VISVESVARAYA TECHNOLOGICAL UNIVERSITY, BELGAUM, KARNATAKA

PROJECT REPORT

ON

"LUMPY SKIN DISEASE DETECTION"

Submitted in partial fulfillment of the requirement for the award of the degree of

BACHELOR OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

Submitted by

USN	NAME
2SD20CS016	Anirudh Gudisagar
2SD20CS031	Bharatesh Nagaraj Labhagond
2SD20CS033	Bhoomika Wali
2SD20CS113	Sushmita Patil

Under the Guidance of

Prof. Yashodha Sambrani
Dept. of CSE, SDMCET, Dharwad

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
S.D.M. COLLEGE OF ENGINEERING & TECHNOLOGY,
DHARWAD-580002

2022-2023

S.D.M COLLEGE OF ENGINEERING & TECHNOLOGY,

DHARWAD –580002

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

Certified that the Minor-Project-1 work and presentation entitled **"LUMPY SKIN DISEASE DETECTION"** *is a bonafide work carried out by* **Anirudh Gudisagar (2SD20CS031)**, **Bharatesh Nagaraj Labhagond (2SD20CS031)**, **Bhoomika Wali (2SD20CS033)**, and **Sushmita Patil (2SD20CS113),** *students of S.D.M. College of Engineering & Technology, Dharwad, in partial fulfillment for the award of* **Bachelor of Engineering in Computer Science and Engineering of** ***Visvesvaraya Technological University***, ***Belgaum***, *during the year 2022-2023. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The Minor-Project-1 has been approved, as it satisfies the academic requirements in respect of project report prescribed for the said degree.*

----------------------------- ----------------------

Prof. Yashodha Sambrani — **Dr. S.M. Joshi**

Project Guide — HOD-CSE

ABSTRACT

Lumpy Skin Disease is known as a major risk to cattle production and substantial impacts on livelihoods and food security especially for our country. Currently, detection of Lumpy Skin Disease in our country is assessed manually[4]. However, manual evaluation takes significant amount of time and requires trained professional and experienced person. Therefore, technology is needed to prevent animal disease epidemics. Automated detection of Animal Lumpy Skin Disease has advantages over the manual technique. Detection of Lumpy Skin Disease in Cows is developed in literature. But Animal Lumpy skin disease has different classification based on its severity. There is a need to further identify the different stages of Lumpy skin disease to know to what extent the animal is affected by lumpy skin disease. In this study, Lumpy skin disease detection model is constructed using Convolutional Neural Network (CNN) for feature extraction and SVM for classification. CNN is the state of the art for deep feature extraction, hence we used it for feature extraction. The model used to detect and classify animal Lumpy Skin Disease skin diseases into Healthy and Lumpy. The dataset is collected from Livestock production offices and from internet external images repository. After collecting data, Image augmentation, Image Preprocessing, and Image Segmentation techniques are applied to enhance image quality and identify region of interest.

It is a challenging task for properly detecting the region of interest and hence we recommend as a way forward to use advanced image processing techniques to improve image quality for proper segmentation and Lumpy skin disease detection.

Keywords: *Animal Lumpy Skin Disease, convolutional neural network, Image Processing .*

Table of Contents

PROBLEM STATEMENT

In India Lumpy skin disease has caused a major outbreak in the year 2022 resulting in the death of over 97,000 cattle in three months between July and September affecting more than 15 states across the country. Manual evaluation takes significant amount of time and require trained professional and which is time consuming, hence an automated machine vision expert is required to address the issue at the earliest.

CHAPTER 1: INTRODUCTION

Artificial Intelligence is among the next big things in the software engineering field that empowers numerous applications in health care for good measure. Several scientists have embarked on utilizing the advancement of Artificial Intelligence and high-grade dataset to change our lives profoundly. One notable field in Artificial Intelligence is Computer Vision, which rapidly emerged over the last decade thank to an enormous amount of visual data and significant development in GPU processing power. Neural networks can now have millions of trainable parameters which make technologies like diseases self-diagnoses.

However, Computer Vision problems require real-time prediction and the AI community calls out for a client-side deep learning system. On 18th March 2018, Google released Tensorflow.js, an open-source library written in the JavaScript programming language for creating in-browser Artificial Intelligence platform. Tensorflow.js provides an environment for AI enthusiasts to train, run and deploy their AI programs in their browsers. Real-time AI has unlocked many opportunities and possibilities for AI projects, like interactive Machine Learning programs, data privacy, app sharing and decentralized computing resources

The objectives of this to demonstrate the usability and effectiveness of in browser Artificial Intelligence system. This is to achieve that goal by constructing a neural network model to classify healthy and lumpy images and convert such model to JSON format for in- browser usage. The model will take a 28x28 format image as input and return the corresponding category.

CHAPTER 2: LITERATURE SURVEY

Artificial Intelligence is a vast field which concerns several areas like Statistic, Optimization, Machine Learning and Deep Learning. This section introduces DL as the state-of-the-art solution for AI problems with complex datasets like visual data, sound or natural languages and how it's related other fields of AI. Besides, this section explains the concepts of Convolution Neural Network, optimization process of Neural Network training and then dives deeper to Mathematics theory behind their implementation.

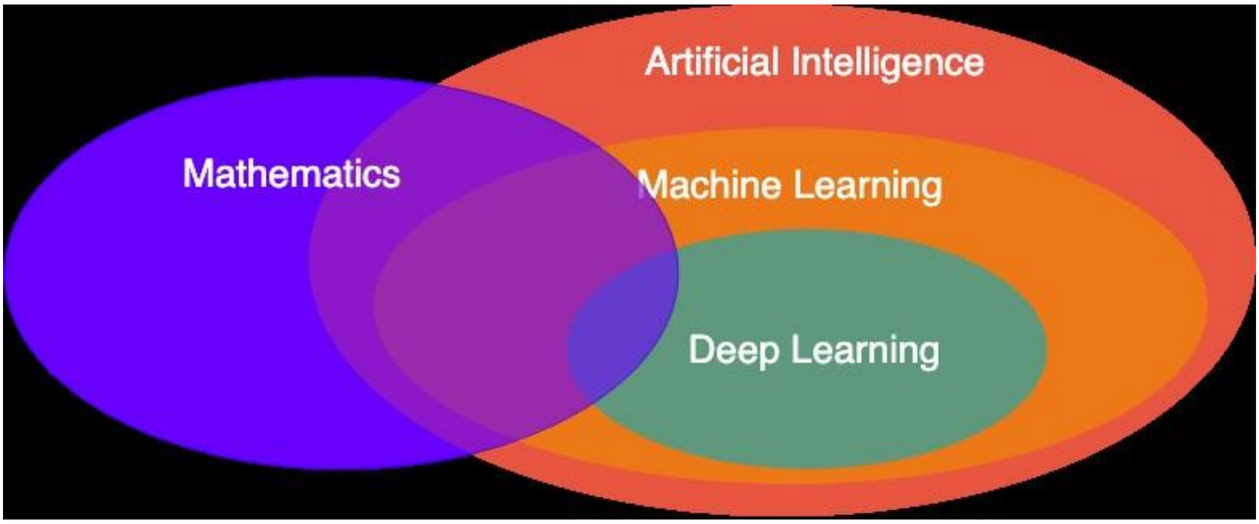


Figure 1. Relationships between Math, AI, ML and DL.

Figure 1 shows that AI, ML, and DL all involve Mathematic to some extends. Machine Learning and Deep Learning are two sub-fields of Artificial Intelligence. Machine Learning focuses on small and medium datasets, while Deep learning applies mathematical models to learn large and complex ones.

Summary of surveyed works

Present a one-class classifier based on extreme learning machine (ELM), in which the hidden layer of the neural network does not need tuning, and the output weights are computed analytically leading to relatively faster learning time. They compare their proposed method with the auto encoder neural networks, with a reconstruction method for building a one-class classifier. An outlier detection analysis in the context of seven UCI data sets and three artificially generated data sets is conducted. While both random feature mappings and kernels can be used for the proposed classifier, the latter yields better results than the former. The primary comparative study between their ELM-based model and the auto encoder neural network suggested the former has an analytical solution that obtains better generalization.

performance and that too at relatively faster network learning times. A downside of this study is that the data sets investigated in the study are relatively small in size, leaving the research gap of how the proposed method would scale up to much larger data sets especially since neural networks are notorious for relatively slow learning. How would the authors' ELM-based approach work effectively with big data?

Examine outlier detection with one-class support vector machines in the context of detecting abnormal cases of melanoma prognosis. The one-class classificationaims to model the distribution of melanoma patients who have not obtained metastases status, which in this context is the normal class (case) for patients with melanoma. The case study data was obtained from the Department of Dermatology of the Medical University of Vienna. The post-cleansing data set consisted of 270 serologic blood tests, including those from 37 patients with metastatic disease and 233 patients without metastatic disease. The one-class SVM approach was compared with regular two-class SVM and Artificial Neural Networks (ANN) algorithms. These were investigated using the WEKA data mining tool. Their empirical work suggests that one-class SVMs are a good alternative to standard classification algorithms in the case where there are only few cases available from the class of interest, i.e., patients with metastatic disease in this context. The one-class SVM models performed better than the two-class models when the latter used less than half the number of cases in the minority class. A potential problem with this study is the very small data set size of the case study, and whether their approach is scalable to larger data sets, such as big data.

2.1.2 Convolution Neural Network

Information about this we took reference from [3]. Over fitting usually occurs when large-size image datasets are fed to the training Deep Feed forward Neural Network process. For AI problem that use high-resolution images as input, the number of trainable parameters of an ANN can quickly reach over 10 million. In order for a FCNN to process a greyscale picture of dimension 1920x1080, its first layer would require at least 2,073,600 parameters. The number of computing nodes exponentially increases as the size of input grows and significantly slows down the training process. To solve this issue, Convolutional Neural Network introduces a concept called **kernel** (also called **filter**). The mechanism of kernel is based on the convolution operation in Mathematics. The convolution operation allows CNN to capture sub-feature of the inputs; therefore, the output of CNN remains unchanged despite small transformations of the inputs, such as flipping and shifting.

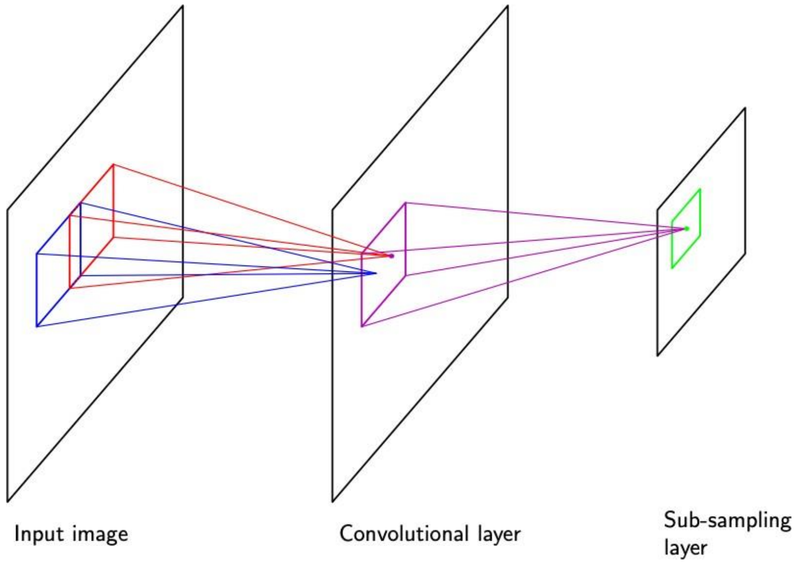


Figure 2. CNN filtering

The figure above illustrates the process of capturing sub-feature in Convolutional Neural Network using convolution operation. Each sub region of the image is filtered, and the most notable feature of that sub region is extracted and passed to the next layer of the network. The convolution operation following the below formula, with I and K are two dimensional input and kernel respectively:

In Convolution Neural Network, many kernels together form a special layer called the convolution layer. This particular layer responsible for performing the core functionality of the CNN: detecting features of sub regions of the input image like edge and colour scheme for good measure.

Convolutional Neural Network reduces the number of required parameters significantly by applying identical convolution layer to all sub regions of the input image. Also, the kernel that learns useful features for the training process can be re-used to detect such features virtually everywhere without the need to re-train. These mechanisms in the translation invariance property of CNN, meaning that it can identify patterns regardless of their location inside the input image.

VGG16

VGG16 was designed by Karen Simomyan and Andrew Zisserman from the Visual Geometry Group of University of Oxford. The architecture achieved the top-5 error of 6.8% on the ILSVRC-2014 dataset, securing the second place of the ILSVRC-2014challenge.

The VGG16 architecture employs several consecutives convolution layers which enhance the ability to captures important features of the image and therefore improve the overall accuracy. VGG16 is adapted to object detection and localization research field and typically serve the purpose of base neural network.

Challenges and limitation Even though Deep Learning is regarded as a state-of-the-art solution for learning large and complex dataset, there are still various challenges and barriers to address. This section focuses on two critical issues of Deep Learning: Over fitting and Computing expenses.

Over fitting

Modern neural network models often have millions of neurons (trainable parameters).Having a large number of trainable parameters enable the model to learn complex data but usually leads to the over fitting problem in which the model fails to yield good performance when it processes never-seen-before data. There are several strategies to combat the over fitting problem, yet it remains one of the most challenging issue in ANN optimization process.

Libraries and Framework

The used various libraries and frameworks to implement the real-time fashion classification system, namely Keras, TensorFlow, and TensorFlowJS. This section introduces the essential details about these libraries and framework, as well as analyses their advantages and disadvantages.

TensorFlow

TensorFlow was initially implemented by the Google Brain team for internal usages only. Google decided to make the first release of TensorFlow in 2015 as an open sourced software framework under the Apache License 2.0. Since then, TensorFlow has been massively adopted for both research and production usages by various software companies.

TensorFlow offers various level of abstraction for implementing and training neural network models depending on developers' needs. The framework provides an outstanding performance when it comes to computational speed thanks to the C++ kernel. The kernel allows developers to scale the applications without sacrificing speeder performance.

TensorFlow.js

TensorFlow.js is an open-sourced framework designed to build ML models in the JavaScript programming language. The Google's TensorFlow team announced the official release of TensorFlow.js for Artificial Intelligence application implementation in March 2018 [23]. The library allows deploying pre-trained TensorFlow model to the browser for running, re- training and modifying.

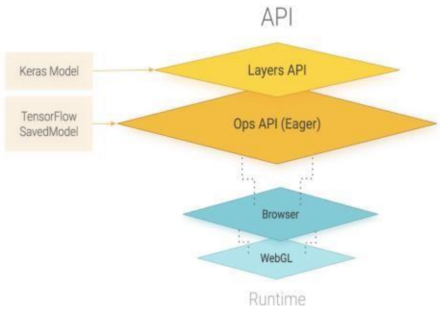


Figure . Architecture of TensorFlow.js application

As can be seen from above Figure, TensorFlow.js relies on WebGL for running execution and provides a high-level APIs for Deep Learning models building. TensorFlow.js also supports pre-built models implemented in Keras and Saved Models.

Training and validation

Understanding the characteristics of the dataset is a crucial part of any AI project. This section introduces the basic features of the dataset and visualization to make it more understandable. Additionally, this section covers basic details about Keras as a Neural Network Application Program Interface, the detailed architecture of the CNN, and the training process along with sample codes.

CHAPTER 3: DETAILED DESIGN

Working Model:

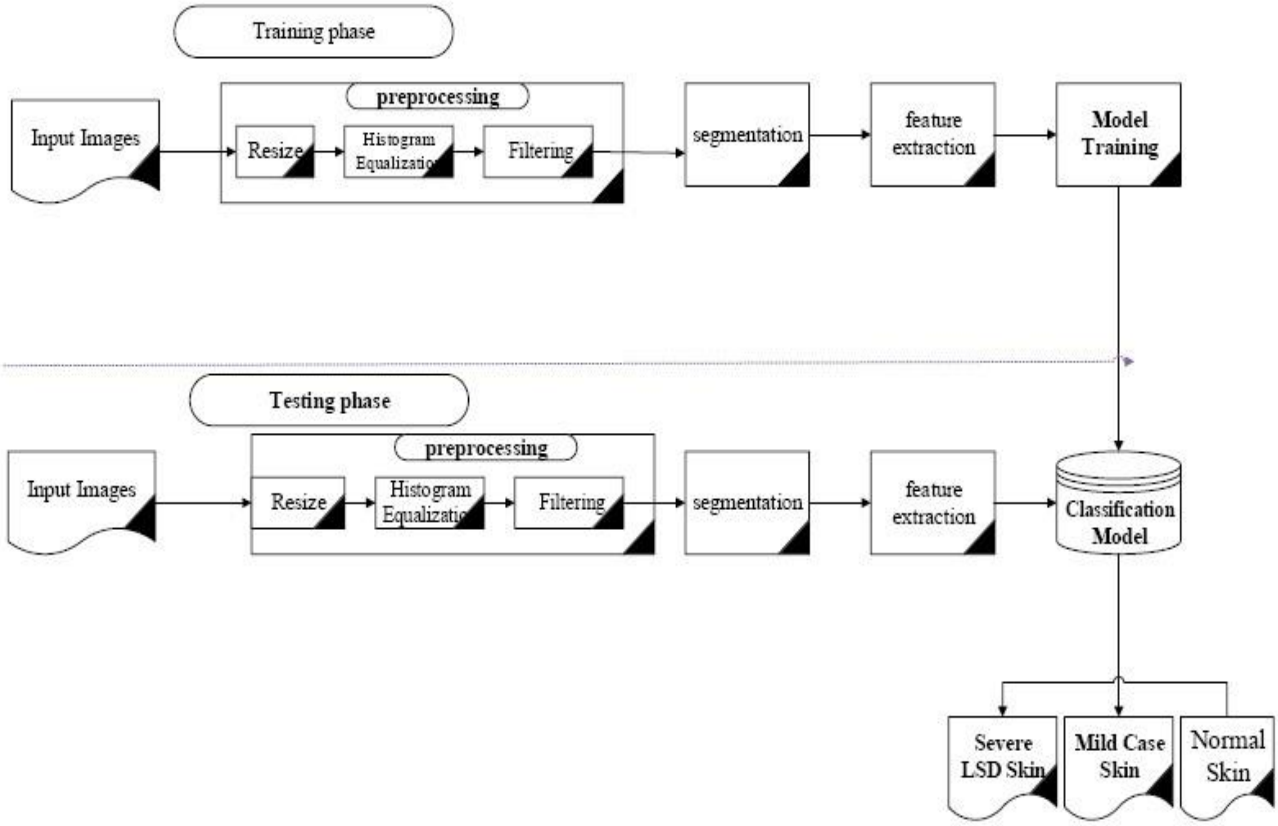


Figure .3 Working Model

In this study we acquired the datasets from [5] public source using google search engine and we transform the collected image data from both sources by cropping the area where Lumpy Skin Disease are present. By the help of veterinary Doctor, we divided the dataset into Severe Lumpy Skin Disease, Mild Lumpy Skin Disease and Normal Skin. The total number of images taken for this study, from each class; Normal skin, and Severe Lumpy Skin Disease infected skin and Mild Lumpy Skin Disease infected skin[7].

sample images that depicts the different levels of Lumpy Skin Disease infected cattle skin is given.

No	Category	Number of collected image data	After transformation
1	Normal skin	120	150
2	Severe Case Lumpy Skin Disease infected skin	50	150
	Total	250	450

(A)

(A) Normal skin

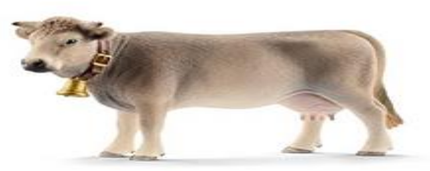
(B)

(B) Severe lumpy skin disease

Dataset :









CHAPTER 4: PROJECT SPECIFIC REQUIREMENTS

- **Hardware requirements:-**

  1. Windows 7 or higher.
  2. 4 GB RAM
  3. i3 Processor

- **Software Requirements:-**

  1. Anaconda Navigator
  2. Jupyter Notebook
  3. Python Libraries

CHAPTER 5: IMPLEMENTATION

Implementation of CNN :

The convolution neural network model was implemented using Keras – a framework API designed for building ANN implemented in Python and capable of using The ano, CNTK,and TensorFlow as backend.

VGG16 Implementation:

VGG16 is a convolution neural net (CNN) architecture which was used to win ILSVR(Imagenet) competition in 2014. It is considered to be one of the excellent vision model architecture till date. Most unique thing about VGG16 is that instead of having a large number of hyper-parameter they focused on having convolution layers of 3x3 filter with a stride 1 and always used same padding and maxpool layer of 2x2 filter of stride 2. It follows this arrangement of convolution and max pool layers consistently throughout the whole architecture. In the end it has 2 FC(fully connected layers) followed by a softmax for output. The 16 in VGG16 refers to it has 16 layers that have weights. This network is a pretty large network and it has about 138 million (approx) parameters.

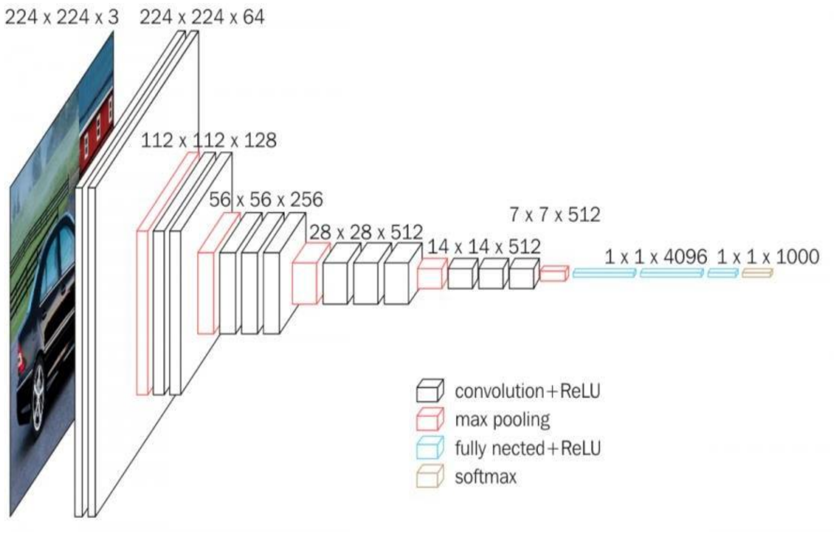


Fig.5 *The illustration of VGG16 architecture*

CHAPTER 6: RESULTS

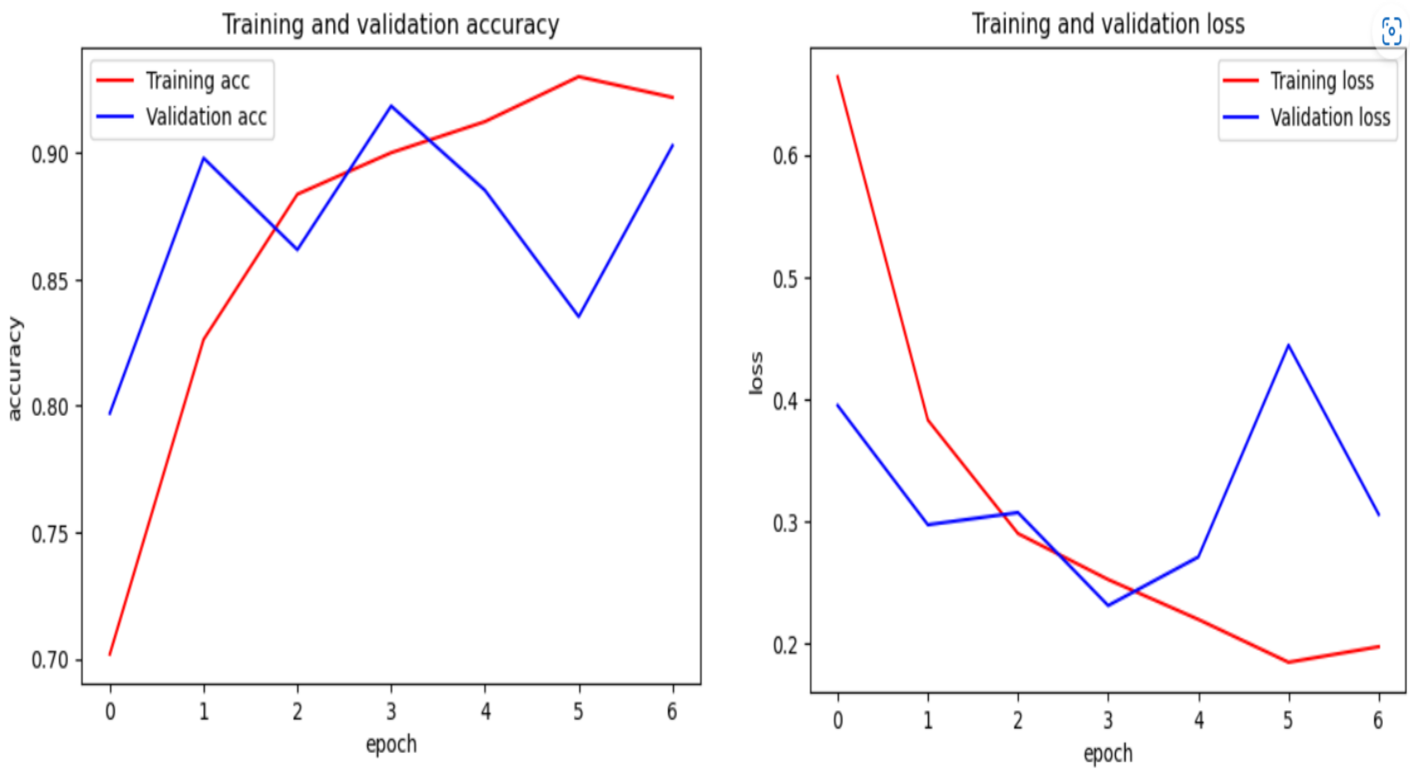


Fig. 7 Training and Validation Graph

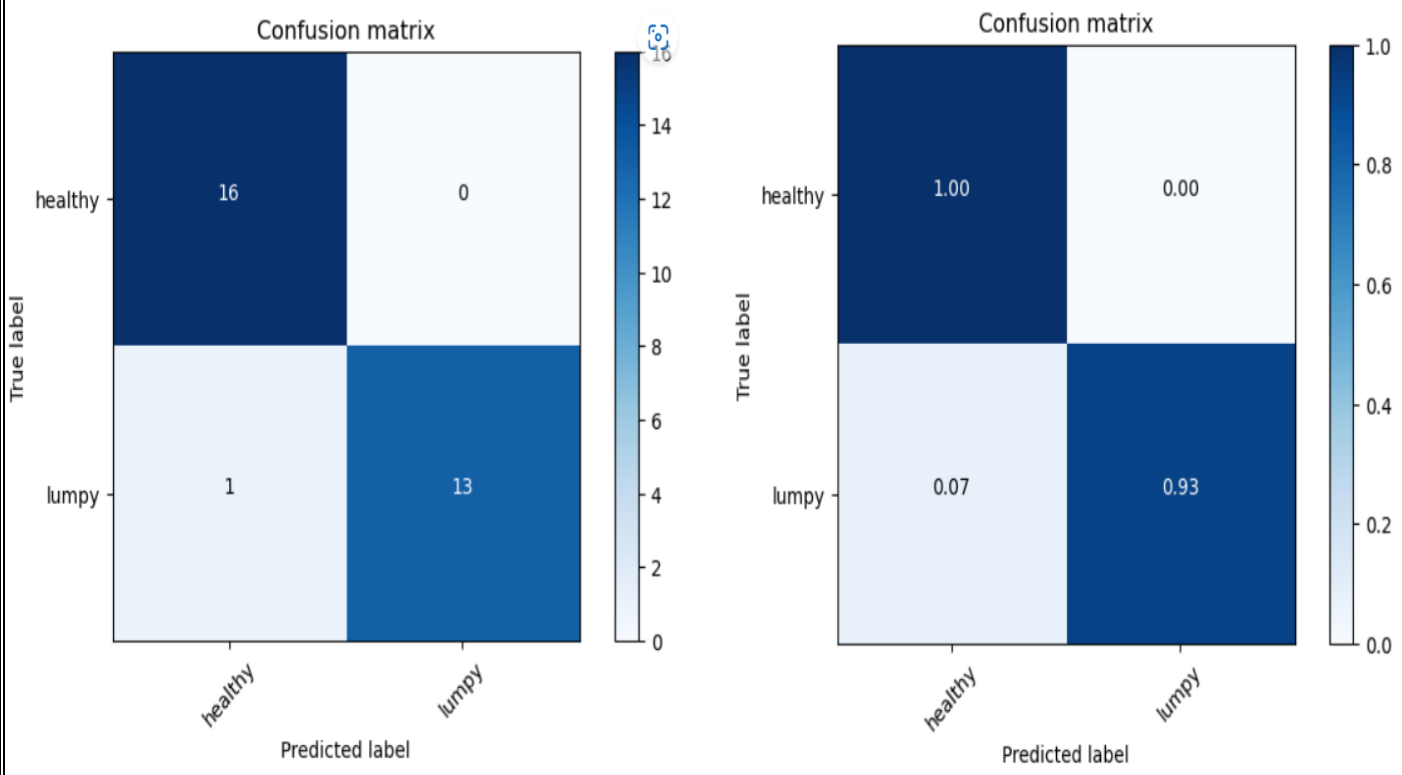
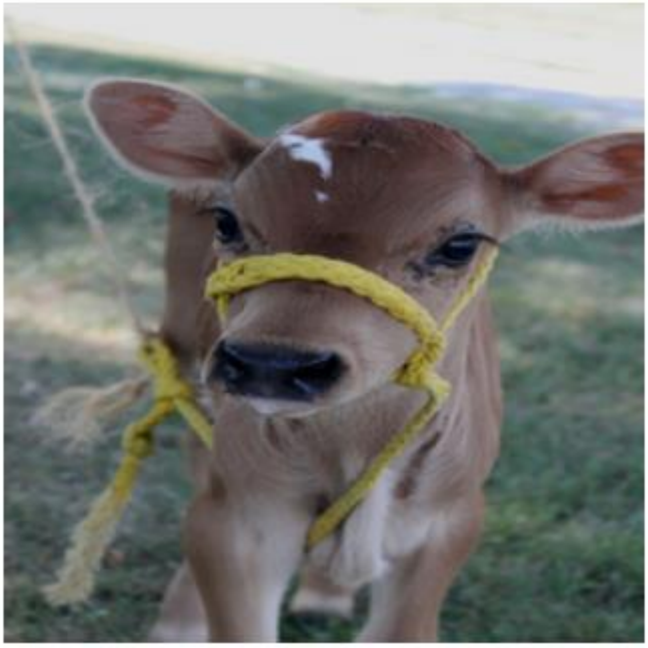


Fig. 8 Confusion Matrix

Output 1 :

```
[INFO] loading and preprocessing image...
1/1 [==============================] - 0s 417ms/step
1/1 [==============================] - 0s 62ms/step
ID: 0, Label: healthy 99.98%
ID: 1, Label: lumpy 0.02%
Final Decision:
.
..
...
1/1 [==============================] - 0s 47ms/step
ID: [1 0], Label: healthy
```



Output 2 :

```
[INFO] loading and preprocessing image...
1/1 [==============================] - 0s 284ms/step
1/1 [==============================] - 0s 56ms/step
ID: 0, Label: healthy 0.02%
ID: 1, Label: lumpy 99.98%
Final Decision:
.
..
...
1/1 [==============================] - 0s 31ms/step
ID: [0 1], Label: lumpy
```

Output 3:

```
[INFO] loading and preprocessing image...
1/1 [==============================] - 0s 356ms/step
1/1 [==============================] - 0s 59ms/step
ID: 0, Label: healthy 99.69%
ID: 1, Label: lumpy 0.31%
Final Decision:
.
..
...
1/1 [==============================] - 0s 60ms/step
ID: [1 0], Label: healthy
```

Output 4:

```
[INFO] loading and preprocessing image...
1/1 [==============================] - 0s 289ms/step
1/1 [==============================] - 0s 49ms/step
ID: 0, Label: healthy 0.01%
ID: 1, Label: lumpy 99.99%
Final Decision:
.
..
...
1/1 [==============================] - 0s 52ms/step
ID: [0 1], Label: lumpy
```

CHAPTER 7: CONCLUSION

Many research works are done so far in the area of human skin disease classification but only few studies conducted in animal skin disease classification.

In this study, we developed a method to detect animal Lumpy Skin Disease and classify as Healthy and Lumpy Skin using image processing.

In general, this study achieves early detection of Lumpy skin disease of cattle and classify as Healthy and Lumpy Skin. The contribution of this study includes preparation of Lumpy skin disease Image dataset, construction of Lumpy skin disease Image classification model and Method to use local information to known incidence of animal Epidemic disease.

REFERENCES

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza , Bing Xu, David Warde-Farley, SherjilOzair, “Generative Adversarial Nets,” in Advances in Neural Information Processing Systems 27, 2014.

[2] G. Ognjanovski, “Towards Data Science,” 14 January 2019. neuralnetworks- and-backpropagation-machine-learning-made-easy.

[3] Emergence of the first Lumpy Skin Disease outbreak among livestock in the Republic of Kazakhstan in 2016 (doi: 10.20944/preprints202011.0298).

[4] Identify Animal lumpy Skin Disease Using Image Processing and Machine Learning A Thesis presented by Elias Girma (July 2021).

[5] Kurt Hornik, Maxwell Stinchcombe, Halbert White, “Multilayer feedforward networksare universal approximators,” Neural networks, vol. 2, no. 5, pp. 359-366, 1989.

[6] Pravin S. Ambad and A. S. Shirsat, (2016), “A Image analysis System to Detect Skin Diseases”, IOSR Journal of VLSI and Signal Processing (IOSRJVSP) Volume 6, Issue 5, Ver. I (Sep. - Oct. 2016).