COMPREHENSIVE EVALUATION AND PERFORMANCE ANALYSIS OF A DEEP LEARNING MODEL WITH HYPERPARAMETER TUNING FOR LUMPY SKIN DISEASE CLASSIFICATION IN DAIRY COWS

Gunikhan Sonowal[1], Soraisam Gobinkumar Singh[1], Prasanta Bairagi[1], Utpal Barman[1], Dulumani Das[1], Mammadov Iltimas[2], Gulnar Gurbanova[2]

•

[1]Assam down town University, Assam, India
[2]Azerbaijan Technical University, Baku, Azerbaijan
gunikhan.sonowal@gmail.com, soraisam@gmail.com, bairagi@gmail.com, utpal@gmail.com, dulumani@gmail.com, iltimas.memmedov@aztu.edu.az, gulnar.qurbanova@aztu.edu.az

Abstract

This work attempts to classify lumpy skin conditions using CNN and hyperparameter tuning. This model is comprised of many procedures, including selecting a pre-trained model, altering the architecture, and training the model on a specific dataset. During tweaking, the proposed model attained a validation accuracy of 89.73 percent. The model's generalisation performance was confirmed with an accuracy of 80.68% in the final test set evaluation. It significantly increased the timeliness of LSD identification, making it a valuable tool for farmers and veterinarians. Furthermore, a Receiver Operating Characteristic (ROC) curve with an Area Under the Curve (AUC) of 0.88 indicates that our binary classifier performed satisfactorily.

Keywords: LSD; CNN; hyperparameter tuning; Lumpy Skin Disease; Cows

I. Introduction

Lumpy Skin Disease (LSD) is a viral disease affecting cattle, characterized by nodules on the skin, fever, and other systemic symptoms. The disease can lead to severe economic losses due to decreased milk production, weight loss, and increased mortality. According to the FAOSTAT1 production data, India is the leading milk producer globally, holding the top position with a 24% share of world milk production in 2021-22. Over the past eight years, from 2014-15 to 2021-22, India's milk production has surged by 51%, reaching a total of 22 crore tonnes in 2021-22.

Traditional diagnostic methods, such as physical examination and laboratory tests, are often time-consuming and may not be feasible for large herds[5]. In recent years, deep learning techniques, particularly Convolutional Neural Networks (CNNs), have demonstrated considerable potential in the field of medical imaging and disease diagnosis [8, 7]. CNNs, a class of deep learning algorithms designed to process and analyze visual data, have shown remarkable performance in tasks such as image classification, object detection, and disease recognition. These models leverage hierarchical feature extraction and learning capabilities, allowing for improved accuracy and efficiency in detecting and classifying diseases from images.

The application of CNNs to LSD diagnosis represents a promising advancement. CNNs can automate and expedite the diagnostic process by analyzing images of cattle with high precision, potentially addressing the limitations of traditional methods. However, the effectiveness of CNNs in diagnosing LSD is contingent upon several factors, including the quality of the dataset, the architecture of the neural network, and the optimization of hyperparameters. Hyperparameter tuning is a critical step in training deep learning models, as it involves adjusting various parameters to enhance model performance and generalization capabilities. This study aims to develop and evaluate a CNN model for the early diagnosis of LSD in dairy cows. The specific contributions include:

- To preparing a comprehensive dataset of images.
- To designing and implementing a CNN architecture.
- To training and optimizing the CNN model.
- To evaluating the model based on accuracy, sensitivity, specificity, and other relevant metrics.

II. Literature Review

Lumpy Skin Disease (LSD) is a significant viral infection affecting cattle, caused by the Lumpy Skin Disease Virus (LSDV), a member of the Capripoxvirus genus. Therefore, many researchers provided a wide number of models to detect these diseases which are discussed below:

Rai et. al., [7] developed an architecture utilizing machine learning techniques for disease diagnosis and detection. This framework employs tools such as VGG-16, VGG-19, and Inception-v3 for feature extraction. The work was tested on a proprietary dataset and compared with other advanced methodologies, including kNN, SVM, NB, ANN, and LR. The results demonstrated considerable performance in feature extraction.

Girma et al., [3] developed a model for detecting and classifying Lumpy Skin Disease (LSD) in animals categorizing skin conditions into Severe, Mild, and Normal . The dataset was sourced from the Oromia region, specifically Bale Zone's Medawelabu Wereda and Arsi Zone's Chole Wereda Livestock Production Offices, as well as from an external image repository on the internet. Experimental results indicate that the Support Vector Machine (SVM) classifier out- performs both the Random Forest (RF) and Softmax classifiers. The SVM classifier achieved an overall accuracy of 95.7%, whereas the RF classifier reached 87.4%, and the Softmax classifier achieved 94.8%.

Genemo et al., [2] proposed a model for the segmentation and classification of cattle's lumpy skin disease . The framework incorporates a deep learning-based segmentation method and CNN feature optimization. The proposed method was evaluated on well-known datasets for cattle's lumpy skin disease, and the results indicate promising performance. The best classification result achieved in this work is with the ELM classifier, which attained an accuracy of 0.9012. ELM was found to have the overall best performance on the dataset. However, one constraint of our work is the computational time, which will be addressed in future research. Additionally, in future studies, we aim to enhance our segmentation technique to prevent our deep models from training on irrelevant visual features.

Ujjwal et al., [9] aimed to predict the likelihood of cattle contracting lumpy skin disease in a specific geographic region, either in the present or the future, to facilitate timely preventive actions. We applied multiple machine learning algorithms to a dataset containing 18,603 instances and 16 features, with the target column indicating whether the disease occurred (0) or not (1). Among all the algorithms tested, Random Forest achieved the highest accuracy at 97.7%, outperforming other methods in predicting the occurrence of lumpy skin disease.

Patel et al., [6] mentioned that veterinary doctors typically detect Lumpy Skin Disease through manual observation, but it is not possible to detect the disease in its early stages using these methods.

In such cases, AI-based methods can achieve higher accuracy in disease prediction. In this work, a Random Forest-based machine learning model is used to detect Lumpy Skin Disease, utilizing data from Kaggle for training and validation.

Kukreja et al., [4] developed a robust machine learning model for the accurate identification of a variety of skin diseases in cattle. Precision values, ranging from 88.12% to 97.57%, indicate the model's proficiency in distinguishing between different disease classes. With an overall accuracy of 91.56%, the model demonstrates high reliability, crucial for its real-world application in veterinary contexts. By leveraging a comprehensive dataset and advanced machine learning techniques, the model enables timely interventions and treatments, offering veterinarians a valuable tool for managing cattle skin diseases.

This work highlights the potential for future advancements in disease detection strategies, improving animal health and treatment outcomes. This research paper aims to provide a comprehensive evaluation and performance analysis of a deep learning model, specifically a CNN, for LSD classification in dairy cows. Hyperparameter tuning is a critical step in training deep learning models, as it involves adjusting various parameters to enhance model performance and generalization capabilities.

III. Methodology

The following diagram (Figure 1) provides an overview of the model's workflow, from data collection to predicting the presence of cow lump disease. It outlines the key stages of preprocessing, CNN implementation, hyperparameter tuning, and model evaluation. CNNs are particularly effective for image classification tasks, as they capture spatial hierarchies in images, which are crucial for distinguishing between healthy and diseased cows. 3.1.

IV. Data Preprocessing

Dataset Split: The dataset was split into 80% for training and 20% for testing, following a common practice to ensure robust model evaluation. The training set was used to train the CNN model, while the test set was reserved for final model evaluation. Image Normalization: Prior to training, all images were normalized to ensure consistent pixel intensity distributions across the dataset. This step helps to improve convergence during model training by preventing large gradients.

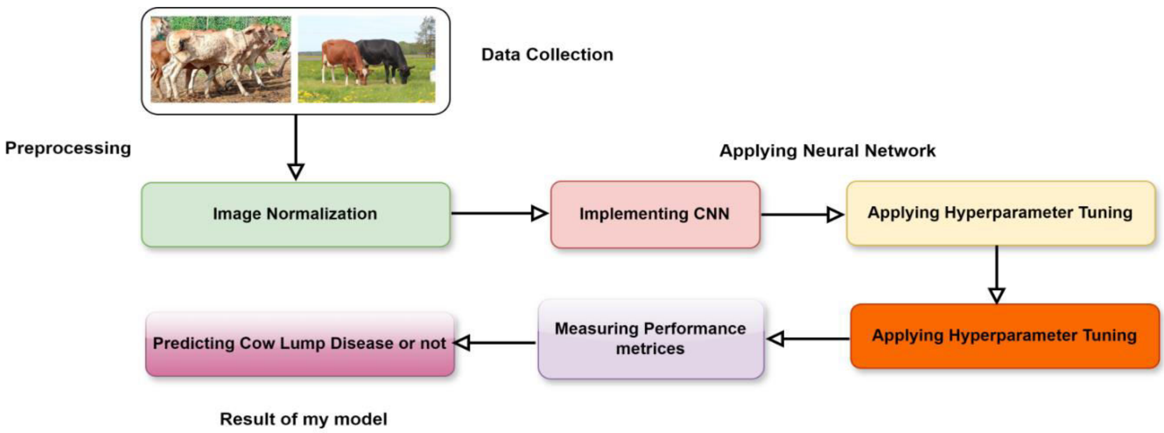

Figure 1: *CNN model*

V. Model Selection and Initialization

For the classification task, a pre-trained CNN model such as VGG16, ResNet50, or InceptionV3 is selected. These models have been trained on large datasets like ImageNet and provide a strong starting point. In this methodology, VGG16 is used. The pre-trained model is loaded without the top fully connected layers, allowing for the addition of custom layers suited to the specific task of LSD classification. The base model is initially frozen, meaning its weights are not updated during the initial training phase.

We opted for a custom CNN rather than a pre-trained model like InceptionV3, ResNet50, or ResNet152V2. This decision was based on the size of the dataset and the need for fine control over the number of filters, layers, and dropout rates. A lighter, customized architecture is often more efficient for smaller datasets where overfitting is a concern.

Pre-trained models like ResNet152V2 are more suitable for large-scale datasets. However, given the nature of the cow disease classification dataset, using a large pre-trained model would have added unnecessary complexity and could lead to overfitting due to the limited data. Instead, we focused on a simpler architecture, using hyperparameter tuning to find the optimal configuration for our specific dataset. 3.3.

VI. Model Architecture

The Convolutional Neural Network (CNN) was designed as follows:

• Input Layer: The model accepts input images of size 256 × 256 × 3 (RGB images).

• Convolutional Layers: The CNN consists of multiple convolutional blocks (1 or 2). Detects local features such as edges and textures with filter sizes ranging from 16 to 64. The activation function used is ReLU.

• MaxPooling Layer: Reduces the spatial dimensions while retaining important features.

• Dropout Layer: Applied to reduce overfitting with a rate tuned between 0 to 0.5.

• Flatten Layer: Flattens the 2D feature maps into a 1D vector to be fed into fully connected layers.

• Dense Layer: A fully connected layer with 64 to 128 units, activated using ReLU, processes the extracted features. Output Layer: A single neuron with a sigmoid activation function is used for binary classification to predict whether the cow has a lump disease or not.

VII. Hyperparameter Tuning

Keras Tuner's Hyperband was used for optimizing key hyperparameters. It tuned the number of convolutional blocks, filter sizes, dropout rates, and units in the dense layer. The search objective was to maximize the validation accuracy. The optimal hyperparameters found during tuning include using 1 convolutional block, 32 filters, 128 dense units, and dropout rates of 0.1 for the convolutional block and 0.4 for the dense layer.

VIII. Result Analysis

To classify Lumpy Skin Disease (LSD) in dairy cows, the first step involves collecting a comprehensive dataset of images. These images should include both affected and unaffected cows. Table 1 shows that 421 affected cows and 515 unaffected cows were collected for experimenting with the proposed model. The Lumpy disease cow is shown in Figure 2a and The Healthy cows are shown in Figure 2b.

Table 1: *Dataset*

Affected Cows	Unaffected Cows	Total Cows
421	515	936

Data preprocessing is crucial for ensuring the quality and consistency of the dataset. All images should be resized to a consistent resolution and format to ensure uniformity. In this model, the images are resized into height 256, and width 256. The batch size parameter is set to 16, which means each batch will contain 16 images. Suppose we have 1,000 images in our dataset. With a batch size of 16, our dataset will be divided into 1000/16 = 62.5 batches. Since we can't have half a batch, there will be 62 full batches of 16 images and one final batch containing the remaining 8 images.

Our next step is to split a dataset into training, validation, and test sets, we allocate a certain percentage of the data to each set. In this model 70% of the total dataset is allocated to the training set, 20% of the total dataset is allocated to the validation set, and 10% of the total dataset is allocated to the test set. We aim to ensure that the data is properly divided into the specified proportions for training, validation, and testing. Each subset can then be used independently for model training, validation, and evaluation.

(a) Lumpy disease cow

(b) Healthy cow

Figure 2: *Sample Images of Dataset*

Scaling data is an essential step in preprocessing, especially when working with image data in neural networks. The provided code snippet scales the image pixel values from the range [0, 255] to [0, 1], which is a common practice to improve the performance of deep learning models.

To define a model-building function for hyperparameter tuning using Keras Tuner, we can use the build model function we provided. This function allows for the tuning of several hyperparameters, such as the number of filters in each convolutional layer, dropout rates, the number of convolutional blocks, and the number of units in the dense layer. This workflow sets up hyperparameter tuning using Keras Tuner with a customizable model. The buildmodel function defines the model architecture with tunable hyperparameters. The RandomSearch tuner searches for the best hyperparameter configuration based on validation accuracy. After finding the best model, we can evaluate it on the test set to get its final performance metrics.

Our next step is to search for the best hyperparameters for our model using the Hyperband tuner, which balances the exploration of a wide range of hyperparameters and the exploitation of promising configurations. The Keras Tuner's Hyperband tuner is used in our model. It is an effective method for hyperparameter optimization. It intelligently allocates resources to different configurations, allowing for an efficient search process.

Once the best hyperparameters are retrieved, the model is built using these hyperparameters. The model is trained with the optimal hyperparameters for a specified number of epochs (80 in this case). The trained model is evaluated on the test set to determine its accuracy. By executing these steps, we evaluated the performance of our trained model on unseen test data and visualized the training and validation metrics over epochs, providing insights into the model's learning process and performance. The performance of the model is shown in Figure 3.

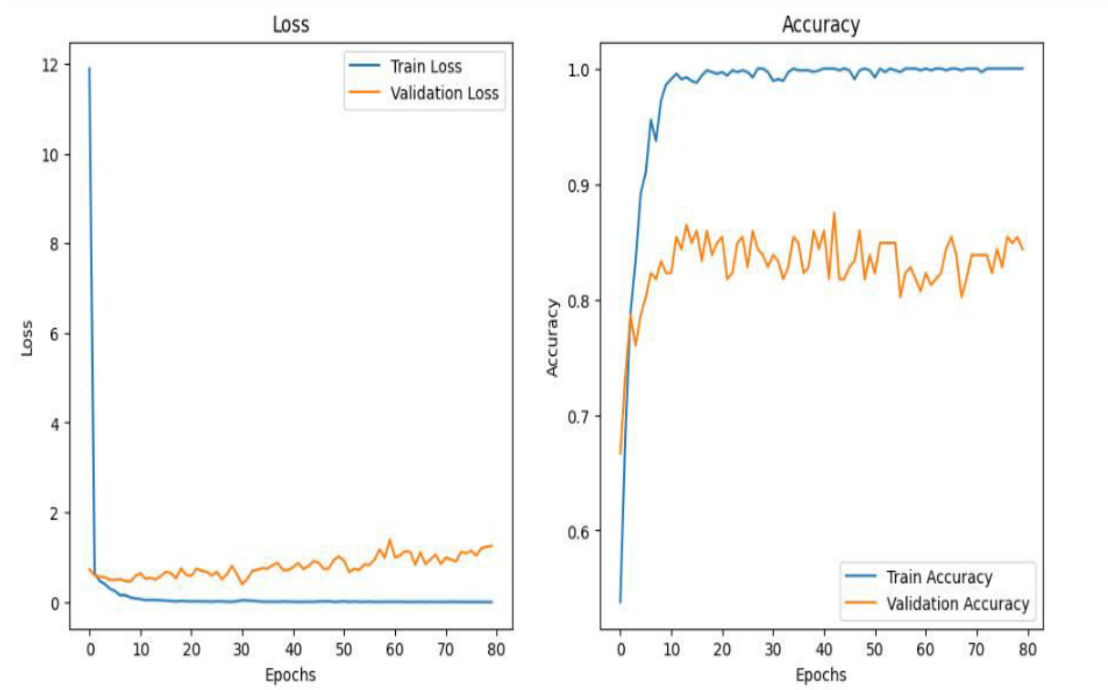

Figure 3: *Test Accuracy and Test Loss*

From Figure 3, it has been seen that the model achieved a test accuracy of approximately 80.68% and a test loss of 1.01. Test Accuracy represents the percentage of correctly classified images in the test dataset and Test Loss value represents the average loss (or error) per sample in our test dataset. Our next step is to evaluate the confusion matrix which is shown in Figure 4.

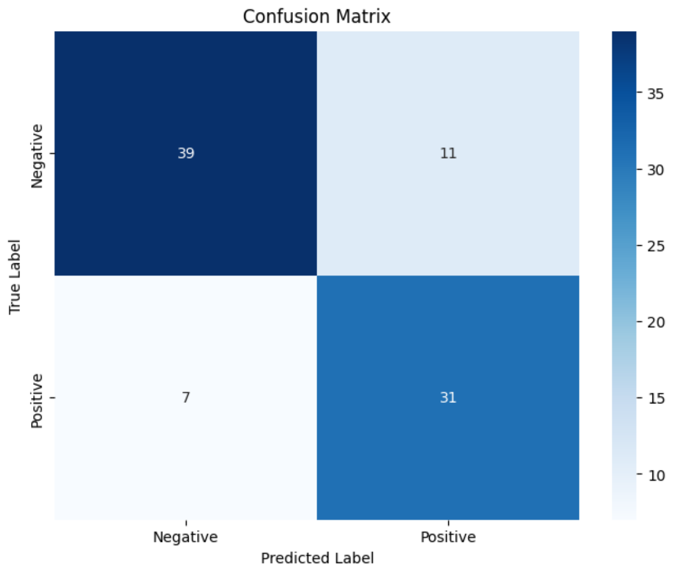

Figure 4: *Confusion matrix*

The confusion matrix provides a detailed breakdown of how well the model performed in classifying instances into positive and negative categories. It shows that out of all instances where the actual class was positive, the model correctly predicted 31 as positive (True Positives). However, there were 7 instances where the model incorrectly predicted them as negative when they were positive (False Negatives). On the other hand, when the actual class was negative, the model correctly predicted 39 instances as negative (True Negatives). There were 11 instances where the model incorrectly predicted them as positive when they were actually negative (False Positives). These metrics, derived from the confusion matrix, offer insights into the model's accuracy, precision, and recall for both positive and negative classes, indicating areas where the model performs well and where improvements may be needed.

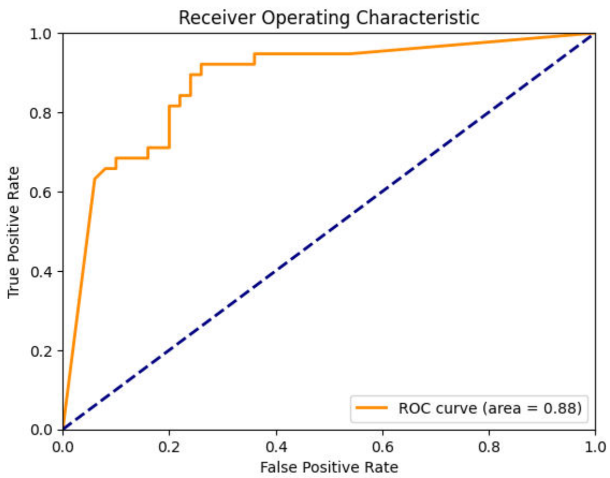

Figure 5: *ROC-AUC*

The final step is to evaluate the ROC-AUC which is shown in Fig 5. A Receiver Operating Characteristic (ROC) curve with an Area Under the Curve (AUC) of 0.88 indicates good performance of our binary classifier. Interpreting AUC of 0.88, AUC of 0.88 suggests that our model has good discriminatory power. It correctly ranks a randomly chosen positive instance higher than a randomly chosen negative instance approximately 88% The higher the AUC, the better the model's ability to distinguish between positive and negative classes. AUC values above 0.5 indicate better-than-random performance, where 0.5 is equivalent to random guessing. In order to justify the models, a comparative analysis is conducted with other existing models as shown in Table 2.

Table 2: *Comparative analysis of other models*

Authors	Accuracy (%)
Workee et al., [10]	80
Ansari et al., [1]	80
The proposed model	80.68.

IX. Discussion

In earlier studies on animal disease detection, various traditional machine learning approaches such as SVM, kNN, and Random Forest have been applied, which rely heavily on feature engineering and may struggle with complex image data. More recent studies have adopted Convolutional Neural Networks (CNNs) due to their ability to automatically learn hierarchical features from images, significantly outperforming manual feature extraction methods.

In addition, Traditional machine learning models rely on hand-crafted features, which can be inadequate for image data with complex patterns. These models are often sensitive to noise and irrelevant features. Some studies apply pre- trained CNN models, which may not be fully optimized for specific tasks like cow disease detection unless fine-tuned appropriately. Although pre-trained models are useful, our approach aims to develop a model optimized specifically for this dataset using custom architecture and hyperparameter tuning, providing a more focused solution.

Unlike traditional methods, CNNs do not require manual feature extraction, which reduces biases and enhances accuracy. CNN architectures can be tailored to various datasets by adjusting layers, filters, and dropout rates, making them more adaptable than static traditional models.

X. Conclusion

This methodology outlines a comprehensive approach to classifying Lumpy Skin Disease in dairy cows using fine- tuning of a pre-trained CNN model. Our study demonstrates that the developed deep learning model shows promise in the automated detection of Lumpy Skin Disease in dairy cows. The combination of accurate classification metrics and strong discriminatory power (AUC of 0.88) supports its potential application in veterinary diagnostics, contributing to early disease detection and proactive management strategies in dairy farming.

References

[1] Ansari, A., Singh, A., Singh, M., Kukreja, V., 2024. Enhancing skin disease classification: A hybrid cnn-svm model approach, in: 2024 International Conference on Automation and Computation (AUTOCOM), IEEE. pp. 29–32.

[2] Genemo, M., 2023. Detecting high-risk area for lumpy skin disease in cattle using deep learning feature. Advances in Artificial Intelligence Research 3, 27–35.

[3] Girma, E., 2021. Identify animal lumpy skin disease using image processing and machine learning. Ph.D. thesis. St. Mary's University.

[4] Kukreja, V., Srivastava, P., Garg, A., Hariharan, S., et al., 2024. Synergizing cnn and random forest for accurate cattle disease identification,

in: 2024 IEEE International Conference on Information Technology, Electronics and Intelligent Communication Systems (ICITEICS), IEEE. pp. 1–6.

[5] Madhanakumar, S., et al., 2024. Lumpy skin disease prediction using machine learning, in: 2024 2nd International Conference on Intelligent Data Communication Technologies and Internet of Things (IDCIoT), IEEE. pp. 887–895.

[6] Patel, S., Thakkar, V., Swain, D., Bhilare, A., 2023. An early lumpy skin disease detection system using machine learning, in: International Conference on Data Science, Computation and Security, Springer. pp. 51–59.

[7] Rai, G., Naveen, Hussain, A., Kumar, A., Ansari, A., Khanduja, N., 2021. A deep learning approach to detect lumpy skin disease in cows, in: Computer Networks, Big Data and IoT: Proceedings of ICCBI 2020, Springer. pp. 369–377.

[8] Saha, D.K., 2024. An extensive investigation of convolutional neural network designs for the diagnosis of lumpy skin disease in dairy cows. Heliyon .

[9] Ujjwal, N., Singh, A., Jain, A.K., Tiwari, R.G., 2022. Exploiting machine learning for lumpy skin disease occurrence detection, in: 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO), IEEE. pp. 1–6.

[10] WORKEE, G., 2021. Cattle skin diseases identification model using machine learning approach. Ph.D. thesis