as reproductive and health evaluation (Dahiya *et al.* 2026). Machine learning and deep learning approaches have been increasingly explored for cattle breed classification, demonstrating promising results in controlled experimental settings (Weber *et al.* 2020; Pan *et al.* 2022; Warhade *et al.* 2024). Nevertheless, these models typically require large annotated datasets, high computational capacity, and specialised hardware (e.g., GPUs), thereby limiting their scalability in real-world. Therefore, a significant research gap exists in developing computationally efficient yet highly accurate models that can be realistically implemented for real-time decisions. Such models can assist farmers and breeders in automatically identifying 'true to breed' animals using simple mobile or field-based imaging systems. This

Present address: [1] Scientist, ICAR-NDRI, Karnal-132 001, Haryana. [1]Ph. D. Scholar, ICAR-NDRI, Karnal-132 001, Haryana.[2]Scientist, ICAR-IASRI, New Delhi.[3] Assistant Professor, Rani Lakshmi Bai Central Agricultural University, Jhansi- 284003; [4]District Extension Specialist, Krishi Vigyan Kendra, HAU, Hisar, Panipat, Haryana, 132 103. ✉Corresponding author email: samarth.godara@gmail.com

can support better marketing prices, breeding decisions, specific record management, and livestock trading platforms and help maintain the genetic purity of indigenous breeds. Additionally, they can support government and research institutions in monitoring indigenous breed populations and implementing conservation programs. Use of lightweight models makes it feasible to deploy these systems on low-power devices such as smartphones, embedded systems, and edge-computing platforms, enabling real-time decision support in rural and resource-constrained environments.

In a systematic survey of vision-based cattle identification, Hossain *et al.* (2022) reported that traditional models such as Support Vector Machines (SVM), K-Nearest Neighbour (KNN), and Artificial Neural Networks (ANN) have been commonly applied alongside deep learning approaches like Convolutional Neural Networks (CNN) and residual networks in cattle identification tasks. Jwade *et al.* (2019) obtained an accuracy of 95.8% for sheep breed classification while Weber *et al.* (2020) attained an accuracy of 99.0% for cattle breed identification, and Fu *et al.* (2020) attained a breed classification accuracy of 89.34% for horse breeds. Pan *et al.* (2022) achieved a maximum of 93 % accuracy using SVM for the differentiation of the Nili-Ravi breed from other buffalo breeds. Gupta *et al.* (2022) explored YOLOv4 for breed detection and classification, indicating that object detection-based deep learning models can serve both localisation and classification tasks simultaneously. A lightweight deep learning system for Tharparkar and Hariana cattle was proposed by Arya *et al.* (2025). They proposed a semantic segmentation CNN with classification, confirming deep models' applicability to differentiate phenotypically similar breeds with an accuracy of 72%.

Most existing studies on cattle breed classification primarily focused on deep learning architectures such as CNN variants (ResNet, VGG, EfficientNet), which require large annotated datasets, high computational power, and GPU-based training environments. While these approaches often achieve good performance, they are not always suitable for practical deployment in rural or resource-constrained agricultural settings where computing infrastructure is limited. There is limited understanding of whether simpler models can achieve comparable accuracy with significantly lower computational requirements. Another gap is the limited exploration of multi-view image data, such as combining frontal and side perspectives, which can provide richer visual information for classification. In this context, the present study aimed to bridge the existing research gap by investigating the feasibility of low-complexity machine learning models for accurate cattle breed classification. Particular emphasis was placed on assessing whether computationally efficient models can achieve classification accuracy exceeding 95%, a threshold considered acceptable for practical deployment in real-world applications.

MATERIALS AND METHODS

Image dataset preparation and pre-processing: In this



Fig. 1. Sample images of four breeds from the front and left side

study, a comprehensive dataset of quality images was prepared, from four indigenous cattle breeds: Sahiwal, Red Sindhi, Hariana and Tharparkar to capture distinctive breed-specific characteristics. For each breed, RGB images were taken using mobile phones (resolution 2296×4080 pixels) from frontal and left-side perspectives, as shown in Figure 1, to ensure adequate representation of visual features. The images were captured during the daylight hours, between 7:00 AM to 11:00 AM, and 3:00 PM to 5:00 PM within the natural environment of animal sheds. To avoid any illumination effect, we followed the condition that direct sunlight was not falling on animals indoors or outdoors. A total of 120 cows of each breed (total n= 480 cows) were covered for capturing of images from the farm and nearby villages. The average ages of cows were 6.9±3.0 years (mean±S.D.) for all the breeds. All collected images were carefully reviewed and labelled according to the respective breed and orientation to maintain dataset consistency. The final dataset comprised 2,767 labelled images distributed across eight classes representing breed–view combinations (sample images in Figure 2). Hariana had the largest representation with 873 front-view and 303 left-view images, followed by Tharparkar with 471 front-view and 272 left-view images. Red Sindhi and Sahiwal breeds included 214 and 268 front-view images and 148 and 218 left-view images, respectively, providing a diverse dataset for training and evaluation of the classification models.

The collected images were subjected to a systematic pre-processing pipeline to ensure consistency and suitability for machine learning analysis. Initially, all images were organised according to their respective breed categories, and directory structure was used to derive the class labels. The dataset was then screened to retain only valid image formats while excluding unsupported or corrupted files. To standardise the input representation, all images were resized to a uniform spatial resolution of 128 × 128 pixels. This resizing step ensured dimensional consistency across samples and reduces computational complexity during model training. Subsequently, images were converted into numerical arrays to facilitate algorithmic processing. Pixel intensity values were normalised to a standardised range between 0 and 1 to minimise scale variation and improve model convergence. Both colour and grayscale images were handled appropriately to maintain data integrity. After pre-processing, the processed images and their corresponding labels were consolidated into a structured dataset for downstream modelling. Finally, the curated dataset was stored in a compressed format to enable efficient loading, storage, and reproducible experimentation.

Model development: The model development phase focused on constructing lightweight image classification frameworks using supervised learning. Initially, the pre-processed dataset containing normalized images and corresponding class labels were loaded and verified for consistency. The categorical breed labels were transformed into numerical representations using a label encoding procedure to enable compatibility with ML algorithms. Subsequently, the dataset was partitioned using a 5-fold cross-validation strategy to ensure robust model training and reliable performance evaluation across multiple data splits. The training data were used to develop machine learning models designed with minimal complexity. In the study, the following four models were taken into consideration:

Convolutional Neural Networks (CNNs) are widely used for image classification tasks due to their ability to automatically learn hierarchical spatial features from input images. A CNN applies convolution operations using learnable filters that capture local patterns such as edges, textures, and shapes across the image (Guo *et al.* 2017). The fundamental convolution operation in a CNN layer can be expressed as:

$$y_{i,j}^{(k)} = \sigma\left(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} w_{m,n}^{(k)}, x_{i+m,j+n} + b^{(k)}\right)$$

where x represents the input image, w denotes the convolutional kernel weights, b is the bias term, and $\sigma(\cdot)$ is the activation function. Through stacked convolution and pooling layers, CNNs progressively extract higher-level features that are subsequently used for classification in fully connected layers.

Artificial Neural Networks (ANNs), particularly Multi-Layer Perceptrons (MLPs), are widely used for pattern recognition tasks by learning complex nonlinear relationships between input features and output classes (Thai *et al.* 2012). In image classification using ANN, images are typically flattened into one-dimensional feature vectors and passed through multiple fully connected layers where neurons learn weighted combinations of the input features. The output of a neuron in an ANN layer can be mathematically expressed as:

$$y = \sigma\left(\sum_{i=1}^{n} w_i x_i + b\right)$$

where x_i represents the input features, w_i denotes the corresponding weights, b is the bias term, and $\sigma(\cdot)$ is the activation function. By stacking multiple hidden layers, the ANN learns progressively complex feature representations that enable effective classification of the input images.

Random Forest (RF) is an ensemble learning algorithm that constructs multiple decision trees and aggregates their predictions to improve classification performance and reduce overfitting (Xu *et al.* 2012). In image classification tasks, the input images are first converted into feature vectors, and each decision tree is trained on a random subset of the training data and features using bootstrap sampling. The final prediction is obtained through a majority voting mechanism among all individual trees in the forest. Mathematically, the ensemble prediction can be expressed as:

$$\hat{y} = \text{mode}\ (T_1(x), T_2(x) \ldots T_n(x))$$

where $T_i(x)$ represents the prediction of the i^{th} decision tree

for input x, and the final class $\hat{y}$ is determined by the most frequent prediction among all trees.

K-Nearest Neighbour (KNN) algorithm is a non-parametric, instance-based learning method that classifies a sample based on the majority class of its nearest neighbours in the feature space. In this study, images were first transformed into numerical feature vectors, and dimensionality reduction was applied to improve computational efficiency before classification (Kim *et al.* 2012). During prediction, the algorithm computes the distance between a query sample and all training samples and selects the closest instances. The predicted class is determined by majority voting among these neighbours and can be expressed as:

$$\hat{y} = \text{mode } y_i; |; x_i \in N_k(x)$$

where $N_k(x)$ represents the set of k nearest neighbours of the input sample x, and denotes the class label of the neighbour. To reduce the dimensionality of the extracted image features before applying the KNN classifier, Principal Component Analysis (PCA) was employed to transform high-dimensional data into a smaller set of orthogonal components while preserving most of the variance present in the original data. By projecting the image features onto these principal components, redundant and less informative variations are removed. This process significantly reduces the feature space, leading to lower computational complexity and faster model training. Consequently, the reduced feature set obtained through PCA improves the efficiency of the KNN model while retaining the essential information required for accurate classification.

Hyperparameters tuning and K-fold cross-validation: To enhance the predictive performance of the classification models, hyperparameter tuning was performed using a random search strategy combined with 5-fold cross-validation of the models. Random search explores the hyperparameter space by sampling parameter combinations randomly rather than exhaustively testing all possible configurations, thereby improving computational efficiency. This approach is particularly useful when dealing with multiple parameters and large search spaces. For each model, a predefined range of key hyperparameters was specified, such as the number of neighbours in K-NN, number of trees in Random Forest, the network layer sizes in MLP, and the convolutional filter parameters in CNN. During optimisation, the dataset was partitioned into five equal folds. In each iteration, four folds were used for model training while the remaining fold was used for validation. This procedure was repeated over five iterations, with each fold serving as the validation set exactly once. The performance scores from the five iterations were averaged to obtain a robust estimate of model performance.

Model performance evaluation: To comprehensively assess the performance and reliability of the developed classification models, multiple evaluation metrics, including accuracy, precision, recall and F1-score, were used (Arya *et al.* 2025). In cattle breed classification, datasets frequently contain unequal numbers of images from different breeds, making accuracy susceptible to bias toward majority classes. Precision evaluates the proportion of animals predicted as a particular breed that are correctly classified, whereas Recall measures the proportion of animals belonging to a breed that are successfully identified by the model. High recall means few animals of the target breed are missed. Together, these metrics provide a more comprehensive assessment of classifier reliability and robustness.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Software and hardware details for training the models: The models were developed and executed using the Google Colaboratory cloud-based platform, which provides an accessible environment for machine learning without requiring local high-performance hardware. The free platform offers about 12 GB of RAM and a virtual machine with two Intel Xeon CPU cores (around 2.2 to 2.3 GHz), which are sufficient for training lightweight models on moderate-sized image datasets. The implementation used the Python programming language. Scikit-learn was used for data splitting, label encoding, hyperparameter tuning and the classical models (RF and KNN); TensorFlow with the Keras API was used for neural network models; and Matplotlib was used for visualisation of results. The model code developed in the study is publicly available at https://github.com/ICAR-TECH/NDRI-ZCBRS

RESULTS AND DISCUSSION

Comparison of models' training results for cattle breed recognition: Figure 2(a) shows the training and validation accuracy of the CNN model across epochs. At epoch 0, the training accuracy was relatively low (about 0.56) while the validation accuracy was already higher (about 0.90). As training progressed, the training accuracy rose sharply and stabilised around 0.99, while the validation accuracy increased more gradually to approximately 0.95 and remained stable, which suggests good generalisation with minimal overfitting. Figure 2(b) shows the ANN model: training accuracy started around 0.72 and increased steadily to peaks near 0.98, while validation accuracy fluctuated more and reached around 0.90, indicating some variability in generalisation and mild overfitting, as the model learned the training data faster than it generalised to unseen data. The ANN, therefore, showed more oscillation and slower convergence than the CNN. Figure 2(c) shows the distribution of feature significance computed by the

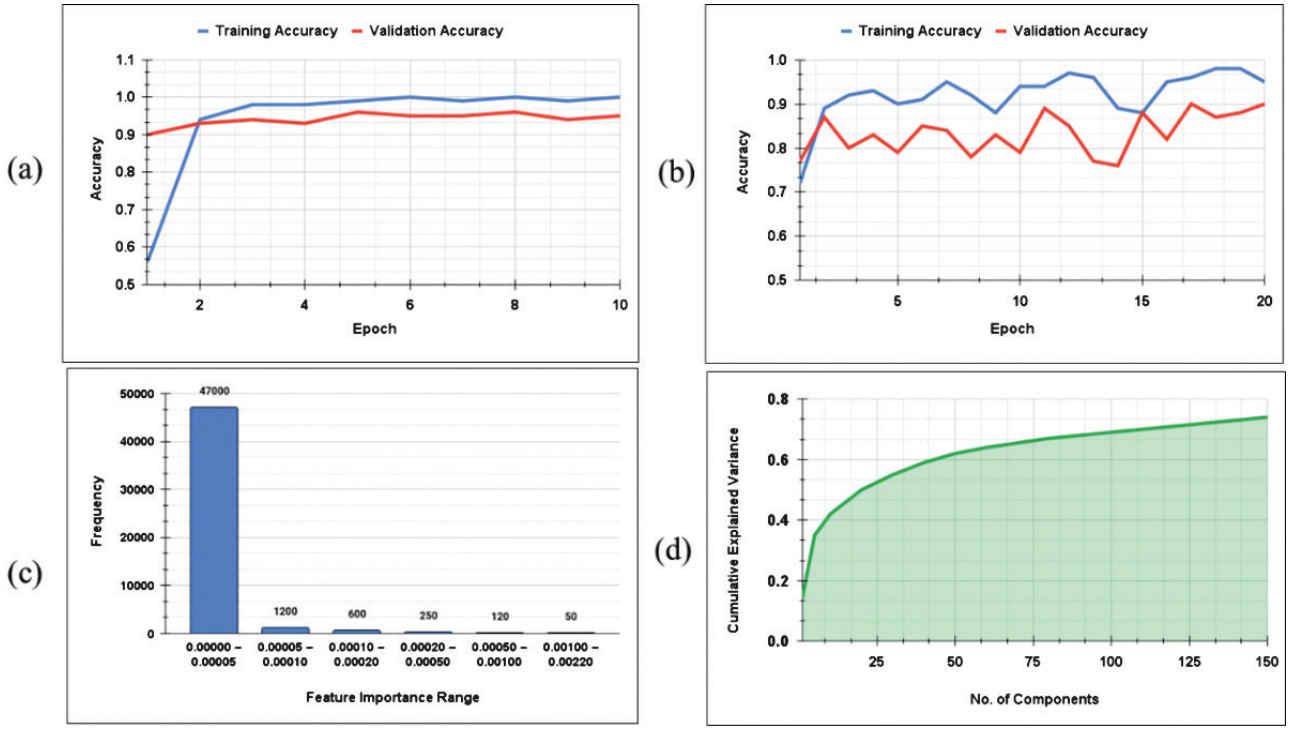


Fig. 2. Training and validation accuracy of (a) CNN-based model (b) ANN-based model, (c) Distribution of feature significance as computed by a Random Forest (RF) model, (d) Cumulative explained variance ratio from PCA

RF model. Most features had very low importance values clustering near zero, while a small number had relatively higher importance, which reflects the sparsity in feature relevance often seen in high-dimensional datasets and highlights the value of dimensionality reduction. Figure 2(d) shows the cumulative explained variance ratio from PCA. The first few dozen components captured most of the variance, and by around 60 to 80 components the curve plateaued near 0.70, indicating that about 70% of image variability was retained with a substantial reduction from thousands of pixels.

In this study, the shallow CNN model achieved 95% accuracy, effectively capturing spatial hierarchies in frontal and lateral profile images, thereby achieving superior performance (Figure 3). The superior performance of the shallow CNN model, across eight breed-view classes, validates its efficacy for automated identification of indigenous Indian cattle breeds like Hariana, Red Sindhi, Sahiwal, and Tharparkar. The MLP and RF models both achieved a solid 90% accuracy, demonstrating robust feature handling post-PCA dimensionality reduction while maintaining low computational demands suitable for field deployment. The KNN model closely followed, with 88% accuracy, offering a simple yet effective baseline for resource-constrained farmers. These lightweight models highlighted the viability of classical machine learning over resource-intensive deep networks for scalable cattle breed identification.

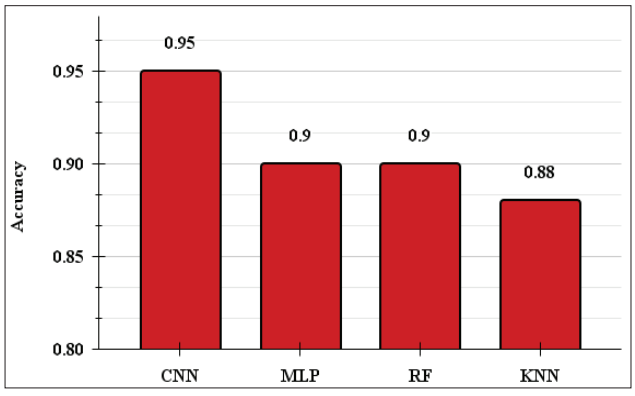


Fig. 3. Accuracy of different models for cattle breed identification

Comparison of models' performance for breed identification in different views of cows: Figure 4 presents the model-wise precision, recall and F1-score together with the confusion matrix of the CNN classifier. For precision, CNN achieved consistently high scores, such as 1.0 for Red Sindhi front and Sahiwal left, averaging around 0.95. MLP peaked at 1.0 for Red Sindhi front but dipped to 0.85 for Tharparkar front and Red Sindhi left. RF was strong in lateral views, reaching 1.0 for Red Sindhi, Sahiwal and Tharparkar left, though it lagged on frontal Hariana at 0.85. KNN delivered top precision of 1.0 for Sahiwal front and Red Sindhi front but struggled on Tharparkar left at 0.75. For recall, CNN gave strong and consistent performance, achieving 1.0 for Red Sindhi left, Sahiwal front and Sahiwal left and about 0.98 for both Tharparkar views and Hariana left. MLP reached 1.0 recall for Sahiwal front and left and Red Sindhi left but dropped to 0.71 for Hariana left and 0.76 for Tharparkar front. RF maintained

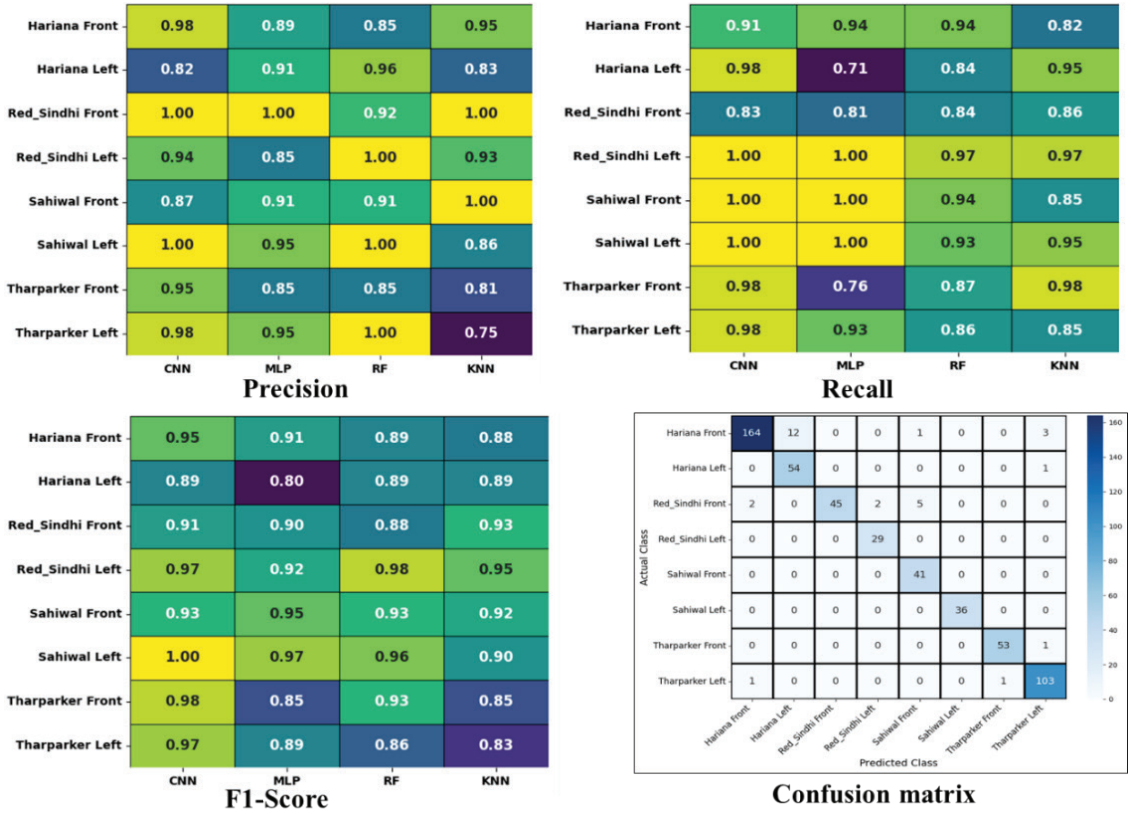


Fig. 4. Model-wise Precision, Recall, F1 Score and confusion matrix for the CNN-based cattle breed classifier for front and left view

recall around 0.84 to 0.97, while KNN performed well on lateral views but lagged on Hariana front at 0.82. For F1-score, CNN again led, with 1.0 for Sahiwal left and high scores near 0.97 to 0.98 for Tharparkar front, Red Sindhi left and Tharparkar left. These metrics affirm CNN as the top lightweight model for breed classification across views, while RF is particularly strong for side profiles.

The confusion matrix for CNN classifier showed strong overall performance, with prominent diagonal elements: 164 for Hariana front, 54 for Hariana left, 45 for Red Sindhi front, 29 for Red Sindhi left, 41 for Sahiwal front, 36 for Sahiwal left, 55 for Tharparkar front and 103 for Tharparkar left. Most misclassifications were intra-breed confusions between the frontal and lateral profiles of the same breed, such as 12 Hariana front instances read as Hariana left, which reflected shared morphological traits between the two views. Errors between different breeds were close to zero, particularly for side profiles, confirming good class separability. Off-diagonal error rates stayed below about 10% per row, consistent with the 95% overall accuracy. Red Sindhi front, the smallest class at 214 images, showed the weakest separation, which indicated that its under-representation limited recall for that class. Comparatively, this study outperformed prior lightweight CNNs for cattle breed classification such as 73.48% accuracy using MobileNet for Horse breed (Fu *et al.* 2020); 85% accuracy using self-activated CNN model for buffalo breed (Pan *et al.* 2020); 79.63% accuracy using InceptionV3 for camel breed (Alfarhood *et al.* 2023); 86%accuracy using

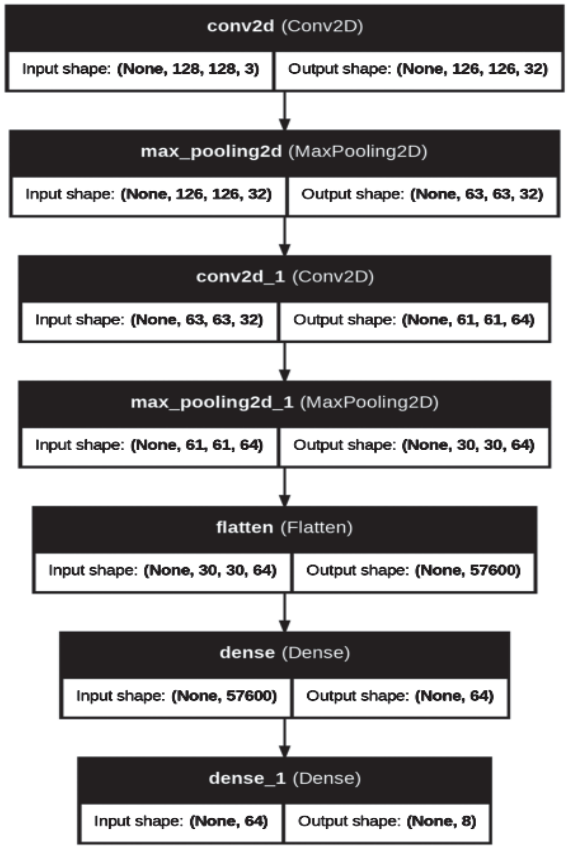


Fig. 5. Model architecture of the light-weight CNN-based cattle breed classifier for front and left view images

EfficientNetB3 (Warhade *et al.* 2024); 72% accuracy using lightweight CNN model (Arya *et al.* 2025) by leveraging dual-view inputs and Google Colab's accessible 12GB RAM/2-core Xeon environment, bypassing high-end GPU needs.

The architecture of the lightweight shallow CNN model is shown in Figure 5. The model processes input images of shape (128, 128, 3) through two sequential Conv2D blocks, each followed by MaxPooling2D for feature extraction and spatial downsampling. The first Conv2D applies 32 filters to yield (128, 128, 32), reduced by pooling to (64, 64, 32), while the second Conv2D with 64 filters produces (64, 64, 64) before pooling, capturing breed-specific traits such as horn shape and coat pattern. Flattening converts the final feature maps into a one-dimensional vector of 57,600 elements, which keeps the parameter count low for resource-constrained devices. A dense layer narrows the representation to 64 units, followed by an output dense layer with 8 units for the eight classes. This design achieved 95% accuracy with low complexity, suitable for edge deployment in precision breed identification systems.

Testing and validation of models on unseen images: Figure 6 includes random test images from an unseen dataset. This exemplifies the CNN model's robust generalisation capability, with all instances correctly classified across Indian cattle breeds like Tharparkar Front/Left, Sahiwal Front, Hariana Front/Left. Each image showcases frontal and lateral profiles under real-world field conditions, validating the lightweight architecture's ability to discern subtle morphological traits such as horn shape (curvature), muzzle shape, and coat texture. Accurate predictions on diverse samples like Tharparkar Front (Actual: Tharparkar Front, Predicted: Tharparkar Front) confirm 95% accuracy. The performance on random new test images showed the robust generalisation capability of the CNN model, with all instances correctly classified across the cattle breeds. Each image showcases frontal and lateral profiles under real-world field conditions, validating the lightweight architecture's ability to discern subtle morphological traits such as horn shape, muzzle shape, and coat texture. The sound performance on Hariana and Sahiwal views highlighted effective feature extraction from PCA-reduced inputs, minimizing overfitting risks in resource-limited environments. Overall, these correctly classified images underscored the model's readiness for scalable deployment. Resource-efficient designs (95% accuracy on Google Colab, edge-ready post-compression) democratized AI for smallholder farmers, integrating with apps for threat detection of endangered breeds and market valuation based on breed purity. Economically, AI-driven phenotyping boost rural incomes by verifying breed authenticity for premium pricing of indigenous zebu breeds.

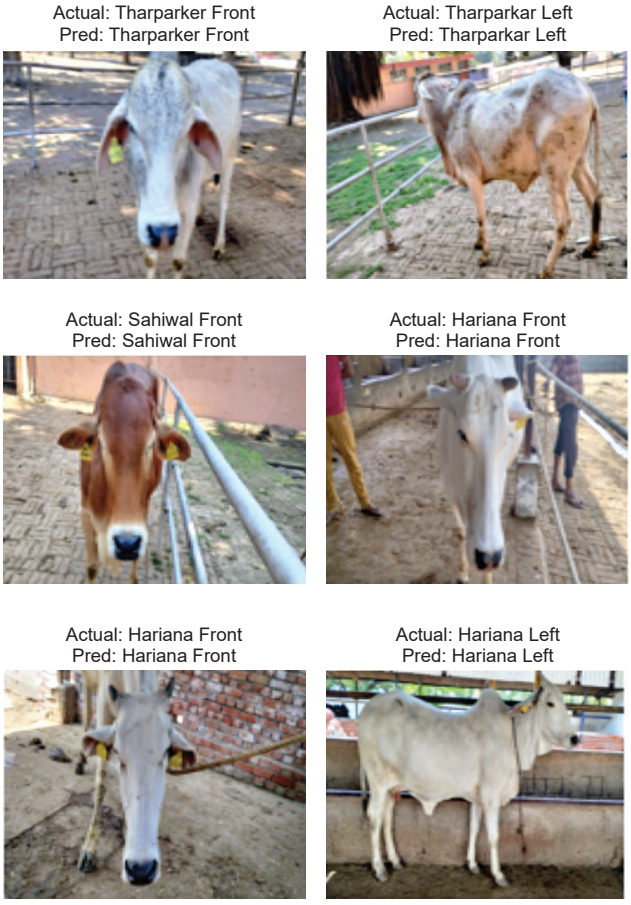


Fig. 6. Developed model's performance on unseen dataset

Some of the limitations of the presented work include class imbalance, which can inflate majority-class metrics such as precision for features of frontal views. Future work can address class imbalance through targeted data augmentation, such as synthetic oversampling, for under-represented classes like Red Sindhi front. Multi-view fusion using attention-based transformers or three-dimensional pose estimation could improve accuracy further, and federated learning across farms could enable collaborative training while preserving data privacy and adapting to regional breed variations. In conclusion, lightweight machine learning models were successfully developed and validated for automated classification of four indigenous Indian cattle breeds using frontal and left-side profile images. The shallow CNN model was the best performer, followed by the classical models for ordinary field devices. These models can support on-device breed identification, selection and conservation of native cattle by farmers and field staff, and balanced sampling with wider breed coverage would extend the approach towards national use.

ACKNOWLEDGEMENTS

The authors are thankful to the Director ICAR-NDRI, Karnal for providing all necessary facilities. Authors express their sincere gratitude to funding agency for providing financial grant from the "ICAR- Network Programme on Precision Agriculture (NePPA)" project (ICT unit-3(5)/2021-ICT).

DATA AVAILABILITY

The computer programs of the models developed in the study are publically available at URL: https://github.com/ICAR-TECH/NDRI-ZCBRS. The data underlying the findings of this study can be obtained from the corresponding author upon reasonable request.

REFERENCES

Alfarhood S, Alrayeh A, Safran M, Alfarhood M and Che D. 2023. Image-based Arabian camel breed classification using transfer learning on CNNs. *Applied Sciences* **13**(14): 8192.

Arya S, Devi I, Tomar D S, Singh M, Warhade R, Gautam A and Mohanty T K. 2025. Development of a lightweight deep learning model for the identification and classification of indigenous cattle breeds. *Indian Journal of Dairy Science* **78**(4): 344–349.

Dahiya N, Bajaj S B, Dalal S, Kumar S and Yadav H C. 2026. Deep learning based approach for detection and measurement of challenging linear traits in Sahiwal cows. *Indian Journal of Dairy Science* **79**(1): 74–79.

FAO. 2012. *Phenotypic Characterization of Animal Genetic Resources.* Animal Production and Health Guidelines No. 11. Food and Agriculture Organization, Rome.

Fu Y, Huang X and Li Y. 2020. Horse breed classification based on transfer learning. (In) *Proceedings of the 4th International Conference on Advances in Image Processing (ICAIP)*, pp. 42–47. ACM.

Guo T, Dong J, Li H and Gao Y. 2017. Simple convolutional neural network on image classification. (In) *Proceedings of the 2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pp. 721–24. IEEE.

Gupta H, Jindal P, Verma O P, Arya R K, Ateya A A, Soliman N F and Mohan V. 2022. Computer vision-based approach for automatic detection of dairy cow breed. *Electronics* **11**(22): 3791.

Hossain M E, Kabir M A, Zheng L, Swain D L, McGrath S and Medway J. 2022. A systematic review of machine learning techniques for cattle identification: datasets, methods and future directions. *Artificial Intelligence in Agriculture* **6**(1): 138–155.

Jogi R, Temburnikar G, Jadhav A, Biradar A, Gajbhiv S and Malge A. 2024. Cattle breed classification techniques: framework and algorithm evaluation. *Journal of Propulsion Technology* **45**(2): 15–25.

Jwade S A, Guzzomi A and Mian A. 2019. On farm automatic sheep breed classification using deep learning. *Computers and Electronics in Agriculture* **167**: 105055.

Kim J, Kim B S and Savarese S. 2012. Comparing image classification methods: k-nearest-neighbor and support-vector machines. *(In) Proceedings of the WSEAS International Conference on Computer Engineering and Applications*, pp. 133–138.

Livestock Census. 2019. *20th Livestock Census: All India Report.* Ministry of Animal Husbandry, Dairying and Fisheries, Government of India, New Delhi.

NBAGR. 2025. *Information System on Animal Genetic Resources of India.* National Bureau of Animal Genetic Resources, Karnal. https://nbagr.res.in/ (accessed 2025).

Pan Y, Jin H, Gao J and Rauf H T. 2022. Identification of buffalo breeds using self-activated-based improved convolutional neural networks. *Agriculture* **12**(9): 1386.

Thai L H, Hai T S and Thuy N T. 2012. Image classification using support vector machine and artificial neural network. *International Journal of Information Technology and Computer Science* **4**(5): 32–38.

Warhade R, Devi I, Singh N, Arya S, Dudi K, Lathwal S S and Tomar D S. 2024. Attention module incorporated transfer learning empowered deep learning-based models for classification of phenotypically similar tropical cattle breeds (*Bos indicus*). *Tropical Animal Health and Production* **56**(1): 192.

Weber F de L, Weber V A de M, Menezes G V, Oliveira Junior A S da, Alves D A, Oliveira M V M de, Matsubara E T, Pistori H and Abreu U G P de. 2020. Recognition of Pantaneira cattle breed using computer vision and convolutional neural networks. *Computers and Electronics in Agriculture* **175**(1): 105548.

Xu B, Ye Y and Nie L. 2012. An improved random forest classifier for image classification. *(In) Proceedings of the 2012 IEEE International Conference on Information and Automation*, pp. 795–800. IEEE.