Sigma J Eng Nat Sci, Vol. 44, No. 2, pp. 856–868, April, 2026

Sigma Journal of Engineering and Natural Sciences
Web page info: https://sigma.yildiz.edu.tr
DOI: 10.14744/sigma.2025.00171

Research Article

Deep learning-based approach for automated detection of lumpy skin disease

Rohit SINGH[1], Ramraj DANGI[2], Lokesh MALVIYA[2,*], Akshay JADHAV[1]

[1]*Manipal University Jaipur, Jaipur, 303007, India*
[2]*School of Computing Science and Engineering, VIT Bhopal University, Madhya Pradesh, 466114, India*

ARTICLE INFO

Article history
Received: 10 October 2024
Revised: 07 December 2024
Accepted: 08 January 2025

Keywords:
Convolution Neural Network; Deep Learning; Lumpy Skin Disease; MobileNetv2; SMOTE

ABSTRACT

The developing viral illness of cattle known as lumpy skin disease (LSD) has terrible economic consequences. The virus responsible for lumpy skin disease is a member of the Poxviridae family and the Capripoxvirus genus. It is an economically significant transboundary disease that affects cattle, water buffalo, and camels. The symptoms of the disease include the development of lumps or nodules on the skin of the animal. This illness is widespread in Africa and the Middle East and has recently appeared in Asia. This article discusses the Lumpy skin disease outbreak and detection among cattle using Mobilenetv2 and Deep learning techniques. The lumpy skin disease dataset is utilized for this experiment and is balanced using the oversampling technique. MobileNetv2, a pre-trained neural network extracts the features from the images for image processing. Later deep learning model with the combination of a two-dimensional convolution neural network, max pooling, flatten and dense layer is utilized for classification purposes. The proposed model outperforms in terms of lumpy skin disease detection; the performance is compared using confusion metrics parameters with a classification accuracy of 99.88%.

Cite this article as: Singh R, Dangi R, Malviya L, Jadhav A. Deep learning-based approach for automated detection of lumpy skin disease. Sigma J Eng Nat Sci 2026;44(2):856–868.

INTRODUCTION

Lumpy disease, also known as lumpy skin disease (LSD), is a viral disease that affects cattle and is caused by the capripoxvirus. Early detection of Lumpy disease is essential to prevent its spread among cattle and reduce economic losses for farmers [1]. The most common detection method is through visual observation of the characteristic skin lesions. More reliable methods of detection include laboratory testing of blood or tissue samples using techniques such as PCR (polymerase chain reaction) and ELISA (enzyme-linked immunosorbent assay) [2].

Cattle, domestic water buffaloes across Africa and even camels [3] were more sus- ceptible to a viral illness known as lumpy skin disease. The migration of infected animals may be responsible for the long-distance spread of LSD. Still, the disease's contrasting seasonal patterns point to the possible involvement of arthropod-borne transmission in its rapid

***Corresponding author.**
*E-mail address: lokeshmalviya@vitbhopal.ac.in.

This paper was recommended for publication in revised form by Editor-in-Chief Ahmet Selim Dalkilic

Published by Yıldız Technical University Press, İstanbul, Turkey

and aggressive short-distance expansion. Recent research suggests that the common house fly, Musca domestica, and other blood-sucking arthropods may have a role in LSDV transmission; however, this has not been shown in a clinical context [4].

Bangladesh reported the first case of the disease in South Asia in July 2019. In August 2019, India announced the first-ever case of Lumpy skin disease. Since May 2022, LSD has spread over various states in India [5]. Vaccination, bovine animal movement restriction, and quarantine are among of the current preventative measures being used, along with the implementation of biosecurity through vector management. In India, the standard method of treatment is vaccination with a goat pox vaccine that provides 60-70% protection against Lumpy skin disease. This vaccine takes 15-20 days to become fully effective. The Indian Council for Agriculture Research National Research Centre on Equines (ICAR-NRCE) in Hisar (Haryana) and the Indian Vet- erinary Research Institute (IVRI) in Uttar Pradesh have been working together to produce an indigenous vaccine called Lumpi-ProVacInd. There is no proof that it has any effect on people. There is a greater variation in sickness and mortality rates among cattle in the current epidemic in India compared to the previous outbreak [6]. LSD may also reduce the export of livestock and livestock goods [7].

The research mentioned in this article focuses on predicting LSD using machine and deep learning methodologies. Many academics have been working on LSD classi- fication during the last decade in order to develop very efficient machine learning and deep learning models. Because of the increasing accuracy and efficiency of prediction, the use of artificial intelligence and machine learning algorithms has gained promi- nence in recent years. The importance of research in this sector is in the capacity to construct and choose models with optimum accuracy and efficiency while minimising false negatives. The paper utilised image datasets of cattle infected and disinfected are two major classifications. The dataset is pre-processed and the later model is pre-trained using MobileNetv2, convolution neural network (CNN) models, utilised for feature extraction and classification respectively. Hyper-parameter tuning is per- formed for classifiers using a Keras tuner, and the performance of the proposed model is compared with the individual models and the existing work in the same domain over confusion metric parameters; Accuracy, Precision, F1-score, and Recall. The graphical comparison is also shown in the form of ROC and convergence curves. The overall architecture of the lumpy skin disease detection system adapted in this research is explained in Figure 1.

Key Contributions

- The dataset is pre-processed in order to remove unwanted noise and extract distorted images from the lumpy image dataset.
- The LSD dataset is imbalanced, in order to handle imbalanced data synthetic minority oversampling technique (SMOTE) is utilised.
- After successful preprocessing, the features are extracted from the images using MobileNetv2, which itself is a neural network-based classification algorithm for image datasets.
- The hyper-parameter tuning is performed for classifiers using a Keras tuner.
- The classification of the images as infected and disinfected is performed using a two-dimensional convolution neural network model.
- The performance of the models is evaluated using confusion metric parameters, ROC and convergence curves.

RELATED WORK

This sub-section describes the existing work and literature in the field of LSD by various researchers. The paper [8] used machine learning techniques to predict the prevalence of Lumpy skin illness using meteorological and geological factors. In order to predict the prevalence of a disease in unseen (test) data, the author first utilized the Extra Trees

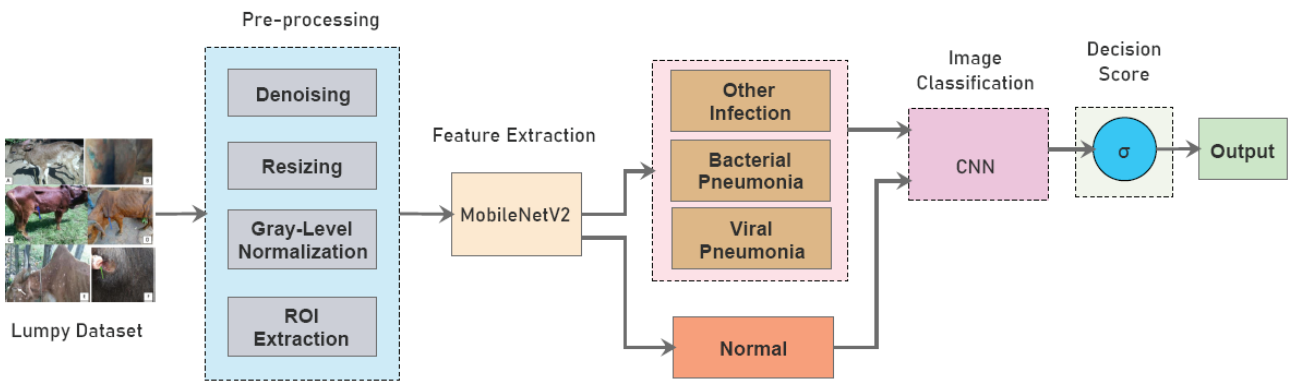

Figure 1. Basic architectural diagram of lumpy disease detection system.

Classifier method to pick out the most essential predictive factors, such as weather, animal population density, dominating land cover, and elevation. Test results showed that some machine learning methods were up to 97% accurate in predicting the development of the Lumpy skin illness. In predicting the existence of Lumpy skin infection in unseen data, the ANN model outperformed other machine learning algorithms, with an AUC and F1 performance measure score of 0.97 and 0.94, respectively.

This qualitative risk assessment was undertaken by the authors of [9] to learn more about the potential for the arrival of the LSD virus to Ukraine and the subsequent spread of the disease inside the country in the next year. The suggestions of the OIE import risk analysis for animals and animal products were used to inform a risk assessment that was carried out with the assistance of local experts assembled in an expert elicitation workshop. The Delphi approach was updated to get advice from specialists.

The research of [10] centered on the detection of illness in cattle. A mobile phone camera was used to record the outward manifestations of sickness in cattle. Information on palpated symptoms was gathered using conversational text. The picture analysis subsystem used a convolutional neural network (CNN) method to classify the symptoms into their respective categories. With 95% accuracy, the system categorized the given symptoms. The reasoner part of the expert system drew the final diagnostic conclusion by combining picture categorization results, location, and text input from the users. As proof of concept, we built a system that uses picture classification techniques and a reasoner. The created system's assessment results confirmed the new diagnostic system's capability of providing an efficient and timely diagnosis of cattle illnesses.

This article [11] presented a model to forecast if the cattle in a given geographic region are at risk of contracting Lumpy skin infection illness now or in the future. In this research, the authors use a dataset of instances of lumpy skin disorders to test the efficacy of multiple ML algorithms. Target columns in the dataset may take on values of 0 (showing the absence of lumpy sickness) or 1 (representing the presence of lumpy disease), and there are a total of 18603 cases and 16 characteristics. The authors found that of all the algorithms they examined, Random Forest performed the best.

Deep learning-based segmentation and classification using deep features are suggested for disease classification in this study [12]. Convolutional neural networks with 10 layers were used for this purpose. The framework is tested on a dataset built from gathered data. Due to the importance of the skin's color in determining the location of the damaged region during disease representation, we employed a color histogram to extract features from the input photos. A deep, pre-trained CNN is then utilized to extract features from the segmented region of impacted skin color. The output is then threshold-converted to a binary representation. Classification is performed using the Extreme learning machine (ELM) classifier. On CLSD, the suggested method's classification performance was able to reach an accuracy of 0.9012%.

The goal of this article [13] was to find the optimal settings for a re-sampling strategy for imbalanced datasets (SMOTE) and a hyper-parameter tuning strategy (Genetic Algorithm; GA) for the classification method (Random Forest; RF) while predicting LSD. Experiments indicated that combinations of SMOTE and GA in the RF algorithm improved performance measures such as Recall and F1-scores from 0.90 to 0.99 and AUC from 0.94 to 0.98. If the model has an accuracy of 0.99, it can reliably identify healthy cattle.

The study [14] is evaluated using a dataset that the authors themselves compiled, and its performance in feature extraction is compared to that of other, more cutting-edge methods including KNN, SVM, NB, ANN, and LR. Additionally, VGG-16, VGG- 19, and Inception-v3 were used to extract the features. Finally, Lumpy skin and normal skin were both successfully predicted using this deep convolutional neural network, with an accuracy of 92.5%.

The goal of this study [15] was to provide a web-based system for diagnosing skin diseases such as atopic dermatitis, acne vulgaris, and scabies using a convolutional neural network classifier built on top of the TensorFlow framework. The system was given the name medical-plus.

Red, Green, and Blue (RGB) colour characteristics and Grey Level Co-occurrence Matrix (GLCM) texture features were recovered from digital camera photos of chronic eczema, lichen planus, and plaque psoriasis gathered by the authors [16].

Comparison with Existing Surveys

In this section, Table 1 provides a comprehensive comparison of the discussed research articles, focusing on various datasets used for the analysis of Lumpy Skin Disease (LSD) and other skin diseases. The table highlights the application of machine learning and deep learning methods in these studies.

Challenges of existing approaches

The challenges in the field of skin disease analysis using machine learning and deep learning methods include:

- The creation of reliable models is hampered by the scarcity of complete outbreak data for training and validation.
- Future transmission patterns cannot be reliably predicted since qualitative judgements are used instead of solid facts.
- Accurate symptom categorization is challenging due to a lack of available annotated pictures for training CNN models.
- The generalizability of models that perform well on certain datasets might be compromised by overfitting.

The rest of the article is organized as follows; Section 1 contains introduction and literature related to LSD, along

Table 1. A state-of-the-art description on existing work

Ref	Proposed Model	Methodology	Dataset
[8]	ANN model for predicting Lumpy Skin Infection	The author demonstrated that the ANN performed better than other machine learning approaches in predicting the presence of Lumpy Skin Infection in unseen data. The model achieved an Area Under Curve (AUC) score of 0.97 and an F1 performance measure score of 0.94.	Lumpy skin disease outbreak data.
[9]	Qualitative Risk Assessment	The authors conducted a qualitative risk assessment to evaluate the likelihood of the LSD virus entering Ukraine and its subsequent transmission in the next year.	Live animals data.
[10]	CNN-based symptom classification	A CNN-based method was used for picture analysis to classify symptoms into their respective categories. The system achieved 95% accuracy in symptom categorization.	Debre Markos University, Addis Ababa University-School of Veterinary Medicine, the World Wide Web, and other secondary sources were used for photographic holdings.
[11]	Random Forest algorithm for Lumpy Skin Disease	The authors determined that the Random Forest method had the highest accuracy 97.7% among the tested algorithms.	Lumpy skin disease dataset with 18,603 instances and 16 features
[12]	CNN-based feature extraction and ELM classification	A pre-trained CNN was employed to extract features from segmented regions of impacted skin color. The ELM classifier was used for classification, achieving an accuracy of 0.9012%.	Dataset of different Cattel's Lumpy Skin Disease.
[13]	Random Forest algorithm with SMOTE and GA	Including SMOTE and GA to improved performance measure, such as Recall, F1 score, and AUC. The Recall and F1 score improved 0.99, and AUC improved 0.98.	Lumpy Skin Disease (LSD) dataset with 24,803 data.
[14]	CNN using VGG-16, VGG-19, and Inception-v3 for feature extraction	The developed framework successfully predicted both lumpy skin and normal skin using deep CNN, achieving an accuracy of 92.5%. The performance was compared with other advanced methodologies such as kNN, SVM, NB, ANN, and LR.	Dataset created by the authors.
[15]	Medilab-plus CNN classifier	The suggested method achieved high accuracies in classifying atopic dermatitis (88%), acne vulgaris (85%), and scabies (84.7%) in experimental settings.	Sample data (pictures) from four health care facilities in the Sunyani Municipality of Ghana.
[16]	ML methods for classification	Classifier performance was evaluated using ML methods such as ANN, LDA, Naive Bayes, and SVM for various feature combinations. LDA and SVM performed best in a test of four different classification methods.	Patient's images from the Department of Dermatology, Yenepoya Medical College, Mangaluru

with the comparison of existing surveys and challenges. Section 2 describes the methodologies used in the research. The proposed work is defined in Section 3, along with dataset description, data preprocessing, balancing techniques, and hyper- parameter tuning. Section 4 describes the performance metrics used for the evaluation of the proposed models. The evaluated results are presented and discussed in Section 5. Section 6 concludes the research findings and provides the future scope of further research in this field.

PRELIMINARY

This section includes a description of the methodologies used in this research. It includes the description of data balancing techniques, feature extraction techniques to extract the features from the image dataset and classification techniques for the identification of infected and disinfected cattle.

Synthetic Minority Oversampling Technique (SMOTE)

To overcome the issue of class imbalance, the classification problem of lumpy disease detection strongly depends on data balancing. A class imbalance might manifest itself, for instance, in a considerably lower proportion of lumpy image cases compared to non-lumpy image cases or pneumonia patients. To categorise lumpy diseases, we have acquired a lumpy dataset, which brings some degree of irregularity into the dataset. In the proposed system, we have used the SMOTE Algorithm [17, 18] to balance the data. To create a separate line between the samples in the feature space and create a unique example along the line, the SMOTE algorithm first chooses examples that are close to the feature space. First, a sample from the underrepresented group is randomly chosen. The nearest k neighbours to the sample are then found. K is often equal to 5. At some point in the feature space between the two samples, a neighbour is chosen at random, and a synthetic sample is created. The practice is repeated until the proportion of the minority group equals that of the majority group.

Feature Extraction Models

The paper utilises feature extraction techniques to extract the features of the image dataset. This section discusses various feature extraction techniques and their advantages [19, 20].

DenseNet-121

A popular deep-learning architecture for extracting features from images is DenseNet-121 [21]. In 2017, scientists from Facebook's AI Research Lab unveiled the concept of tightly linked convolutional layers [22]. Each layer in a DenseNet-121 architecture is fed feature maps from the layers above it and contributes its feature maps to the layers below it. This highly interconnected structure reduces the impact of disappearing gradients and promotes feature reuse [23]. DenseNet-121's dense connections may also be seen as a kind of feature aggregation, enabling the network to pick up more evocative elements. DenseNet-121, like ResNet-121, is a pre-trained model useful for extracting features from images. DenseNet-121's intermediate features may be utilized as input to a new classifier or for other downstream tasks like object recognition or picture segmentation by eliminating the final classification layer. DenseNet-121 is often more memory-efficient and has fewer parameters during training and inference than ResNet-121. However, owing to its high connection structure, it may be costlier to compute.

InceptionV3

Google's InceptionV3 [24] deep learning architecture was released in 2015. It is a kind of convolutional neural network (CNN) that is often used for feature extraction and image categorization. To provide a more expressive feature representation, InceptionV3 is notable for its novel design, which consists of numerous branches of convolutional layers with varying kernel sizes. This architecture was created with the idea that many pictures identification jobs benefit from capturing both fine-grained and coarse-grained characteristics at distinct sizes. Simply deleting the model's final classification layer will allow you to use the intermediate characteristics that InceptionV3 collected for downstream tasks like object identification or image segmentation. There are several ways in which InceptionV3 excels above other deep learning models. Parameter efficiency enables it to attain excellent accuracy with fewer parameters than other designs, while the use of numerous branches of convolutional layers allows it to collect information at various sizes. However, the utilization of many branches may cause its computational cost to be greater than that of other structures.

ResNet50

ResNet50 [25] is a deep learning architecture with similarities to ResNet-121 but with additional layers and enhanced feature extraction capabilities. Microsoft Research first introduced it in 2016, and since then, it has seen extensive use in the fields of feature extraction and image categorization. ResNet50 is a 50-layer deep convolutional neural network (CNN). It employs residual blocks, much like ResNet-121, to facilitate the learning of very deep neural networks. ResNet50's shortcut connections simplify the process of training deep neural networks and reduce the occurrence of vanishing gradients. ResNet50, like the other stated deep learning architectures, may be used as a pre-trained model for image feature extraction. ResNet50's intermediate features may be used as input to a new classifier or for other downstream tasks like object identification or picture segmentation by deleting the model's final classification layer. ResNet50 is more capable of feature extraction since it has more layers than ResNet-121. This has the potential to improve performance on increasingly difficult picture identification tasks, but it may need more computing power and training data.

VGG19

In 2014, researchers at Oxford University proposed a deep learning architecture called VGG19 [26]. It's a common tool for image categorization and feature extraction since it's a convolutional neural network (CNN). VGG19's intermediate features may be used as input to a new classifier or for other downstream tasks like object recognition or picture segmentation by deleting the model's final classification layer. There are a number of ways in which VGG19 excels over competing deep learning systems [27]. It uses modest filters to extract characteristics at various scales and has a unified design that makes it easy to learn and execute. The high number of model parameters, however, may make VGG19 costlier to compute than other designs.

MobileNetV2

MobileNetV2 [28] is a deep learning architecture that was introduced by Google in 2018. It is a convolutional neural network (CNN) that is designed for mobile and embedded devices, but it can also be used for image classification and feature extraction tasks on larger machines. MobileNetV2 has several advantages over other deep learning architectures. Its lightweight architecture makes it ideal for resource-constrained environments, and its use of depth-wise separable convolutions allows it to achieve high accuracy with fewer parameters and faster inference times. However, MobileNetV2 may not be as accurate as larger architectures such as ResNet50 or VGG19 for more complex image recognition tasks.

Classification using CNN

To classify the diseased and healthy cattle using images, the section discusses the detailed explanation of the layered deep learning classification models utilised in the experiment. Convolutional neural networks (CNNs) [29] [30] are a popular kind of deep neural network design for image categorization problem. Features are mapped to class probability in the fully connected layers at the network's output node. Adding additional layers, more filters, or more sophisticated methods like residual connections or attention processes may all make a CNN more complicated. These enhancements allow the network to pick up more complicated information and perform better on difficult picture categorization tasks.

A CNN typically consists of multiple layers [31], including:

Convolutional layers: Input data, which is often an image, is processed by these layers using a series of filters. When the filters pass over the picture, they calculate dot products between their components and the images.

Activation functions: These functions are used to incorporate non-linearity into the model and are applied element wise to the convolutional layer's output. In convolutional neural networks (CNNs), the rectified linear unit (ReLU) function is the most used activation function.

Pooling layers: To reduce the computational burden on the model and limit overfitting, these layers reduce the size of the feature maps.

Flatten layer: Flatten layer is typically used to transform the multi-dimensional feature maps into a one-dimensional vector.

Fully connected layers: The final classification or regression operation is carried out by these layers, which compute dot products between the feature maps and a set of weights before feeding the result into an activation function to provide a prediction.

Dropout layer: Overfitting may occur in deep neural networks, however regularisation methods like dropout can help. Figure 2 shows the pictorial representation of CNN architecture.

PROPOSED METHODOLOGY

The section describes the proposed methodology of the research paper, along with the description of the lumpy image dataset utilised. The section also narrates the flow of the proposed work in varying sub-sections.

Dataset Description

Secondary data was utilised for this research. The dataset, titled "Lumpy Skin Disease (LSD)", had 24,803 records and was downloaded from the Mendeley database [8]. There are only 3,039 records in this dataset about livestock with LSD (12.25%), whereas there are 21,764 records in this dataset about cattle without LSD (87.75%). It indicates an unbalanced dataset. The LSD dataset's properties are made up of geographical and

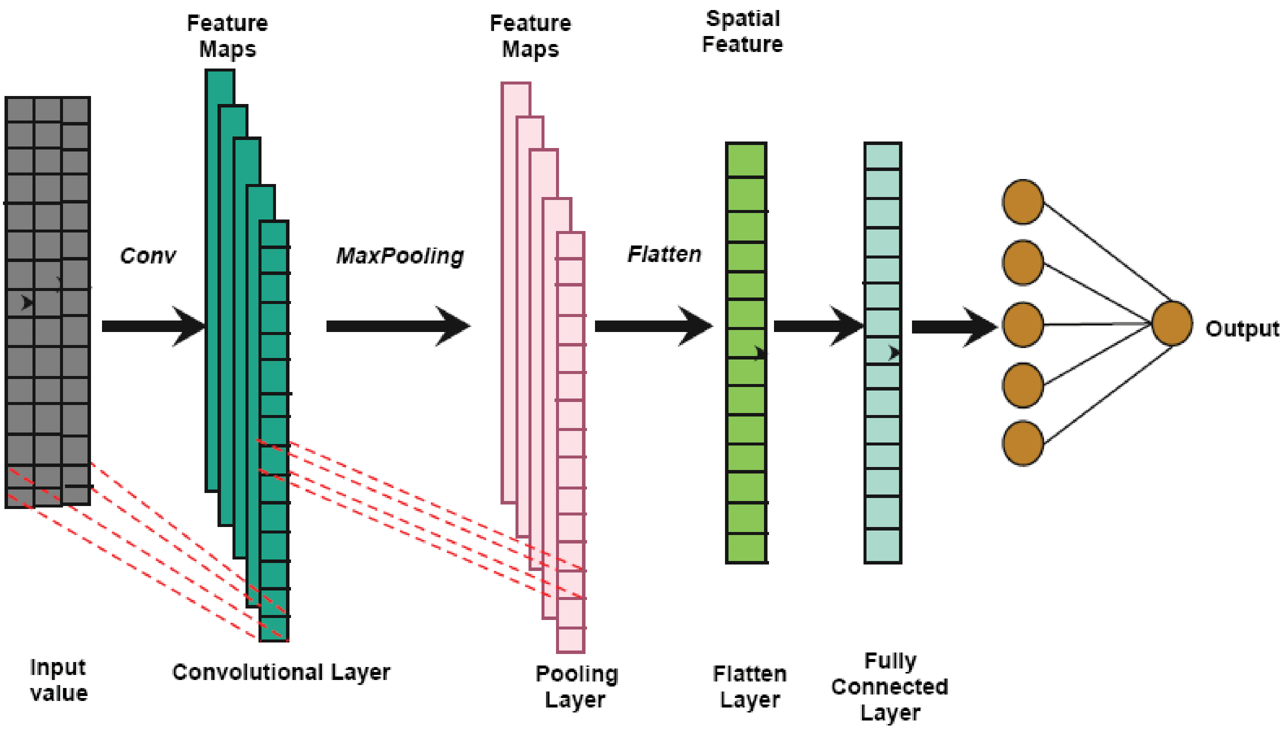

Figure 2. Convolution neural network.

meteorological features [8]. As regards the characteristics of meteorology, they include: Monthly Cloud Cover in Percent is represented by cld; The term "dtr" is for "diurnal temperature range"; "frs" stands for "frost day frequency"; "pet" stands for "potential evapotranspiration" in millimetres per day; and "pre" stands for "precipitation" in millimetres per month.

The following acronyms stand for temperature: tmn, which stands for daily mean, tmp, which stands for temperature, tmx, which stands for monthly average maximum and minimum, vap, which stands for vapour pressure, and wet, which stands for wet day frequency in days.

The geographical features are listed as follows: As with the other categories, "region" refers to the continent, "country" to the nation where the outbreak was first reported, "reporting date" to the day on which the outbreak was first reported, "elevation" to the height above sea level, and "dominant landcover" to the type of vegetation that predominated at the site in question. The X5 Ct 2010 Da file format provides a brief overview of asymmetric cattle GIS data, while the X5 Bf 2010 Da file format does the same for buffalo, and the lumpy file format classifies whether or not an animal is LSD-infected: 1, no code infection: 0.

Data Preprocessessing

There are always no mistakes in the dataset. Text, numbers, time series, continuity, and discontinuity are only a few of the data kinds that may be found in a dataset. Data quality issues such as noise, abnormalities, missing, erroneous, and duplicates may also be present. The data set is either too big or too little. For the model to be effective, the data must agree with it. To make the dataset suitable for the model, pre-processing is required. Data pre-processing includes actions like the elimination of singular features and the handling of missing information. Categorical variables were transformed into numeric values using the one-hot encoding approach so that they could be utilised by ML algorithms. Additionally, min-max scaling was used to standardise the values of several predictive features. In the end, the train test split class in the scikit-learn package was used to divide the dataset into test and training sets. Models were validated using the original training dataset and a new, independent test set.

Data Balancing

SMOTE is used to handle imbalanced data classes. SMOTE constructs synthetic examples by selecting minority class examples at random and employing interpolation techniques to build examples between the target and surrounding instances. The SMOTE method classifies data before sampling it either more or less thoroughly. These three operational elements are the neighbor count used to enter new cases, the up-sampling rate, and the down-sampling rate, respectively. The dataset is now clean and efficient for classification.

Hyperparameter tuning using Keras Tuner

The Keras Tuner library was developed as a hyperparameter tweaking tool for Keras- based deep neural networks. [33]:

- ✓ Define a model: Tuning a deep learning model requires first defining that model. A model constructed using either the sequential or functional Keras APIs will do.
- ✓ Define a hyperparameter search space: Establish a search space for the hyper- parameters you want to tune Next, you need to establish a search space for the hyperparameters you want to tune.
- ✓ Define a search algorithm: One need a search algorithm to sift through possible values for the hyperparameters and zero in on the optimal ones.
- ✓ Define a tuner: To encapsulate the model, the search space, and the search algorithm in order to execute the hyperparameter search.
- ✓ Run the hyperparameter search: Finally, one may start the tuner and look for opti- mal hyperparameter settings.

Once the hyperparameter search has been completed, the best hyperparameters may be used to train a final model using the whole training dataset [34], and then the model's performance can be assessed using the test dataset. Time and effort can be saved using Keras Tuner to automate the process of hyperparameter tweaking, which can lead to higher performance in your deep learning models [35].

Proposed Work

This section explains our proposed approach in order to classify lumpy disease in cat- tle and buffaloes. The lumpy image dataset in our research is preprocessed for the sake of removing distorted and blurred images from the dataset. Then SMOTE is used to handle imbalanced data classes. SMOTE constructs synthetic examples by selecting minority class examples at random and employing interpolation techniques to build examples between the target and surrounding instances. The SMOTE method classifies data before sampling it either more or less thoroughly. These three operational elements are the neighbor count used to enter new cases, the up-sampling rate, and the down-sampling rate, respectively. The dataset is now clean and efficient for classification.

Before classification, the various pre-trained models are utilized for extracting features from the images, these are pre-trained neural network architecture for large datasets such as ImageNet, where it learns to extract features that are useful for a wide variety of image classification tasks. The pre-trained models utilized are; MobileNetV2, VGG19, ResNet50, InceptionV3, and DenseNet-121.

The proposed work has taken the pre-trained network and removed the final classification layers. This leaves us with a network that can extract features from images but does not classify them. Then we used these features as input to a new classification network.

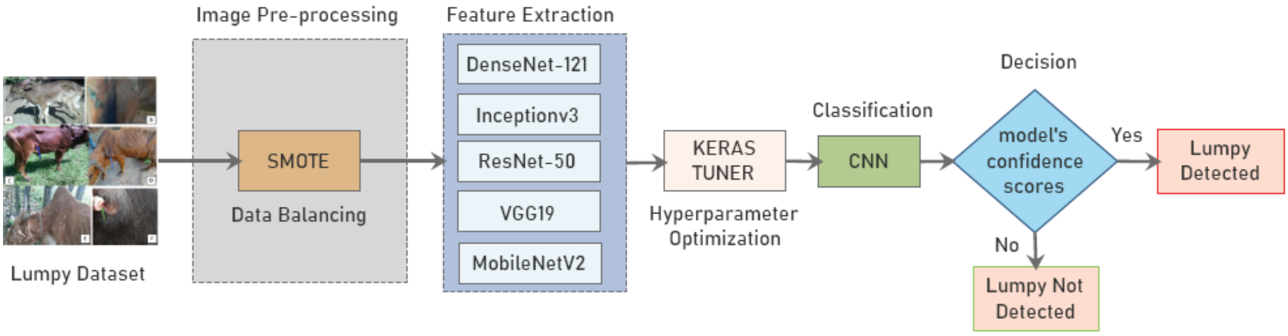

Figure 3. Proposed model.

The classification is performed using a two-dimensional Convolution Neural Net- work with the combination of max pooling layers, flatten layers, and dense layers, arranged in a fashion as visible in Figure 2. The parameters of the CNN are tuned using a Keras tuner which automates the process of hyper-parameter tweaking, which can lead to higher performance in the deep-learning classification models.

Finally, the dataset is trained with 70% of the lumpy image dataset and tested with the remaining 30% dataset. The experiment is performed using Python language, on jupyter notebook in Anaconda framework. The system with high processing with 32Gb RAM is used for the experiment purpose. In order to construct and train the CNN model, we must first load the required libraries, one of which is TensorFlow. We also import ImageDataGenerator, which is used to generate batches of augmented image data for training.

Next, we set up the directories that will hold our test and training images. Next, we create an image generator that uses data augmentation to have our training data ready to go. We imported keras and its keras tuner library to tune the parameter of the classification CNN model. Figure 3 provides the overall flow of the proposed model.

After successful execution (training and testing of model) of the models, the results are evaluated and analysed using confusion metric parameters; Accuracy, Precision, Recall, and F1-score. The graphical analysis is also performed using inbuilt Python libraries in order to generate ROC and Convergence curves, which are later discussed in the next section.

PERFORMANCE METRICS

The research employs the confusion matrix parameters (accuracy, recall, precision, and F1 score) to assess the effectiveness of the proposed model. The following details the confusion matrix [34] terms: To understand the evaluation criteria, one should be familiar with the following four terms:

- True Positive (TP): The number of occurrences in which the model correctly classifies them as positive.
- True Negative (TN): The number of occurrences in which the model correctly classifies them as negative.
- False Positive (FP): The number of negative occurrences in which the model incorrectly classifies them as positive.
- False Negative (FN): The number of positive occurrences that the model incorrectly classifies as negative.

Accuracy: It is the ratio of the number of correct predictions to the total num- ber of predictions made by the model. [36].

$$\text{Accuracy} = (TP + TN)/\ (TP + TN + FP + FN \quad (1)$$

Precision: It is the ratio of true positives to the total number of positive predic- tions made by the model. [37].

$$\text{Precision} = TP/\ (TP + FP)(2)$$

Recall: It is the ratio of true positives to the total number of actual positive cases in the dataset. [16].

$$\text{Recall} = TP/(TP + FN) \quad (3)$$

F1-score: The F1-score balances accuracy and recall into a single number. It's helpful when there's an inequity in the dataset since it strikes a balance between accuracy and recall [38].

$$F\,1 = 2 * (\text{precision} * \text{recall})/(\text{precision} + \text{recall}) \quad (4)$$

RESULTS AND DISCUSSION

In this work, MobileNetV2 was used with a CNN classifier as a feature extractor to diagnose LSD. There are a total of 14 data characteristics and 24,803 data points included in the forecasts. This dataset had 3,039 infected records from set (1) and 21,764 from set (0). Resampling of the dataset is required since it is clear that the current data does not fairly represent cattles who are infected with lumpy compared to those who are not. Before resampling the data, the attributes such as, 'country', 'region', 'X5 Ct 2010 Da', 'X5 Bf 2010

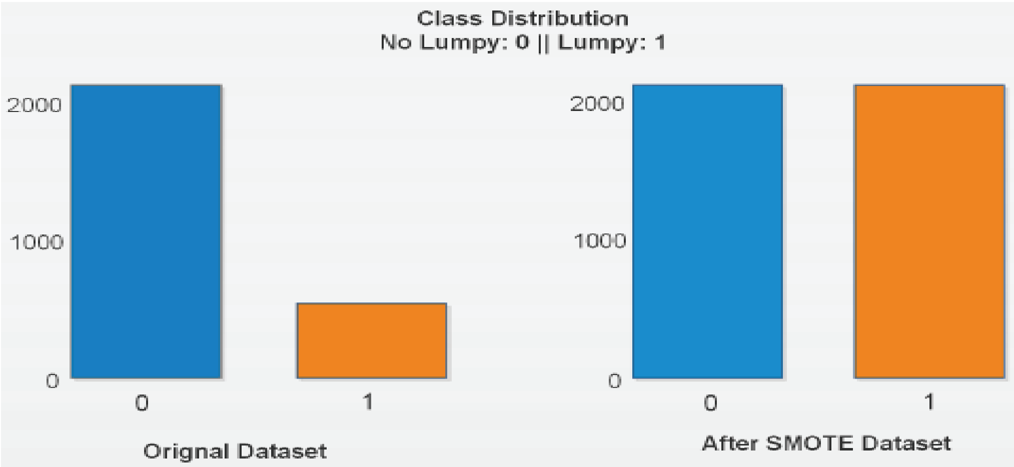

Figure 4. Dataset balancing using SMOTE.

Da', and 'reportingDate', will be eliminated, as they are not used in the classification process. The association between attributes was also analysed using a correlation heatmap in an effort to reduce the size of strongly linked features.

Some benefits of using MobileNetv2 for feature extraction are; first, since the network is pre-trained, we can take advantage of the knowledge learned from the original dataset to improve the accuracy of our new classification model. Second, by using depth-wise separable convolutions, MobileNetV2 is computationally efficient and requires fewer resources than other, more complex networks.

The original dataset should be used to run the training models, and then the LSD dataset should be resampled to evaluate the first model's efficacy. In order to put the model to the test, we must first partition the original data frame into a 70:30 split of training and validation data.

After employing the SMOTE fit resample oversampling strategy, the resampled dataset shape counter (1: 21.764, 0: 21.764), indicates that both the infected and uninfected datasets share the same total number of observations (21.764). Figure 4 depicts the balanced distribution of data that results from SMOTE re-sampling.

There are hyperparameters that need to be specified when using CNN for classification, including learning rate, batch size, filter size, pooling type, pool size, dropout rate, activation function, optimizer, and others. The optimal hyperparameters are optimized using Keras Tuner, which examines all possible combinations within the provided setup. learning rate: [0.0001, 0.001, 0.01, 0.1], Batch size: [16, 32, 64, 128], filter size: [3x3, 5x5,7x7], activation function: [ReLU, LeakyReLU, ELU, Sigmoid, Tanh], pooling type: [max pooling, average pooling], pooling size: [2x2, 3x3], dropout rate: [0.1, 0.2, 0.3, 0.4, 0.5], optimizer: [adam, RMSprop, SGD] are used in this study. The most ideal settings based on CNN are determined after hyperparameter tweaking, and they are as follows: learning rate: 0.001, batch size: 32, filter size: 3x3, pooling type: max pooling, pool size: 3x3, dropout rate: 0.5, activation function: ReLU, optimizer: adam.

The accuracy, precision, sensitivity, and F1-Score of the model are assessed using the Confusion Matrix. based on the test scenario utilizing MobileNetV2 and SMOTE with CNN classification on LSD datasets. The experiment's findings are shown in Table 2. The graphical comparison of the evaluation metrics is illustrated in Figure 5.

Receiver Operating Curve (ROC)

The confusion metric is used to assess the receiver operating characteristic (ROC) curve and to identify whether or not the identification is successful. The ROC curve establishes the ratio of true positives to false positives in

Table 2. Performance of various feature extraction models with CNN

Model	Accuracy(%)	Precision(%)	Sensitivity(%)	F1-Score(%)
DenseNet121-CNN	94.25	93.11	93.05	94.08
InceptionV3-CNN	93.29	91.85	91.17	92.88
ResNet50-CNN	94.74	93.66	92.87	93.21
VGG19-CNN	94.44	93.22	92.87	92.45
MobileNetV2-CNN	99.88	98.43	98.67	99.32

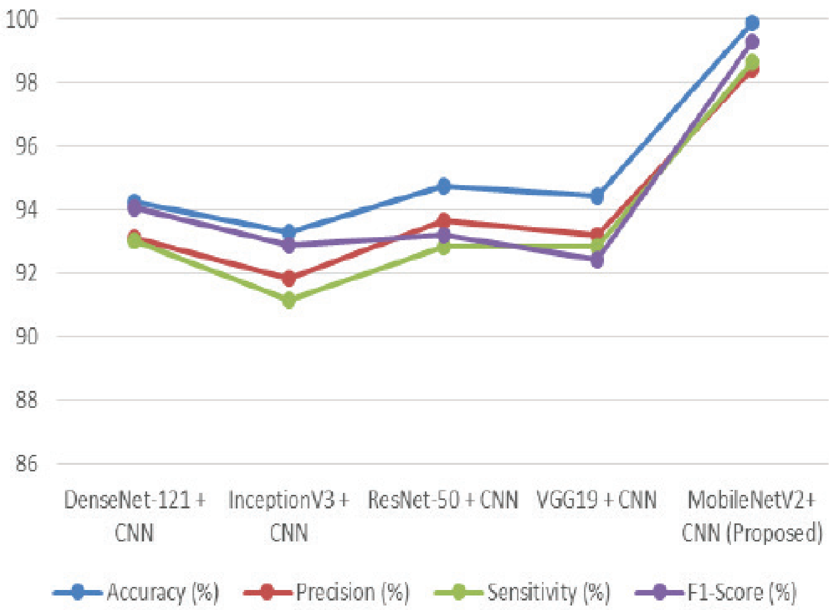

Figure 5. Graphical representation of model performance.

experimental data. True positive rate (Y axis) and false positive rate (X axis) are plotted on the ROC curve. An ideal ROC (receiver operating characteristic) curve climbs sharply toward the top-left corner, signifying that the model has a high true positive rate and a low false positive rate. The closer the curve gets to this corner, the more effectively the model differentiates between positive and negative classes.

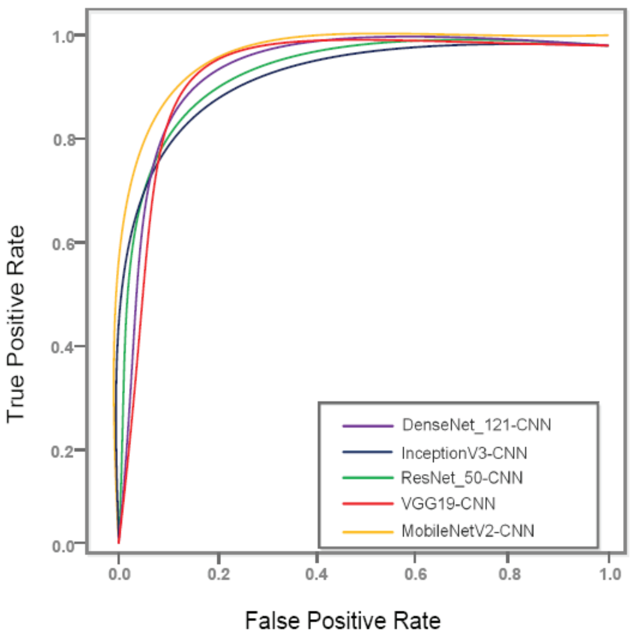

Figure 6. ROC of Various Feature Extraction techniques with CNN.

The Area Under the Curve (AUC) is a crucial indicator of performance, with values near 1 indicating excellent classification ability. In contrast, a random or weak model will produce a curve closer to the diagonal, where the AUC approaches 0.5. For optimal performance, the ROC curve should have an AUC near 1, reflecting strong predictive accuracy. Figure 6 depicts how the proposed model ROC stacks up against its competitors. The area under the ROC curve (AUC) is a two-dimensional measure of this connection that shows how well the suggested model outperforms alternative models.

Convergence Curve

While training and testing a model, the optimal value of the learning parameter is depicted by a convergence curve that plots the accuracy of the model against the loss function. The Figure 7 depicts the convergence curves of the proposed and alternative models. The sub-figures represent 7a DenseNet121, 7b InceptionV3, 7c ResNet50, 7d VGG19, and 7e proposed approach (MobileNetV2), all with the CNN model respectively. The 200-epoch training and validation accuracy curves for the proposed models are displayed. The proposed model outperforms the state-of-the-art with a training and validation accuracy of 99.88%, revealing a faster convergence rate.

Comparison with existing work

Table 3 gives a state-of-art analysis based on the performance of existing classification models and the proposed models on considering accuracy measures. The paper taken into account are using different classification techniques some of them are artificial neural networks, convolution

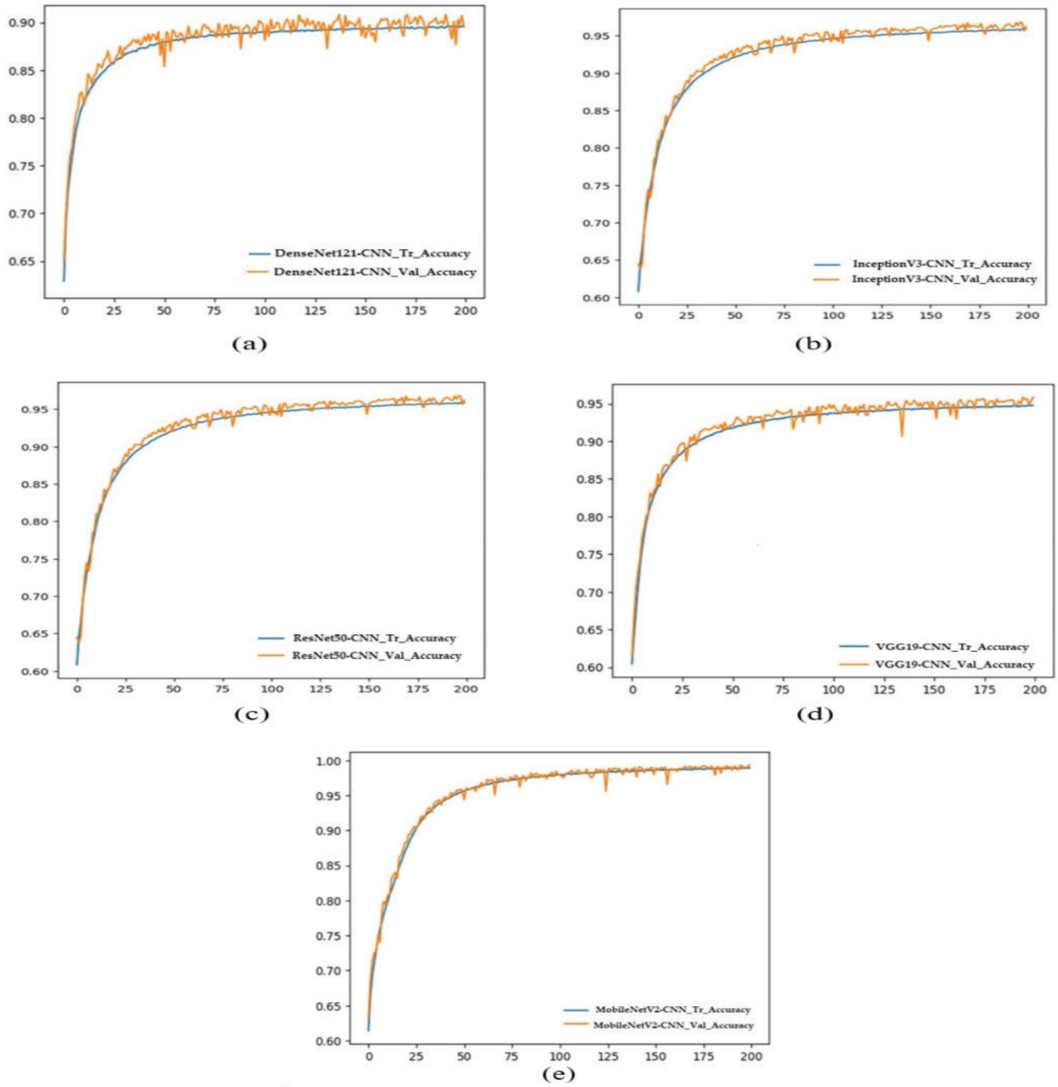

Figure 7. Convergence curve analysis: (a)DenseNet121, (b)InceptionV3 (c) ResNet50 (d)VGG19 (e) Proposed model.

Table 3. The state-of-art comparison of the proposed model with existing work

Ref	Dataset	Model	Accuracy (%)
[8]	Lumpy skin disease outbreak data	ANN	97
[10]	Debre Markos University, Addis Ababa University-School of Veterinary Medicine, the World Wide Web, and other secondary sources were used for photographic holdings.	CNN	95
[11]	LSD dataset with 18,603 instances and 16 features.	Random Forest	97.7
(12]	Dataset of LSD.	ELM	90.06
[14]	Dataset created by the authors.	Neural network model using Inception-v3 for feature extraction	92.5
[15]	Pictures from four health care facilities in the Sunyani Municipality of Ghana.	Medilab-plus CNN classifier	88 (atopic dermatitis), 85 (acne vulgaris), 84.7 (scabies)
Our Work	LSD Dataset	SMOTE+MobileNetV2+CNN	99.88

neural networks, ensemble learning models and individual models such as random forest.

The broader implications of our findings extend beyond the immediate scientific community, as the detection model can be implemented in various agricultural sectors, particularly in regions where LSD outbreaks are prevalent, such as Africa, the Middle East, and Asia. By incorporating this technology, veterinary services, livestock producers, and governmental agencies can utilize automated tools to rapidly detect and respond to disease outbreaks, improving animal welfare and reducing the impact of the disease on food security and trade. This work can also inspire further research into applying deep learning models for other transboundary animal diseases, enhancing overall disease control strategies.

CONCLUSION

Lumpy skin disease which is widespread in cattle, water buffalo and camels in South Asian countries, may be identified by the appearance of lumps or nodules on the animals' skin which can cause fever, anorexia, and reduced milk production. Lumpy skin disease has a considerable impact on economies in several countries. In order to detect and control the spread of the disease among animals, there are various measures taken by the government of various countries. The paper proposed a MobileNetv2- based Deep Learning model for the classification of lumpy skin disease (LSD) by training the machine using a lumpy dataset. The images in the dataset are preprocessed and then features are extracted using MobileNetv2, which is a pre-trained neural network model. The classification is performed using a 2D-layered convolutional neural network model with a combination of convolutional layer, flatten layers, max-pool layer and dense layer, which obtains the highest accuracy of 99.88%. The proposed model outperformed other individual models and existing work by different researchers in the domain.

In future, we intend to apply federation and transfer learning techniques in order to pre-process affected animal images and classify disease more accurately. We will also use the proposed technique on other image datasets in order to create a more generalized and robust model.

DECLARATIONS ETHICAL APPROVAL

Not Applicable

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest

FUNDING

Not applicable

DATA AVAILABILITY STATEMENT

The data used in this study will be made available upon reasonable request.

REFERENCES

[1] AAU. Lumpy skin disease (lsd): outbreak investigation, isolation and molec- ular detection of lumpy skin disease in selected areas of eastern shewa, ethiopia. Available at: https://etd.aau.edu.et/bitstreams/57dd46f6-bb31-48c2-a563-3760428b9fe3/download. Accessed on 4 Apr 2026.

[2] Al-Salihi K. Lumpy skin disease: Review of literature. Mirror Res Vet Sci Anim 2014;3:6–23.

[3] Kumar R, Godara B, Chander Y, Kachhawa JP, Dedar RK, Verma A, et al. Evidence of lumpy skin disease virus infection in camels. Acta Trop 2023;242:106922. [CrossRef]

[4] Sprygin A, Pestova Y, Wallace D, Tuppurainen E, Kononov A. Transmis- sion of lumpy skin disease virus: A short review. Virus Res 2019;269:197637. [CrossRef]

[5] Chaudhary N. A review on lumpy skin disease in india. Available at: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4351704. Accessed on 4 Apr 2026.

[6] Mathivanan E, Raju K, Rajaganapathy. Outbreak of lumpy skin disease in india 2022-an emerging threat to livestock & livelihoods. Global Biosecurity 2023.

[7] Gupta T, Patial V, Bali D, Angaria S, Sharma M, Chahota R. A review: Lumpy skin disease and its emergence in india. Vet Res Commun 2020;44:111–118. [CrossRef]

[8] Safavi EA. Assessing machine learning techniques in forecasting lumpy skin disease occurrence based on meteorological and geospatial features. Trop Anim Health Prod 2022;54:55. [CrossRef]

[9] Farra D, Nardi M, Lets V, Holopura S, Klymenok O, Stephan R, et al. Qualitative assessment of the probability of introduction and onward transmission of lumpy skin disease in ukraine. Microb Risk Anal 2022;20:100200. [CrossRef]

[10] Lake B, Getahun F, Teshome FT. Application of artificial intelligence algo- rithm in image processing for cattle disease diagnosis. J Intell Learn Syst Appl 2022;14:71–88. [CrossRef]

[11] Ujjwal N, Singh A, Jain AK, Tiwari RG. Exploiting machine learning for lumpy skin disease occurrence detection. IEEE Xplore 2022;1–6. [CrossRef]

[12] Genemo M. Detecting high-risk area for lumpy skin disease in cattle using deep learning feature. Adv Artif Intell Res 2023;3:27–35. [CrossRef]

[13] Utami E, Muhammad AH, Suparyati. Lumpy skin disease prediction based on meteorological and geospatial features using random forest algorithm with hyper- parameter tuning. IEEE Xplore 2022;99–104. [CrossRef]

[14] Rai G, Hussain A, Kumar A, Ansari A, Khanduja N. A deep learning approach to detect lumpy skin disease in cows. Springer 2021;369–377. [CrossRef]

[15] Akyeramfo-Sam S, Philip AA, Yeboah D, Nartey NC, Nti IK. A web- based skin disease diagnosis using convolutional neural networks. Int J Inf Technol Comput Sci 2019;11:54–60. [CrossRef]

[16] Hegde PR, Shenoy MM, Shekar B. Comparison of machine learning algo- rithms for skin disease classification using color and texture features. IEEE Xplore 2018;1825–1828. [CrossRef]

[17] Fern´andez A, Garcia S, Herrera F, Chawla NV. Smote for learning from imbalanced data: progress and challenges, marking the 15-year anniversary. J Artif Intell Res 2018;61:863–905. [CrossRef]

[18] Bhattacharya A. How to use smote for dealing with imbalanced image dataset for solving classification problems. Available at: https://medium.com/swlh/how-to-use-smote-for-dealing-with-imbalanced-image-dataset-for-solving-classification-problems-3aba7d2b9cad. Accessed on 4 Apr 2026.

[19] Krishna ST, Kalluri HK. Deep learning and transfer learning approaches for image classification. International J Recent Technol Eng 2019;7:427–432.

[20] Zhuang F, Qi Z, Duan K, Xi D, Zhu Y, Zhu H, et al. A comprehensive survey on transfer learning. IEEE Xplore 2020;109:43– 76. [CrossRef]

[21] Huang G, Liu Z, Weinberger KQ. Densely connected convolutional networks. arXiv 2016.

[22] Xu Y, Liu X, Cao X, Huang C, Liu E, Qian S, et al. Artificial intelligence: A powerful paradigm for scientific research. Innov 2021;2. [CrossRef]

[23] Pang S, Zhang Y, Ding M, Wang X, Xie X. A deep model for lung cancer type identification by densely connected convolutional networks and adaptive boosting. IEEE Xplore 2019;8:4799–4805. [CrossRef]

[24] Szegedy C, Vanhoucke V, Ioffe S, Shlens J, Wojna Z. Rethinking the inception architecture for computer vision. arXiv 2015. [CrossRef]

[25] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. arXiv 2015.

[26] Darth M. Vgg 19 feature extraction. Available at: https://www.kaggle.com/code/omkarmodi/vgg-19-feature-extraction. Accessed on4 Apr 2026.

[27] Rajinikanth V, Raj ANJ, Thanaraj KP, Naik GR. A customized vgg19 network with concatenation of deep and handcrafted features for brain tumor detection. Appl Sci 2020;10:3429. [CrossRef]

[28] Sandler M, Howard AG, Zhu M, Zhmoginov A, Chen L. Inverted resid- uals and linear bottlenecks: Mobile networks for classification, detection and segmentation. arXiv 2018.

[29] Gu J, Wang Z, Kuen J, Ma L, Shahroudy A, Shuai B, et al. Recent advances in convolutional neural networks. Pattern Recognit 2018;77:354–377. [CrossRef]

[30] Li Z, Liu F, Yang W, Peng S, Zhou J. A survey of convolutional neu- ral networks: analysis, applications, and prospects. IEEE Xplore 2021. [CrossRef]

[31] O'Shea K, Nash R. An introduction to convolutional neural networks. arXiv 2015.

[32] Pon MZA, Prakash K. Hyperparameter tuning of deep learning models in keras. Available at: https://sparklinglightpublisher.com/index.php/staiqc/article/view/5/4. Accessed on 4 Apr 2026.

[33] Abdellaoui B, Moumen A, El Idrissi YEB, Remaida A. Training the fer2013 dataset with keras tuner. Available at: https://www.scitepress.org/Link.aspx?doi=10.5220/0010735600003101. Accessed on 4 Apr 2026.

[34] Shawki N, Nunez RR, Obeid I, Picone J. On automating hyperparameter optimization for deep learning applications. IEEE Xplore 2021. [CrossRef]

[35] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. J Mach Learn Res 2011;12:2825–2830.

[36] Zhan J, Huang X, Qian Y, Ding W. A fuzzy C-means clustering-based hybrid multivariate time series prediction framework with feature selection. IEEE Xplore 2024. [CrossRef]

[37] Zhu C, Ma X, D'Urso P, Qian Y, Ding W, Zhan J. Long-term multivariate time series forecasting model based on Gaussian fuzzy information granules. IEEE Xplore 2024. [CrossRef]

[38] Zhu C, Ma X, Ding W, Zhan J. Long-term time series forecasting with multi-linear trend fuzzy information granules for LSTM in a periodic framework. IEEE Xplore 2023. [CrossRef]