International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

DIAGNOSIS OF LUMPY SKIN DISEASE IN DAIRY COWS USING ML AND DEEP LEARNING.

[1] *Prajul Mittal*

[1] Third year student of Department of Information Technology, Maharaja Agrasen Institute of Technology, Rohini

ABSTRACT :

Lumpy skin disease is a transferable viral disease that affects cattle leading directly to huge economic losses to in poultry and dairy industries. A reduction in milk production, growth retardation, reduced fertility and early age death are some of the signs of Lumpy skin disease. Early and accurate diagnosis of lumpy skin disease is very helpful for the minimizing the impacts and controlling the outbreaks. In this research we proposed an automated deep learning based system for the diagnosis of lumpy skin disease using analysis of images. By using InceptionV3 convolutional neural network architecture we developed a binary classification model that can classify the images as healthy cows and LSD affected cows.

Keywords: computer vision ,segmentation ,CNN ,MobileNetV2 ,LSD

INTRODUCTION

Skin is the largest gland in the body of a cow. Essentially it separates the cow's inner organs form its external surrounding. It helps in regulating the temperature of the body by sinking it with the external environment temperature. Skin helps in maintaining dietary fiber, metabolism, omega3, omega6, collagen etc. The major symptoms of lumpy skin disease in cattle include high fever, swollen lymph nodes, reduced milk production, reluctance to move and eat, discharge of fluid from eyes etc. Commonly speaking Africa, Russia, Africa, Oman, and India are some of the countries that are home to lumpy skin disease. In earlier times it was well known in Egypt.

LITERATURE SURVEY

METHODOLOGY AND PROPOSED SYSTEM

The following section shows and depicts the proposed methods and approaches for the machine vision based approach for the detection of lumpy skin disease in cattle. It involves series of steps including dataset collection and designing, exploring the data, data preprocessing and normalization, data engineering, model training and validation. Fig. 1 portrays a high level flow chart of proposed methodology.

3.1 DATASET

The first step in making a machine learning and deep learning model is choosing an appropriate dataset. Generally, the data for diagnosis of lumpy skin disease is present in the form of csv file and images. The dataset may contain categorical columns as well as labelled columns.
When learning is done on the basis of labelled data then it is known as **supervised learning** while unsupervised learning does not require any labelled columns. At any given instance our dataset (csv file) consists of x, y, region, country ,reporting date , etc.The dataset is downloaded from a website named Kaggle.

When the learning/the model is trained on the basis of the classful images the the learning is called deep learning. For training the deep learning models our we have a different dataset. It consist of images belonging to two classes namely healty cows and lumpy cows. It is also downloaded from a website named Kaggle.

Fig.3 Dataset class samples: (a) Healthy cow images (b) Lumpy cow images

3.2 IMAGE DATA PREPROCESSING

Preprocessing an image means transforming a given image into that form that can be easily understood by the machine and the machine can efficiently use it. To achieve a better performance for the diagnosis of the lumpy skin disease image preprocessing is crucially important because the image of the cattle may vary according the environment, surface foundation and the angle of the photograph. There are around 900 images in our dataset and not all the images are sharply focused on the cows. Preprocessing of the images helps the machine in accurate recognition of the cattle by understanding the patterns and removing the noises(unwanted features). In our methodology we have used the following steps to preprocess the image.

a. Rescaling the image (Normalizing the pixels in range 0-1).
b. Setting up a particular rotation range.
c. Setting up height and width of shifting range.
d. Setting up zoom and shear range.
e. Setting up horizontal flip.

3.3 DATA SEGMENTATION

In our approach the data segmentation is based on the directory of our dataset that organizes the images into sub folders that represents different classes. Each subfolder contains an image belonging to a different class. The ImageDataGenerator( ) method in tensorflow when invoked labels the images in according to the different classes based on their folder names (*healthy and lumpy cows*).

Here's an in-depth explanation on how data segmentation takes place in our deep learning models :-

1. ***Directory Structure-Based Segmentation*** :- Our training data consists of two sub folders namely healthy cows and lumpy cows. We have further divided the dataset in two parts namely training data and validation data in ratio 80% : 20%.
2. ***Using ImageDataGenerator for Automated Segmentation***:- When we specify the class_mode as 'binary', ImageDataGenerator.flow_from_directory function interprets each label and assigns distinct class automatically. It segments data into two classes namely healthy cows and lumpy cows.
3. ***Automated Labelling*** :- The ImageDataGenerator automatically labels based on directory names:

Healthy cows : 0

Lumpy cows : 1

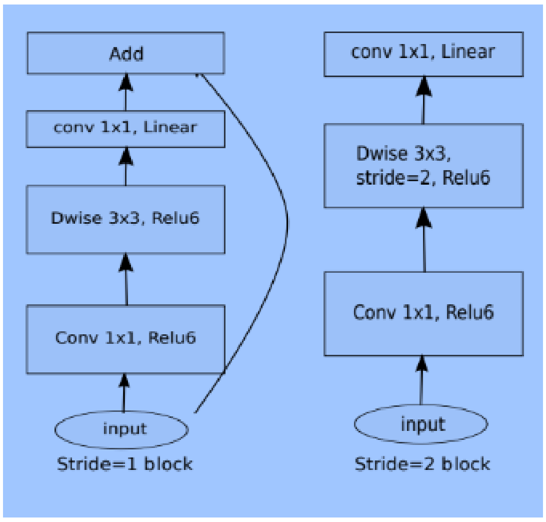
3.4 FEATURE EXTRACTION

DL M E T H O D S

After image preprocessing and segmentation, we set up our deep learning models . In our approach we have implemented four pre trained deep learning models namely *MobileNetV2, DenseNet121, Xception and InceptionV3*. We have trained all these models on our dataset and shall try to determine which is the best preforming model and which model can be highly accurate in prediction of lumpy skin disease.

3.6.1 MobileNetV2

MobileNetV2 is a lightweight classification model that was developed by google. It consists of 53 lightweight convolution layers that are helpful in image visioning in devices like smart phones. In various devices like mobile phones, it provides the feature of real time classifaction and capabilities under several computing constraints. Lightweight convolution layers are used by MobileNetV2 model to transfer features in advancement layer. The following image helps in clear understanding of this model.



EXPRIMENTAL EVALUATION AND RESULTS

3.1. E N V I R O N M E N T S P E C I F I C A T I O N S

GPUs (Graphics Processing Units) offer the processing power essential for tasks like image analysis and classification. However, setting up a GPU requires additional hardware and is often costly. To address this, we use the Google Colab platform to train our model, leveraging the powerful cloud GPUs it provides. Colab comes preloaded with the necessary storage and libraries, so we don't need to add any further installations. It includes a v3 TPU chip equipped with two TensorCores, capable of delivering 275 teraflops, along with 25 GB of disk space. These resources make it feasible to train deep learning models in a high-performance computing environment.

3.2. P E R F O R M A N C E A N D E V A L U A T I O N M E T R I C S

This area focuses on evaluating the effectiveness of experiments conducted within this system. Python served as the primary language for the proposed model's experiments in Google Colab, utilizing libraries for deep neural networks and machine learning, including Keras, TensorFlow, NumPy, pandas, and Scikit-Learn. For diagnosing lumpy skin disease in cows, models such as MobileNetV2, DenseNet121, Xception, and InceptionV3 were employed.

To predict classes as true or false, a classifier is utilized. The results from classifying data related to different categories can fall into one of four types. If the prediction aligns with reality, it is classified as either True Positive (TP) or True Negative (TN), indicating correct predictions. Alternatively, if the prediction differs from reality, it is categorized as either False Positive (FP) or False Negative (FN). By calculating additional metrics from the confusion matrix, we can further evaluate the classification accuracy and performance of our models.