IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511126

Image Based Cattle Breed Recognition System Using Deep Learning

Santosh Kumar Bhandare[1], Pushpraj Santosh Patil[2], Nirbhay Laxman Salunkhe[3], Harshwardhan Pramod Patil, Sujal Rajendra Shirude[5]

Department of Computer Science and Engineering (Data Science),

R. C. Patel Institute of Technology, Shirpur, Maharashtra, India[1-5]

Abstract: Cattle-breed recognition from photographs can support livestock information systems by providing an initial breed estimate when an expert is not immediately available. This paper presents a 17-class image-classification system covering twelve Indian and five exotic cattle breeds. The working dataset contains 3,400 labelled images, with 200 images assigned to each category. A pretrained Vision Transformer (ViT-Base, patch size 16) was fine-tuned using a 70:10:20 training, validation and test partition. The model was trained for five epochs with AdamW and cross-entropy loss on an NVIDIA Tesla T4 environment. On the held-out test partition of 680 images, the reported accuracy was 83%, macro precision was 0.835, macro recall was 0.828 and macro F1-score was 0.831. The paper also examines the training and validation curves, which show a substantial reduction in loss and an increase in accuracy over the five epochs. The classifier was integrated into CattleVision AI, a web application that displays the top-three predictions, confidence scores and a low-confidence rejection message. Attention rollout is provided as a qualitative visual aid. The results indicate that a pretrained transformer can learn useful breed-related visual patterns from this dataset; however, broader field validation, class-wise analysis and more rigorous unknown-class testing are required before practical deployment.

Keywords: Cattle breed classification, Vision Transformer, image recognition, deep learning, livestock analytics, attention rollout, Indian cattle breeds.

I. INTRODUCTION

Cattle contribute to dairy production, agriculture and rural livelihoods. Breed information is relevant to animal management because breeds differ in physical characteristics and are associated with different production and adaptation traits. In practice, breed identification may rely on visible features such as coat colour, body shape, horn form, facial structure and other morphological cues. Visual assessment can become difficult when animals are young, photographed from unusual angles, partially occluded or captured under changing lighting conditions.

Computer-vision systems offer a way to assist this process by learning visual patterns from labelled photographs. A classifier does not replace veterinary or livestock expertise; rather, it can provide a preliminary prediction that a user may review. The task addressed in this work is single-image, single-label classification among a fixed set of 17 cattle-breed categories. It is distinct from cattle detection, which locates animals in an image, and individual-animal identification, which distinguishes one animal from another.

This study uses a pretrained Vision Transformer and documents its integration into CattleVision AI. The main contributions are: (1) a balanced 17-class working dataset containing Indian and exotic breeds; (2) fine-tuning and evaluation of a ViT-Base model; (3) an interpretation of the training/validation curves alongside the held-out test metrics; and (4) a web interface with top-three predictions, a confidence-based rejection rule and attention-rollout visualisation.

II. RELATED WORK

Deep-learning research in livestock imaging has considered animal detection, face localisation, individual identification and species or breed classification. These problems share image-processing methods but differ in their labels and evaluation protocols. Yadav et al. examined sheep- and goat-face classification using deep-learning approaches [1]. Barbedo et al. studied cattle detection in aerial images captured using unmanned aerial vehicles [2]. Xu et al. evaluated deep-learning methods for multi-view cattle-face detection [3]. Lee et al. and Li et al. investigated individual cattle identification from muzzle-pattern images [4], [5].

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



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511126

The studies above demonstrate that visual information can be used for livestock-related recognition tasks. However, their reported performance values should not be compared directly with the present work because locating an animal, detecting a face, identifying an individual and assigning a breed label are different prediction tasks. The present study focuses on breed-level classification over a fixed set of 17 categories.

III. DATASET AND CLASSIFICATION TASK

A. Dataset composition

The working image collection was assembled from two public Kaggle resources: the Indian Cattle Image Dataset and the Cattle Breeds Dataset [13], [14]. It contains 3,400 images distributed equally across 17 classes. Each class contributes 200 images. Twelve categories represent Indian breeds and five represent exotic breeds. Equal class counts reduce class-frequency imbalance in the assembled dataset, although they do not guarantee that every class contains the same diversity of animals, backgrounds, camera types or image quality.

No.	Breed category	No.	Breed category
1	Amritmahal	10	Mewati
2	Ayrshire	11	Nari
3	Brown Swiss	12	Ongole
4	Dangi	13	Poda Thirupu
5	Gir	14	Purnea
6	Hallikar	15	Red Dane
7	Holstein Friesian	16	Sahiwal
8	Jersey	17	Shweta Kapila
9	Kenkatha		

B. Dataset partition

The images were divided into training, validation and test partitions using a 70:10:20 ratio. This corresponds to 2,380 training images, 340 validation images and 680 test images. The training partition is used to update model parameters, while validation data provide feedback during model development. The test partition is held aside for the final reported evaluation.

Partition	Share	Images	Purpose
Training	70%	2,380	Parameter learning
Validation	10%	340	Monitor generalisation during training
Testing	20%	680	Held-out performance evaluation
Total	100%	3,400	All selected images

The available project record does not confirm whether the split was grouped by animal identity or checked for near-duplicate photographs. This is important because images of the same animal, or very similar photographs, appearing in more than one partition can make evaluation easier than testing on entirely unseen animals. This point is treated as a limitation rather than assuming that leakage occurred.

C. Image preparation

Images are resized to 224 × 224 pixels and processed using the image processor associated with the pretrained ViT checkpoint. Resizing creates a consistent spatial input size, while checkpoint-compatible normalisation aligns the input scale with the representation expected by the pretrained network. The supplied project details do not provide a reproducible list of augmentation operations; therefore, no specific augmentation method is claimed in this paper.

IV. VISION TRANSFORMER AND TRAINING METHOD

A. Model architecture

The selected checkpoint is google/vit-base-patch16-224-in21k [6], [8]. Unlike a conventional convolutional network that processes local neighbourhoods through convolutional filters, a Vision Transformer converts an image into a sequence of patch embeddings. For a 224 × 224 input and a patch size of 16 × 16, the image is divided into a 14 × 14

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

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511126

grid, giving 196 patches. Each patch is projected into a 768-dimensional embedding. A learnable classification token and positional embeddings are added to the sequence.

The sequence is processed by 12 Transformer encoder layers. Each layer uses 12 self-attention heads and a hidden dimension of 768. Self-attention enables the model to combine information from different image regions. In simplified form, scaled dot-product attention is expressed as:

$$\mathbf{Attention(Q, K, V) = softmax(QK^T / \sqrt{d_k})V}$$

Here, Q, K and V denote query, key and value matrices, while d_k is the key dimension [7]. The final classification-token representation is passed to a newly configured linear layer with 17 output units. Softmax converts the output scores into class probabilities, and the class with the largest probability is selected as the top prediction.

$$\hat{p}_k = \exp(o_k) / \Sigma_j \exp(o_j), \qquad \hat{y} = \arg\max_k \hat{p}_k$$

The classification head contains 13,073 parameters, while the complete model contains 85,811,729 parameters. The large pretrained backbone provides general visual representations, while fine-tuning adapts these representations to the cattle-breed categories in the working dataset.

B. Fine-tuning configuration

Training element	Configuration
Pretrained model	ViT-Base, patch16, 224 × 224
Framework	PyTorch and Hugging Face Transformers
Compute environment	Google Colab, NVIDIA Tesla T4 GPU
Optimiser	AdamW
Learning rate	5×10^{-5}
Loss function	Cross-entropy
Number of epochs	5
Output classes	17

During supervised fine-tuning, the model produces a probability distribution over the 17 categories. Cross-entropy loss penalises the model when it assigns low probability to the correct label. AdamW updates the model parameters using gradient information while decoupling weight decay from the gradient-based update [10]. The batch size, random seed, learning-rate schedule, exact software versions and total training time were not recorded in the supplied project details. These settings should be documented in a future reproducibility report.

C. Evaluation measures

Accuracy is the fraction of test images classified correctly. Precision measures how many predictions for a class are correct, while recall measures how many actual examples of that class are retrieved. The F1-score is the harmonic mean of precision and recall. Macro averages calculate the metric independently for every class and then average the 17 class values equally. Weighted F1 averages class F1 values according to the number of test examples in each class. Because the source dataset is balanced by design, macro and weighted summaries are expected to be relatively close, although class-level results are still needed to understand individual breed performance.

V. EXPERIMENTAL RESULTS

A. Training and validation curves

Figure 1 presents the loss and accuracy curves across five training epochs. The horizontal axis represents epochs 0 to 4 as displayed in the supplied plot. The loss plot shows a consistent reduction in both training and validation loss. Training loss falls from approximately 2.26 at the first plotted epoch to approximately 0.37 at the final epoch. Validation loss decreases from approximately 1.68 to approximately 0.52. The downward direction of both curves indicates that the model is reducing its prediction error on the training examples while also improving on the validation examples.

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

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511126

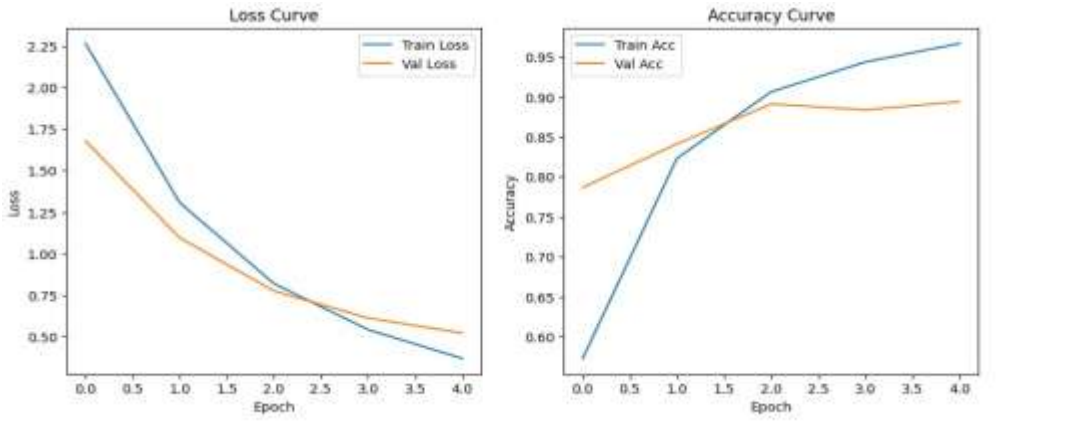


Fig. 1. Training and validation loss and accuracy over five epochs (source: supplied training plot).

The accuracy plot shows a rapid increase in training accuracy from approximately 57.5% to 82.3% between the first and second plotted epochs. It then rises to about 90.7%, 94.4% and 96.7%. Validation accuracy begins at approximately 78.8%, rises to about 84.2% and 89.2%, dips slightly to about 88.4%, and finishes close to 89.5%. These graph-derived values are approximate visual readings; they should not be substituted for exact logged measurements where those are available.

B. Epoch-wise interpretation

The first epoch shows a large gap between training and validation accuracy, with training accuracy substantially lower. By the second epoch, training accuracy improves sharply and approaches the validation curve. During the third epoch, both curves are close to 90%, indicating that the model has learned a stronger representation of the training categories and is also performing well on the validation partition.

In the fourth and fifth epochs, training accuracy continues to improve, whereas validation accuracy remains within a relatively narrow band around 88–90%. At the same time, training loss continues to fall more strongly than validation loss. This pattern suggests that additional training is improving the fit to the training set more than it is improving generalisation. It is compatible with mild overfitting or a validation-performance plateau, but the graph alone cannot identify the cause. A separate experiment using early stopping, stronger data augmentation or weight-decay tuning would be needed to determine whether these changes improve held-out performance.

C. Reported epoch values

Epoch	Train loss	Val. loss	Train accuracy	Val. accuracy
1	2.2636	1.6776	57.27%	78.68%
2	Approx. 1.28	1.0949	Approx. 82.3%	84.12%
3	Approx. 0.82	0.7755	Approx. 90.7%	89.12%
4	Approx. 0.54	0.6123	94.37%	88.38%
5	0.3696	Approx. 0.52	96.68%	Approx. 89%

Note: exact values are reproduced where available in the project record. Values labelled approximate are estimated from the plotted curves or are described as approximate in the project notes. Epoch numbering in the chart is displayed from 0 to 4, while the table labels the same five training passes from 1 to 5.

D. Held-out test performance

Metric	Reported result
Test accuracy	83%
Macro precision	0.835
Macro recall	0.828
Macro F1-score	0.831
Weighted F1-score	0.832
Test loss	0.584

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940



International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026

DOI: 10.17148/IJARCCE.2026.1511126

The final reported test accuracy is 83% on 680 held-out images. Macro precision of 0.835 and macro recall of 0.828 are close, resulting in a macro F1-score of 0.831. The weighted F1-score of 0.832 is also close to the macro F1-score. This similarity is consistent with the balanced class counts in the dataset, but it does not prove that every breed has similar accuracy. The test loss of 0.584 provides an additional measure of the confidence-weighted prediction error.

VI. DISCUSSION

The learning curves and test metrics provide complementary views of model behaviour. The training curves show that the model fitted the training examples increasingly well over five epochs. Validation loss also decreased throughout the plotted period, while validation accuracy improved overall and then stabilised. The final training accuracy of 96.68% is higher than the validation accuracy of approximately 89%, and the held-out test accuracy is 83%. This gap indicates that performance on the training set should not be treated as an estimate of performance on new photographs.

The difference between validation and test accuracy may be influenced by the composition of the random split, visual variation among the photographs, similarities between breed appearances, or differences in the difficulty of the selected images. These are plausible explanations, not proven causes. Without a confusion matrix and per-class precision, recall and F1-score, it is not possible to identify which breed pairs account for the errors. Such analysis is particularly important when two categories share coat colours or body characteristics.

The application also uses a 50% confidence threshold to display “Breed Not Recognized” when the top predicted class score is below the threshold. This rule is useful as a basic interface safeguard, but a softmax score is not automatically a calibrated probability. The threshold has not been selected through a formal calibration or open-set recognition experiment. The rejection behaviour should therefore be understood as a practical interface rule rather than a validated detector for all unknown breeds or non-cattle images.

VII. CATTLE VISION AI APPLICATION

The trained model is integrated into CattleVision AI, a browser-based application. The backend uses FastAPI and Uvicorn to receive image requests and perform inference. The frontend uses HTML, CSS and vanilla JavaScript. The application accepts JPG, PNG and WEBP images up to 10 MB and provides a preview before prediction. After inference, the interface displays the highest-ranked breed, its confidence score and the three leading class predictions.

If the highest score is below 50%, the application presents “Breed Not Recognized”. The interface also provides an attention-rollout display, a copy-prediction function and an option to download a PNG result card. A login and history feature stores the latest 50 prediction records for each registered user in SQLite. A local Ollama-based chatbot is included to answer project-related questions. The supplied project record does not contain measured latency, throughput, usability or deployment-scale results; therefore, no performance claim is made for these application features.

VIII. ATTENTION ROLLOUT

The application includes an attention-rollout visualisation intended to show image regions receiving relatively high attention through the Transformer layers. Attention matrices are averaged across heads, residual connections are incorporated by adding an identity matrix, and the matrices are normalised and combined across layers. The resulting relevance map is projected onto the image patches. In the interface, the highest-ranked 28% of patches are selected and prominent connected regions are highlighted.

This view can help users inspect whether the model appears to focus on the animal rather than an unrelated background region. However, attention visualisation is not a causal explanation and does not establish that the highlighted regions are the true biological cues used for classification. The present project has not validated the visualisations against expert-marked regions or quantitative explanation metrics. It should be presented as an exploratory interpretability aid.

IX. LIMITATIONS AND FUTURE WORK

The experiment has several limitations. The dataset contains only 3,400 images and 17 selected categories, with 200 images per class. The available record does not establish that the train, validation and test partitions are independent at the animal level or that near-duplicate images were removed. Only one training run is reported, and the exact random seed and several hyperparameters are unavailable. Aggregate test metrics are reported without a class-wise breakdown or confusion matrix. The 50% rejection threshold has not been calibrated, attention rollout has not been quantitatively evaluated, and the system has not been tested in a farm field study.



International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 11, November 2026
DOI: 10.17148/IJARCCE.2026.1511126

Future work should expand the dataset using photographs collected across farms, regions, seasons, animal ages, poses and camera devices. Dataset partitioning should be performed by animal identity where possible, with duplicate detection before splitting. Evaluation should report per-class precision, recall and F1-score, a confusion matrix, confidence calibration and repeated runs across multiple random seeds. The model should also be tested on unseen cattle breeds and non-cattle images to measure open-set rejection performance. Comparisons with CNN baselines and other transformer configurations would help establish whether the chosen architecture offers a measurable advantage. Finally, field trials with livestock experts are needed to evaluate usefulness, failure cases and user trust.

X. CONCLUSION

This paper presented a Vision Transformer-based system for recognising 17 Indian and exotic cattle breeds from photographs. The 3,400-image dataset was divided into training, validation and test partitions, and a pretrained ViT-Base model was fine-tuned for five epochs. The supplied experiment reports 83% test accuracy, 0.831 macro F1-score and 0.832 weighted F1-score. The training curves show falling loss and increasing accuracy, while the stabilisation of validation accuracy alongside continued training improvement highlights the need to monitor generalisation. CattleVision AI demonstrates an application workflow with top-three predictions, a low-confidence message and qualitative attention rollout. The current results support further investigation, but larger and more carefully controlled datasets and field validation are necessary before the system can be considered a reliable operational tool.

ACKNOWLEDGMENT

The authors acknowledge the publicly available cattle-image datasets and the computing environment used during model development. Institutional, funding or supervisory acknowledgements should be added only after confirmation by the authors.

REFERENCES

[1] C. S. Yadav et al., “Intelligent Classifier for Identifying and Managing Sheep and Goat Faces Using Deep Learning,” AgriEngineering, vol. 6, no. 4, pp. 3586–3601, 2024.
[2] J. G. A. Barbedo et al., “A Study on the Detection of Cattle in UAV Images Using Deep Learning,” Sensors, vol. 19, no. 24, p. 5436, 2019.
[3] B. Xu et al., “Evaluation of Deep Learning for Automatic Multi-View Face Detection in Cattle,” Agriculture, vol. 11, no. 11, p. 1062, 2021.
[4] T. Lee et al., “Identification of Individual Hanwoo Cattle by Muzzle Pattern Images through Deep Learning,” Animals, vol. 13, no. 18, p. 2856, 2023.
[5] G. Li, G. E. Erickson, and Y. Xiong, “Individual Beef Cattle Identification Using Muzzle Images and Deep Learning Techniques,” Animals, vol. 12, no. 11, p. 1453, 2022.
[6] A. Dosovitskiy et al., “An Image Is Worth 16×16 Words: Transformers for Image Recognition at Scale,” ICLR, 2021.
[7] A. Vaswani et al., “Attention Is All You Need,” Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[8] T. Wolf et al., “Transformers: State-of-the-Art Natural Language Processing,” EMNLP: System Demonstrations, 2020, pp. 38–45.
[9] S. Abnar and W. Zuidema, “Quantifying Attention Flow in Transformers,” ACL, 2020, pp. 4190–4197.
[10] I. Loshchilov and F. Hutter, “Decoupled Weight Decay Regularization,” ICLR, 2019.
[11] A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library,” NeurIPS, 2019, pp. 8024–8035.
[12] J. Deng et al., “ImageNet: A Large-Scale Hierarchical Image Database,” CVPR, 2009, pp. 248–255.
[13] atharvadarpude, “Indian Cattle Image Dataset,” Kaggle, accessed Oct. 30, 2026.
[14] anandkumarsahu09, “Cattle Breeds Dataset,” Kaggle, accessed Oct. 30, 2026.

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