International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 10 Issue: 10 | Oct - 2026 SJIF Rating: 8.659 ISSN: 2582-3930

Multiple Skin Disease Detection for Cattle Using Deep Learning

Kanchana Vijayashree S R

Assistant Professor Department of Computer Science, St. Joseph's First Grade College, Mysore

---

Abstract - Cattle skin diseases are easy to see but hard to confirm in villages with no veterinary clinic nearby. We built a web tool that takes one photograph of an animal and returns the most likely of eight classes: lumpy skin disease, ringworm, mange, dermatophilosis, foot-and-mouth disease (FMD), warts, photosensitization and healthy skin. The classifier is MobileNetV2 with ImageNet weights, trained first with the base network frozen and then with its top 30 layers released. It sits behind a Flask website that has user login and an upload page. If the highest class probability is under 75%, the site declines to name a disease and asks for a better picture. All 15 planned test cases passed on Chrome, Firefox and Edge, on desktop and on phone, and a prediction took only a few seconds. The tool gives a first opinion only; the diagnosis must still be made by a veterinarian.

Key Words: Cattle disease detection, lumpy skin disease, MobileNetV2, transfer learning, convolutional neural network, Flask, veterinary AI

1. INTRODUCTION

A sick cow is a problem for the whole household, not only for the animal. For many Indian families cattle supply milk, draught power and a cash reserve [11], so a skin disease that lowers yield or spreads through the herd hurts quickly. Some of these diseases, lumpy skin disease and foot-and-mouth disease especially, travel fast between animals and can lead to movement bans. Others, such as mange and ringworm, do their harm slowly over months. The usual route to a firm diagnosis is a visit from the veterinarian, sometimes followed by a laboratory test. Where one veterinarian covers a very large number of animals, waiting for days is common.

Photographs offer a cheaper first step. A convolutional neural network (CNN) can learn to separate lesion types directly from pixels, and transfer learning lets us start from a network that has already seen over a million general images instead of training from nothing [5], [6]. Small labelled collections are normal in veterinary work, so this is a real advantage. We used MobileNetV2 [9] because it is light enough for a modest server, and we built a website around it that works in a phone browser. We also made one deliberate choice: the program may decline to answer. Below 75% confidence it shows no disease name.

1.1 Objectives

The aims of the work were to:

- sort cattle photographs into **eight classes**: seven diseases and healthy skin
- prepare the images (**resize, scale, augment**) so that differences in light and camera matter less
- **fine-tune** the top layers of a pre-trained MobileNetV2 on cattle images
- **refuse to name a disease** when the model's confidence is below 75%
- provide a **Flask website** with accounts, upload checks and a result page with care advice
- need nothing but a photograph, and stay **easy to extend** with more diseases

2. LITERATURE REVIEW

Several groups have applied deep learning to livestock skin problems. Umapathi et al. [1] stacked a CNN, a transfer-learned ResNet50 and an LSTM, the last meant to follow a lesion's progress over time, and attached rules that suggest a treatment for whichever disease is found. Yiğitarslan and Kırbaş [2] chose InceptionV3 for digital dermatitis, a painful hoof infection of dairy cows. Their system graded severity from lesion photographs with no expert present.

Most of the work, though, is on lumpy skin disease (LSD). A CNN approach by Shakeel et al. [3] reached about 91% accuracy for early detection. Abdullah et al. [4] tried several architectures and reported 91 to 94%; their useful finding was that pre-trained networks beat ones trained from scratch when data are limited. Genemo [8] segmented the picture to mark the infected areas themselves, with accuracy around 90% or better. A different disease is covered by Park et al. [14], who fed multispectral images to VGG16 to detect photosensitization (88.5%). They also remark that plain RGB photographs usually hold enough information, which is reassuring for a system that depends on ordinary cameras.

Three general references shaped the training recipe. AlexNet [5] brought ReLU, dropout, augmentation and GPU training into the mainstream, with a 15.3% top-5 error on ImageNet. LeCun et al. [6] review why learned features beat hand-designed ones. Shorten and Khoshgoftaar [7] show that flips, crops and colour changes reduce over-fitting on small datasets. On the application side, a review of AI in veterinary diagnosis [13] names the usual obstacles: few labelled images, diseases that look alike, uneven photo quality, and the need for a tool to state its uncertainty. We took that last point literally, hence the threshold.

What we did not find in these papers was one system that handles several diseases, declines weak predictions and is delivered as a website a farmer could use. That combination is what this work attempts.

3. EXISTING AND PROPOSED SYSTEM

Today a farmer who notices skin lesions calls or visits a veterinarian, who inspects the animal and may send samples away. This works only when a veterinarian is within reach. In much of rural India there are few of them for the number of cattle, a sick animal may need transport, and the rest of the herd can be exposed in the meantime. Confirmation costs money and time: PCR for FMD and LSD, scrapings for mange, cultures for ringworm and dermatophilosis [10], [12]. Small farmers often skip it. Judging by eye is uncertain too, since early lesions of different diseases look similar and examiners differ. And because a test targets one suspected disease at a time, checking a whole herd during an outbreak takes long.

In the proposed system a photograph does the first screening. After signing in, the user uploads a picture and gets back, in a few seconds, the likeliest condition with its probability and brief advice. Every class is scored in the same pass, so the runners-up stay visible, and the 75% rule stops weak guesses from looking like diagnoses. No laboratory equipment is involved and an ordinary server is enough to host it.

4. METHODOLOGY

4.1 System Architecture

Figure 1 shows the four layers of the system. A browser page, on desktop or phone, carries the login, upload and result screens. Behind it a security layer verifies credentials, issues session tokens that lapse after a period of inactivity, and blocks anyone not logged in from reaching the prediction routes. Next comes the Flask application, which receives each request, runs preprocessing and inference, and fills a Jinja2 template with the answer. The fourth layer holds the AI parts: the saved model, the image functions and the confidence rule. Accounts, sessions and prediction logs are stored in SQLite or MySQL, while uploaded photos sit in a protected folder on the server. The model is a single .h5 file read at start-up, so it can be replaced without editing the web code.

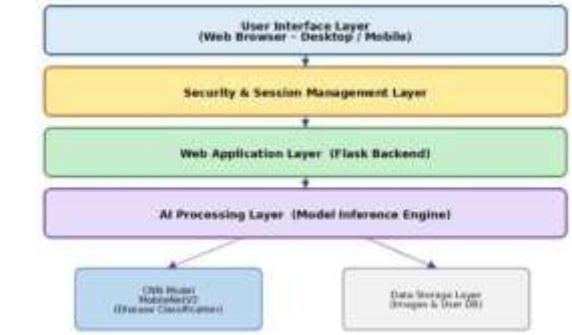


Figure 1: System architecture

4.2 Dataset

Images were collected from public veterinary sources, academic datasets and field photographs supplied by veterinary professionals. Experts assigned each one to a class: Lumpy Skin Disease, Ringworm, Mange, Dermatophilosis, FMD, Warts, Photosensitization or Healthy. Folders for training, validation and testing, each with one sub-folder per class, were filled in an 80:10:10 ratio.

4.3 Preprocessing and Augmentation

OpenCV reads each image, which is resized to 224 × 224 (the size MobileNetV2 was designed for) and divided by 255 to give values between 0 and 1. One extra dimension turns the image into a batch of one. Only training images are augmented, with random horizontal flips, rotation up to 20°, shifts of up to 15% in width and height, up to 15% zoom and brightness changes. Photographs from farms vary in light and angle, and augmentation exposes the network to part of that variation [7].

4.4 Model Design

We picked MobileNetV2 for its low computing cost, since the server may have to answer several users at once. Its ImageNet weights already know about edges, textures and shapes, so only the cattle-specific part has to be learned from our smaller dataset. Training had two stages. First the backbone stayed frozen while a new head was trained: global average pooling, a dense layer of 256 ReLU units, dropout of 0.4 and a softmax layer with 8 outputs. Adam and categorical cross-entropy were used. In stage two the top 30 backbone layers were unfrozen and the whole network was trained again at a learning rate of 0.0001, which lets the upper filters adjust to lesions, crusts and hair loss.

4.5 Training

A GPU workstation was used. Stage one ran for up to 20 epochs at batch size 32 and stage two for 15 more. A checkpoint callback saved the weights with the best validation accuracy, and training stopped early if that accuracy had not improved for 5 epochs. The finished model was saved in Keras HDF5 format.

4.6 Confidence Check

The softmax output gives eight probabilities, and the largest one is the confidence. If it reaches 0.75, the page names the disease and shows the confidence, a short description and care steps; a healthy result gets a confirmation and general husbandry tips. If not, the page leaves out the disease name, shows the message “Image unclear or low confidence. Please upload a clearer image”, and lists tips for retaking the photo. Reviews of veterinary AI ask for exactly this kind of uncertainty reporting [13]. The full list of eight probabilities appears on either page, so a veterinarian can see how close the second choice was.

4.7 Web Application

The Flask routes cover login, logout, home, about and prediction. When an image arrives, the predict route checks its type, saves it under a sanitised file name, preprocesses it, runs the model, applies the threshold and renders the result or warning page; the outcome is also logged. Opening those addresses without signing in redirects to the login page. The pages use Bootstrap 5 and resize for phones, and every result carries a line stating that it is a preliminary screening and no replacement for a veterinarian. The data flow is in Fig. 2 and the sequence of one request in Fig. 3. Development used Python 3.10, TensorFlow/Keras 2.x, Flask 2.x, OpenCV 4.x, Pillow, NumPy and Pandas on Windows 10 or Ubuntu 20.04, on a Core i5 machine with 8 GB of RAM (16 GB, plus a GPU, for training).



Figure 2: Data flow diagram (Level 1)

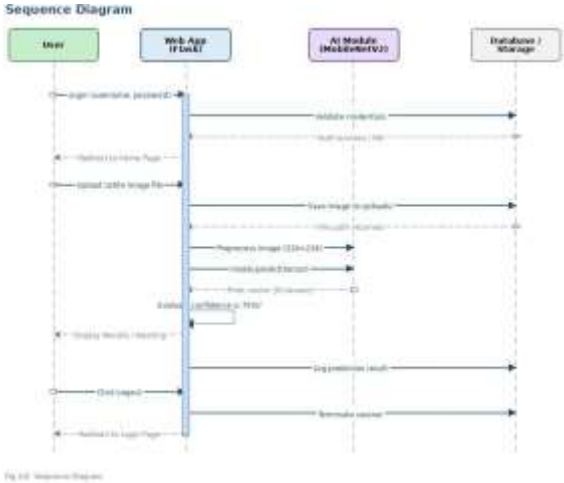


Figure 3: Sequence diagram of a prediction request

5. RESULTS AND DISCUSSION

5.1 Testing

We tested at several levels. Unit tests exercised preprocessing, model loading, the threshold rule, file-type checks and each route; they turned up problems with unusual image sizes and damaged uploads, which we corrected. Integration tests followed an image from the upload form through preprocessing and the model to the result template, and checked that login guards each route. For functional testing, sample images of all eight classes went through the whole workflow, together with blurred, dark and irrelevant pictures to see whether the warning appeared. The site was opened in Chrome, Firefox and Edge on desktop and phone-sized screens, and large images and several parallel sessions stayed within the two-to-five-second target. Direct URL access without login was redirected, and non-image uploads were rejected. Both black-box and white-box methods were applied. All 15 test cases passed; Table I gives ten of them.

TABLE I. Selected test cases

Test	Input	Expected result	Status
Valid login	Correct credentials	Redirect to home page	Pass
Invalid login	Wrong credentials	Error message shown	Pass
Wrong file type	PDF or TXT file	Upload rejected	Pass
LSD image	Clear LSD photograph	LSD predicted, above 75%	Pass
Healthy animal	Healthy cattle photograph	Healthy class shown	Pass
Poor image	Blurred or unclear image	Warning displayed	Pass
No login	Direct URL, no session	Redirect to login	Pass
Large image	High-resolution JPG	Result in under 5 s	Pass
Phone browser	Smartphone	Pages load correctly	Pass
Repeat input	Image tested before	Same prediction	Pass

5.2 Sample Outputs

The upload page (Fig. 4) accepts a dragged or a selected image. In Fig. 5 a photograph with mouth lesions was classified as Foot and Mouth Disease at 97.43%; Photosensitization (1.28%) and Warts (1.12%) came next and all other classes were under 0.1%. The page tells the user to inform the veterinary authorities, since FMD is notifiable, to isolate affected animals, to give soft feed and clean water, and to disinfect equipment and footwear.

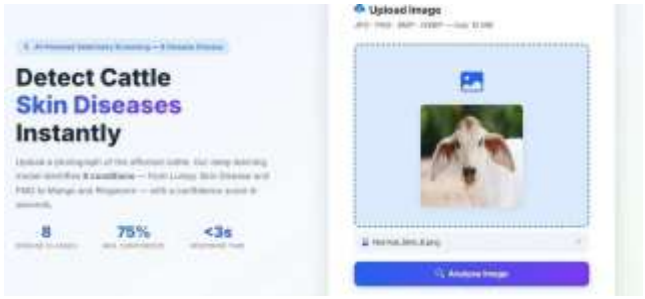


Figure 4: Image upload page



Figure 5: Result page for a high-confidence prediction (FMD, 97.43%)

Figure 6 shows a healthy animal at 98.99%, with routine care advice. Figure 7 is the case we care about most. A poor picture of a cow standing on rough ground gave Lumpy Skin Disease 66.97%, Healthy 17.38% and Mange 7.32%. Since 66.97% is under the limit, the site did not name any disease. It said the image was unclear and gave tips: good light, a close and focused view of the skin, no glare. Without the rule, the farmer might have been told, on thin evidence, that the animal had LSD.



Figure 6: Result page for a healthy animal (98.99%)



Figure 7: Low-confidence warning page (66.97%)

5.3 Discussion

On the validation set the model reached high accuracy, and the system passed every test. These results should not be over-read. Mange, ringworm and dermatophilosis can all show up as crusty bald patches, so some confusion between them is likely. The dataset is of modest size and comes from mixed sources, and photographs from other farms or older phones may behave differently. A single picture also cannot show fever or the other signs a veterinarian would check. A field trial on cases confirmed by a veterinarian is the proper next step; until then the tool should be seen as a screening aid.

6. CONCLUSION

We developed a web-based tool that identifies eight cattle skin conditions from a photograph. A two-stage trained MobileNetV2 does the classification, and a Flask site makes it usable by someone with no technical background: log in, upload, read the answer. The 75% rule keeps doubtful outputs from being shown as diagnoses, and because the model, the

disease list and the hosting are separate parts, each can be changed without touching the others. Our tests indicate that this kind of system can give a quick, inexpensive first opinion where a veterinarian is far away, as long as its limits are stated plainly.

7. FUTURE SCOPE

Next steps include more diseases (for example tuberculosis, brucellosis and nutritional deficiencies), a link to livestock health records, and Android and iOS apps for taking photographs in the field. GPS tagging could map outbreaks. Grad-CAM heat maps would show which part of the image drove a prediction, and Kannada, Hindi, Tamil and other regional-language interfaces would widen access. A cloud deployment would allow many farms to share one service.

REFERENCES

[1] N. Umapathi, B. Vivekanandam and B. K. Mishra, "A deep learning framework for early detection and treatment of livestock skin diseases," SGS Engineering & Sciences, vol. 1, no. 4, 2025.
[2] K. Yiğitarslan and I. Kırbaş, "Classification of digital dermatitis with image processing and machine learning methods," Advances in Artificial Intelligence Research, 2023.
[3] M. Z. Shakeel et al., "A deep learning tool for early detection and control of lumpy skin disease," Journal of Computational Biology and Informatics, 2023.
[4] W. Abdullah, S. Tanwar and M. Abouhawwash, "Deep learning-based detection of lumpy skin disease in livestock using CNNs," Sustainable Machine Intelligence Journal, 2025.
[5] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Advances in Neural Information Processing Systems (NIPS), 2012.
[6] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[7] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," Journal of Big Data, vol. 6, art. 60, 2019.
[8] M. Genemo, "Detecting high-risk area for lumpy skin disease in cattle using deep learning feature," Advances in Artificial Intelligence Research (AAIR), 2023.
[9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2018.
[10] OIE (World Organisation for Animal Health), Manual of Diagnostic Tests and Vaccines for Terrestrial Animals: Cattle Diseases, 2023.
[11] FAO, "Livestock health and disease management," Food and Agriculture Organization of the United Nations, Rome.
[12] O. M. Radostits, C. C. Gay, K. W. Hinchcliff and P. D. Constable, Veterinary Medicine: A Textbook of the Diseases of Cattle, Elsevier, 2010.
[13] J. Thompson, D. Archer et al., "A review of AI applications in veterinary diagnostics," 2022.
[14] S. Park, J. Kim and H. Lee, "Photosensitization detection in cattle using multispectral imaging and CNN," 2023.