A DEEP LEARNING BASED APPROACH FOR FOOT AND MOUTH DISEASE DETECTION

Timothy Kudzanayi Kuhamba
(Student Number R056 589L)

Electrical Engineering Department
University of Zimbabwe
Harare, Zimbabwe

Supervisor: Dr Marisa

Master’s Thesis submitted to the Faculty of Electrical Engineering, the University of Zimbabwe in partial fulfilment of the requirements for the degree of Masters in Communication Engineering
COPYRIGHT

All rights reserved. No part of this innovation project may be reproduced, stored in any retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise from scholarly purpose, without the prior written permission of the author or of The University Of Zimbabwe on behalf of the author.
DECLARATION

I, Timothy Kuhamba do hereby declare that this project is a result of my work except where sources have been acknowledged. This work has not previously been accepted in substance for any degree and is not being concurrently submitted in candidature for any degree.

Student’s Signature: ………………………………. Date …………………………….

(Timothy Kuhamba)

Supervisor’s Signature: ………………………………. Date …………………………….

(Dr Marisa)
ABSTRACT

The research gives a new approach for the detection of foot and mouth disease (FMD) using deep learning techniques. The purpose of this study is to provide a timely and accurate early detection of FMD in cattle based on the symptoms. Image data sets of foot and mouth diseased and healthy cattle were collected, preprocessed and different deep learning models were trained to learn features of both healthy and diseased cattle so that these features can be recognised for the classification on images never seen by the deep learning system. There was a difficulty in acquiring diseased cattle images thus a smaller dataset was acquired from the Internet, Veterinary department, European Union Foot and Mouth division (EuFMD) and also Pirbright Institute. Healthy cattle images were taken from the University of Zimbabwe farm with mixed cattle breeds. Different deep learning architectures using transfer learning were assessed and the Densenet 201 outperformed other models with an accuracy of 93.75%, precision 0.98, sensitivity 1.0, specificity 0.9916, AUC 0.99 and ROC of 0.9958. The results also showed the importance of colour information and image focus on the identification of FMD. The study also showed that transfer learning is the best for image recognition when you have a smaller dataset and offers deployable FMD detection system however there is a need for a larger dataset for the detection and identification of each disease symptom. The deep learning system will be used for the development of a mobile application for the detection of FMD. The research under investigation is not intended to replace existing solutions for disease diagnosis, but rather to supplement them.
ACKNOWLEDGEMENTS

I give my gratitude to all the authors who were involved in the contribution of the articles that contain information related to my project. I also thank my supervisor, Dr. Marisa, for helping me through to complete my project. I would also like to thank Dr M Munochiweyi for his important comments to this study. I would also like to extend my gratitude to the European Union Foot and Mouth division and Pirbright Institute for giving me foot and mouth diseased images. The staff of the University of Zimbabwe farm in Mazoe played a major role in the acquisition of healthy images as they helped in the preparation of cattle for images to be taken.

I would also want to thank my family for rendering necessary support in the difficult times of the project. To my colleague a brother Wadzanai J Zondai, Tinashe Zitsanza I would like to thank for the support we have rendered each other throughout the MSc program. Davison Magaya, Isheanesu Newengo thank you so much for the support you offer during the project. Effort Magoso you have always shouldered me throughout my academic career and may God bless you.

Above all, I give all thanks and honour to the creator of all, the almighty God. I give my gratitude to my mother Dora Kuhamba, my wife Portia Kodzwa and also family members for the support they always give me throughout my academic journey.
TABLE OF CONTENTS

CONTENTS

COPYRIGHT ........................................................................................................................................... ii
DECLARATION ........................................................................................................................................ iii
ABSTRACT ................................................................................................................................................ iv
ACKNOWLEDGEMENTS ......................................................................................................................... v
TABLE OF CONTENTS ........................................................................................................................ vi
LIST OF FIGURES ................................................................................................................................ xi
LIST OF TABLES ................................................................................................................................... 15
CHAPTER ONE: INTRODUCTION ........................................................................................................ 16
  1.1. Introduction .................................................................................................................................... 16
  1.2. Background ................................................................................................................................... 17
  1.3. Problem Statement ....................................................................................................................... 19
  1.4. Aim ................................................................................................................................................ 19
  1.5. Objectives ..................................................................................................................................... 19
  1.6. Expected Outcomes, Significance or Rationale ............................................................................. 19
CHAPTER TWO: LITERATURE REVIEW .............................................................................................. 21
  2.1. Introduction ................................................................................................................................... 21
  2.2. Convolutional neural networks for Image Classification .......................................................... 21
  2.3. Case studies .................................................................................................................................. 22
  2.4. Convolution ................................................................................................................................... 24
    2.4.1. Convolutional neural networks ............................................................................................. 25
    2.4.1. What is a Convolutional neural network? .............................................................................. 25
    2.4.2. How CNN recognises images? ............................................................................................... 26
    2.4.3. Hidden layer ............................................................................................................................ 26
    2.4.4. Output layer ............................................................................................................................ 27
  2.5. Feature extraction ......................................................................................................................... 27
2.9. Deep learning architectures ......................................................... 46
  2.9.6. ResNet .................................................................................. 43
  2.9.7. MobileNet ............................................................................. 43
  2.9. Deep learning architectures ......................................................... 46
  2.10.1. Alex net ............................................................................... 46
  2.10.2. VGGNet ............................................................................. 47
  2.10.3. Pretrained VGG 16 ................................................................. 48
  2.10.4. Google Net (Inception) ............................................................ 50
  2.10.5. Resnet ................................................................................ 53
  2.10.6. DenseNet ............................................................................. 55

2.10. Foot and Mouth ............................................................................ 59
  2.10.1 Clinical signs of foot and mouth ................................................. 63
  2.10.8 EuFmd .................................................................................. 69

2.11. Economic Impact of Foot and Mouth ............................................ 69

2.12. Evaluating classifier Predictions .................................................... 70
  2.13 Confusion Matrix ..................................................................... 70
  2.14.1. Receiver Operating Curve ....................................................... 75

CHAPTER THREE: METHODOLOGY ......................................................... 76

3.1. Introduction .................................................................................. 76

3.2. Systems Engineering V System model ........................................... 76
  3.2.1. Steps in the detection of Foot and mouth .................................. 76
  3.2.2. Mission requirements ............................................................... 77
  3.2.3. Data Acquisition .................................................................. 77

3.3. Acquisition of Healthy cattle images ............................................. 78
  Day1 .............................................................................................. 78
  Day 2 ........................................................................................... 78
  Day 3 ........................................................................................... 79

3.4. Journals and Books ..................................................................... 80
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.5. Data Preparation</td>
<td>80</td>
</tr>
<tr>
<td>3.5.1. Image Pre-processing</td>
<td>81</td>
</tr>
<tr>
<td>3.5.2. Selecting the area of interest</td>
<td>81</td>
</tr>
<tr>
<td>3.5.3. Image Augmentation</td>
<td>82</td>
</tr>
<tr>
<td>3.5.4. Resizing Images</td>
<td>83</td>
</tr>
<tr>
<td>3.6. Foot and Mouth Detection</td>
<td>83</td>
</tr>
<tr>
<td>2.14 Training Strategy</td>
<td>85</td>
</tr>
<tr>
<td>3.7.1. Binary Classifier Training</td>
<td>85</td>
</tr>
<tr>
<td>3.7.2. Multi Classifier Training</td>
<td>86</td>
</tr>
<tr>
<td>3.7.3. Testing of images taken by a smartphone</td>
<td>86</td>
</tr>
<tr>
<td>3.7.4. Deep Learning Models</td>
<td>86</td>
</tr>
<tr>
<td>3.7.5. Evaluation of Model</td>
<td>87</td>
</tr>
<tr>
<td>3.7.6. Hardware Consideration</td>
<td>88</td>
</tr>
<tr>
<td>3.7. Program Verification Methods</td>
<td>89</td>
</tr>
<tr>
<td>3.8. Risk Management</td>
<td>90</td>
</tr>
<tr>
<td>CHAPTER FOUR: RESULTS AND DISCUSSIONS</td>
<td>92</td>
</tr>
<tr>
<td>4.1. Introduction</td>
<td>92</td>
</tr>
<tr>
<td>4.2. VGG16</td>
<td>92</td>
</tr>
<tr>
<td>4.2.2 Inception v3</td>
<td>93</td>
</tr>
<tr>
<td>4.2.3 Densenet 201</td>
<td>93</td>
</tr>
<tr>
<td>4.2.4 Resnet v2 152</td>
<td>93</td>
</tr>
<tr>
<td>4.2.5 Resnet 50</td>
<td>94</td>
</tr>
<tr>
<td>4.2.6 InceptionResnetv2</td>
<td>94</td>
</tr>
<tr>
<td>4.3. Analysis of evaluation metrics</td>
<td>94</td>
</tr>
<tr>
<td>4.4. Evaluation Metrics</td>
<td>95</td>
</tr>
<tr>
<td>4.5. Confusion Matrix</td>
<td>97</td>
</tr>
<tr>
<td>4.5.1. Densenet 201</td>
<td>99</td>
</tr>
<tr>
<td>Section</td>
<td>Title</td>
</tr>
<tr>
<td>---------</td>
<td>-----------------------------------------------------------------------</td>
</tr>
<tr>
<td>4.5.2.</td>
<td>Resnet 152 v2</td>
</tr>
<tr>
<td>4.5.3.</td>
<td>Resnet50</td>
</tr>
<tr>
<td>4.5.4.</td>
<td>Inception Resnetv2</td>
</tr>
<tr>
<td>4.6.</td>
<td>Comparison of Evaluation Metrics</td>
</tr>
<tr>
<td>4.7.</td>
<td>Evaluating the performance on test data</td>
</tr>
<tr>
<td>4.7.1.</td>
<td>ROC and AUC curve graphs</td>
</tr>
<tr>
<td>4.7.2.</td>
<td>Resnet 152v2 Classification Report</td>
</tr>
<tr>
<td>4.8.</td>
<td>Predictions on test images</td>
</tr>
<tr>
<td>4.8.1.</td>
<td>Batch testing</td>
</tr>
<tr>
<td>CHAPTER FIVE: CONCLUSION</td>
<td></td>
</tr>
<tr>
<td>APPENDICES</td>
<td></td>
</tr>
<tr>
<td>Appendix A: Project plan</td>
<td></td>
</tr>
</tbody>
</table>
LIST OF FIGURES

Figure 1 Convolution Operation ................................................................. 24
Figure 2 Conceptual analog between real neurons A and artificial neurons B Lee et al 2017 25
Figure 3: Conceptual analog between real neurons A and artificial neurons B Lee et al 2017 Input layer ........................................................................................................ 26
Figure 4: Comparison of shallow learning and deep learning (Lee et al, 2017) .............. 27
Figure 5: Convolved feature ........................................................................ 27
Figure 6 Pooling layer .................................................................................. 28
Figure 7 Flattening ...................................................................................... 28
Figure 8 Fully connected layer ...................................................................... 29
Figure 9: Illustration of a typical convolutional neural network architecture setup (Nguyen et al, 2017). .................................................................................. 30
Figure 10 Activation functions ..................................................................... 30
Figure 11: Sigmoid function ........................................................................ 31
Figure 12: Tanh (Wani et al., 2020) .............................................................. 31
Figure 13: Softmax function ........................................................................ 32
Figure 14: Relu (Rectified Linear Unit) ........................................................... 32
Figure 15: Artificial Intelligence, Machine learning and deep learning ................. 34
Figure 16: Artificial Intelligence .................................................................... 35
Figure 17 Convection Machine Learning and Deep Learning ............................ 36
Figure 18: Feature extraction in Deep learning ............................................... 36
Figure 19: Showing a Deep and Shallow network .......................................... 37
Figure 20: A neural network parametrized by its weights ............................... 38
Figure 21: A loss function measures the quality of network output .................. 38
Figure 22: BackPropagation ....................................................................... 39
Figure 23: The loss score is used as a feedback signal to adjust the weights .......... 39
Figure 24: Showing a VGG Pretrained model ................................................. 45
Figure 25: Architecture diagram for Lenet5 .................................................. 46
Figure 26: Architecture diagram for AlexNet .................................................. 47
Figure 27: Architecture diagram for VGGNet-16 .......................................... 48
Figure 28: VGG-16 Model Architecture ........................................................ 49
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>29</td>
<td>Heterogeneous set of convolutions to cover spread out</td>
</tr>
<tr>
<td>30</td>
<td>Conceiving the Inception module</td>
</tr>
<tr>
<td>31</td>
<td>Inception module naive version</td>
</tr>
<tr>
<td>32</td>
<td>Inception module with dimension reductions</td>
</tr>
<tr>
<td>33</td>
<td>Inception Architecture</td>
</tr>
<tr>
<td>34</td>
<td>Residual learning</td>
</tr>
<tr>
<td>35</td>
<td>Architecture for Resnet 34</td>
</tr>
<tr>
<td>36</td>
<td>Densenet with three blocks (Huang et al., 2017)</td>
</tr>
<tr>
<td>37</td>
<td>Diagram of a Dense block</td>
</tr>
<tr>
<td>38</td>
<td>Five Layer Dense block</td>
</tr>
<tr>
<td>39</td>
<td>Architecture diagram for DenseNet</td>
</tr>
<tr>
<td>40</td>
<td>Structure of the Foot and Mouth Virus (Jamal and Belsham, 2013)</td>
</tr>
<tr>
<td>41</td>
<td>Serotypes in Africa based on occurrence from 1990 to 2013 (Tekleghiorghis et al., 2016)</td>
</tr>
<tr>
<td>42</td>
<td>Serotypes in different regions</td>
</tr>
<tr>
<td>43</td>
<td>Foot and Mouth Outbreaks in Zimbabwe from 1931 to 2016 (Guerrini et al., 2019)</td>
</tr>
<tr>
<td>44</td>
<td>Number of foot and mouth outbreaks during the period 1931 to 2016 (Guerrini et al., 2019)</td>
</tr>
<tr>
<td>45</td>
<td>Infrared Detection of FMDV in calves (Schaefer et al., 2004)</td>
</tr>
<tr>
<td>46</td>
<td>Digital and infrared images of cattle without (A) or with (B) fever and note that the lower temperatures (blue-green) in the animal without fever or viremia versus the higher temperatures (orange-red) in the viremic and feverish animal</td>
</tr>
<tr>
<td>47</td>
<td>Thermal images taken by (Gloster et al., 2011) with an image of the I button</td>
</tr>
<tr>
<td>48</td>
<td>Specificity and Sensitivity</td>
</tr>
<tr>
<td>49</td>
<td>Receiver operating Curve</td>
</tr>
<tr>
<td>50</td>
<td>Software Engineering V Model</td>
</tr>
<tr>
<td>51</td>
<td>Flow chart for Foot and Mouth Detection</td>
</tr>
<tr>
<td>52</td>
<td>Images taken whilst cattle were grazing</td>
</tr>
<tr>
<td>53</td>
<td>Process for taking images for tongue, gum and teats</td>
</tr>
<tr>
<td>54</td>
<td>Process for taking images of the feet</td>
</tr>
<tr>
<td>55</td>
<td>Training Directory</td>
</tr>
</tbody>
</table>
Figure 56: Validation Directory ................................................................. 81
Figure 57: Region of interest selected (Tongue) ........................................ 82
Figure 58: Displaying augmented images .................................................. 82
Figure 59: Softmax Classification ............................................................... 83
Figure 60: Flow Chart for the Classification of Foot and Mouth ................... 84
Figure 61: Sample Gray Scale images left side healthy cattle and right gum lesion ........ 85
Figure 62: Learning process ................................................................. 87
Figure 63: VGG16 accuracy and loss performance .................................... 92
Figure 64: Inception v3 accuracy and loss performance .............................. 93
Figure 65: Densenet 201 accuracy and loss performance ............................ 93
Figure 66: Resnet 15v2 accuracy and loss performance ............................. 93
Figure 67: Resnet50 accuracy and loss performance .................................... 94
Figure 68: InceptionResnetv2 accuracy and loss performance .................... 94
Figure 69: Comparison with other different studies ...................................... 97
Figure 70: Comparison of the predictions for different deep learning architectures ........ 98
Figure 71: VGG16 Confusion Matrix and Classification Report ..................... 99
Figure 72: Inception V3 Model ............................................................... 99
Figure 73: Densenet 201 Confusion Matrix and Classification Report .......... 100
Figure 74: Resnet 152v2 Confusion Matrix and Classification Report ........... 100
Figure 75: Resnet 50 Confusion Matrix and Classification Report ............... 100
Figure 76: Multiclassification Training and Accuracy Report ...................... 101
Figure 77: Inception v2 Confusion Matrix ............................................... 102
Figure 78: Confusion Matrix for densenet ............................................... 102
Figure 79: Specificity and Sensitivity ...................................................... 103
Figure 80: Evaluating Performance for different deep learning architectures ....... 104
Figure 81: VGG16 ROC ........................................................................ 105
Figure 82: Inception v3 ROC ............................................................... 105
Figure 83: Densenet 201 ROC .............................................................. 105
Figure 84: Resnet50 ROC ................................................................. 106
Figure 85: Resnet 152v2 ROC ............................................................. 106
Figure 86: InceptionResnetv2 ............................................................... 106
Figure 87: Comparison of ROC and AUC .............................................. 107
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>88</td>
<td>Prediction of Drooling and non-drooling image</td>
<td>108</td>
</tr>
<tr>
<td>89</td>
<td>Prediction of gum lesion and non-gum lesion images</td>
<td>108</td>
</tr>
<tr>
<td>90</td>
<td>Prediction of drooling and non-drooling image</td>
<td>109</td>
</tr>
<tr>
<td>91</td>
<td>Prediction of teat lesion and non lesion</td>
<td>109</td>
</tr>
<tr>
<td>92</td>
<td>Prediction of Tongue lesion and non-tongue lesion</td>
<td>110</td>
</tr>
<tr>
<td>93</td>
<td>Batch testing</td>
<td>110</td>
</tr>
</tbody>
</table>
LIST OF TABLES

Table 1 Pre-trained Classification Models .................................................................42
Table 2: Details of various layers of Alex Net .............................................................46
Table 3: Details of various layers of VGGNet-16 .........................................................47
Table 4: Details of various GoogleNet(Inception) .......................................................52
Table 5: Residual architecture parameters .................................................................54
Table 6: Dense Architecture parameters ..................................................................59
Table 7 Confusion matrix .........................................................................................71
Table 8: Experimental Hardware ............................................................................88
Table 9 Summary of FMD distribution by serotype and topotype for Southern Africa, .......88
Table 10: Program Verification Methods ..................................................................89
Table 11: Risks and Counter measures ...................................................................91
Table 12 Accuracy and loss of training, validation for coloured images ..................92
Table 13 Studies in disease identification ................................................................96
Table 14 Summary of confusion Matrix for different models ....................................98
Table 15 Evaluating Performance Metrics ...............................................................103
Table 16 ROC and AUC for different deep learning architectures ............................107
CHAPTER ONE: INTRODUCTION

1.1. Introduction

Foot-and-mouth Disease (FMD) is an exceptionally infectious livestock disease caused by the Foot and Mouth Disease Virus (FMDV; variety Aphthovirus, family Picornaviridae), (Control, 2012) and (Foot and Mouth Disease (FMD), 2015). FMD affects cloven-hoofed domestic and around 70 wild creature species for example African Buffalo, including cows, pigs, and little ruminants (Jamal and Belsham, 2013) and (Infection, 2018). The researcher engaged the Zimbabwe Veterinary department to enquire which disease has been problematic in recent years and FMD was pointed out as one of the diseases.

Any infection among animals requires a monetary effect on the livestock keepers do as to contain the disease. These impacts can be classified into two groups; direct losses and indirect losses. Direct losses are associated with death, abortion; Indirect losses are linked to the costs associated with preventing the disease i.e vaccines, costs of quarantining animals. FAO reports that FMD outbreaks were the disease has been previously eradicated cause losses of approximately USD 1.5 billion per year and also the cost burden in endemic regions is estimated to be mire that USD 6.5 Billion a year. (Babb and Emery, 2018). Thus, reducing FMD in endemic countries by a coordinated control strategy at the global and regional level is of shared interest and should be considered a global public good.

(National Agriculture Policy Framework, 2018) highlights that there has been a significant decline in crop and livestock production and this has been attributed to lows levels of mechanisation, lack of technology and innovation, reliance on rain-fed agriculture; limited access to market information and marketing facilities. The National Agriculture Policy Framework 2018-2030 reports that the significant increase in the adoption of smartphones in Zimbabwe provides an opportunity for farmers to access useful information in the face of inadequate and/or poorly resourced extension staff. According to the POTRAZ report for 2018, the internet penetration rate increased by 2.8% during the second quarter. The largest internet connections were attributed to the fibre and mobile subscriptions which rose by 17% and 6% respectively. Thus the development of a tool on a smartphone mobile application will lead to early detection of foot and mouth.
Zimbabwe seeks to re-access beef markets as highlighted in the EU report on value chain analysis of beef in Zimbabwe however the country must reinstate the veterinary control to manage transboundary diseases such as foot and mouth disease (FMD). The report also emphasizes that Export sales and Food and Mouth Control (FMC) are the focus of the proposed “Command Livestock, Fisheries and Wildlife Program”, aiming to return Zimbabwe to competitive export (EU, 2018). According to the Food and Agriculture Organization (FAO) of the United Nations Foot-and-Mouth Disease Situation report, three FMD outbreaks occurred in cattle in January 2018 in Mashonaland East. (Gizaw, 2018).

This study focuses on the utilisation of deep learning techniques a subset of artificial intelligence in the early detection of FMD. Computer-aided deep learning technique will be used for the development of a mobile application for FMD detection. This tool will reduce the social and economical associated costs with the FMD disease. (Smith, 2019) highlighted that artificial intelligence will bring more quick applications will be to improve exactness data about what's going on the farms by improving what is being identified and estimated. (Sung, 2018) also alluded that advancement in artificial intelligence and big data will take into account precision agriculture, for example, yield checking, diagnosing bug bugs, estimating soil dampness, diagnosing harvest time, and observing harvest well-being status.

1.2. Background

(Guerrini et al., 2019) carried out a study for the FMD outbreaks in Zimbabwe during the period 1931 to 2016. The study came up with FMD control measures which are; early detection and reporting of the FMD to limit the spread of the disease, quarantining of the infected animals at the premised where it was detected, containing the spread of the disease by restricting the movement of the animals from the premises, vaccination of cattle to eradicate the disease and continuous surveillance in the FMD prone diseases.

In 2012 the Food Agriculture Organisation (FAO ) and the World Organisation for Animal Health developed a 15 years Global Control Strategy to reduce FMD in endemic countries and also maintain the status of FMD free countries. However, it was noted that the endemic may not be eradicated within 15 years but there will be significant progress in reducing the burden caused by FMD. FAO reports that there has been a high prevalence of Foot and Mouth in many African, the Middle East, Asia, and South America. Europe, North and Central America, Pacific nations. (Knight-Jones, McLaws and Rushton, 2017) noted that FMD endemic areas
contain three-quarters of the world mostly poorest livestock keepers are vulnerable. The Global control strategy was developed tapping from the experiences from different regions. The strategy includes three components, improving global FMD control, the strengthening of Veterinary Services and the control of other Transboundary Animal Disease (TADs). (Babb and Emery, 2018).

Traditionally disease detection and diagnosis had been by experts using experience through observations of the symptoms using the visual examinations. However, (Ferentinos K, 2018) noted that experienced agronomists and pathologists often fail successfully to diagnose the specific disease which led to mistaken conclusions. (Mohanty et al, 2016) pointed out that an automated computational system offers valuable assistance to the agronomists in disease detection. It was also noted that a system that is accessible over the web or via a mobile app can be a valuable tool that can assist farmers in parts of the world lacking appropriate infrastructure for pathological and agronomical advice.

Technology has always been involving with the introduction of machine learning, artificial intelligence, deep learning, and computer vision have changed the way machines and systems behave in handling different functions and systems. Deep learning is a subset of machine learning that has networks that are capable of learning from data that is unstructured or unlabelled and works similar to the functioning of the brain. Deep learning can also be defined as an application of machine learning that uses complex algorithms and deep neural networks to train a model.

A neural network is a computing model whose layered structure resembles the network structure of neurons in the brain, with layers of connected nodes. It can learn from data, so it can be trained to recognize patterns, classify data, and forecast future events. A neural network breaks down your input into layers of abstraction. It consists of an input layer, one or more hidden layers, and an output layer. The layers are interconnected via nodes or neurons with each layer using the output of the previous layer as its input. Its main function is to receive a set of inputs, perform calculations and then use the output to solve the problem. It is, however, important to note that the research is done to complement rather than to replace disease detection.
1.3. Problem Statement

There is a shortage of veterinary specialists across the country due to brain drain which leaves farmers cattle vulnerable, timeous advice for the detection of FMD as the diseases leads to loss of production of livestock meats and also milk to farmers and also a major impediment as countries with foot and mouth faces trade restrictions moreover the disease is difficult and costly to control and eradicate.

1.4. Aim

To investigate how deep learning architectures can be used to detect foot and mouth

1.5. Objectives

i. To detect foot and mouth diseases using deep learning architectures

ii. To assess deep learning architecture model performance for the detection of foot and mouth

iii. To come up with recommendations when taking images for the detection of foot and mouth

1.6. Expected Outcomes, Significance or Rationale

The study will generate more insights into the foot and mouth disease detection of foot and mouth in cattle as there is not such a system in Zimbabwe that can assist pathologists and veterinary in carrying out their work. Early identification of cattle diseased with foot and mouth is very vital in preventing disease outbreaks as FMD are difficult and costly to control and eradicate.

There is a shortage of veterinary specialists across the country due to brain drain which leaves farmers cattle vulnerable timeous advice for the detection of cattle diseases like foot and mouth and this leads to loss of production of livestock meats and also milk to farmers.

The National Agriculture Policy Framework 2018-2030 reports that the significant increase in the adoption of smartphones in Zimbabwe provides an opportunity for farmers to access useful information in the face of inadequate and/or poorly resourced extension staff. The development of (mobile application ) can be a valuable tool that can assist farmers in parts of the world lacking appropriate infrastructure for pathological and agronomical advice.
Zimbabwe seeks to return to the competitive export of livestock, fisheries, and wildlife and detection and control foot and mouth is very vital in achieving the return into the export market as highlighted in the EU report (Gizaw, 2018). A country that has been affected by FMD faces trade restrictions thus Zimbabwe needs tools that easily used to detect and contain before it quickly spreads.

Food and Agriculture Organisation (FAO) reports that FMD outbreaks were the disease has been previously eradicated cause losses of approximately USD 1.5 billion per year and also the cost burden in endemic regions is estimated to be mire that USD 6.5 Billion a year (Babb and Emery, 2018). Thus, reducing FMD in endemic countries by a coordinated control strategy at the global and regional level is of shared interest and should be considered a global public good. Early detection of FMD will significantly improve the control strategy at the global and also regional level.

The research has attracted the attention of European Union Foot and Mouth Division (FMD) and Pribright who has been assisting in providing images for cattle diseased with FMD as they are not such a system worldwide.
CHAPTER TWO: LITERATURE REVIEW

2.1. Introduction

This chapter will look into the recent studies carried out in the investigation of the use context of deep learning in the detection of ailments in plant life, animals and humans. It will dig deeper into the context of how it is implemented and looking at the guidelines from exceptional studies.

Computer-aided detection systems were introduced and developed starting in the early 2000s. However, there were limitations in the detections of false positives, which resulted in humans having greater assessment times. (Lee et al., 2017). The ability of deep learning in extracting keep features in the data has resulted in being implemented many medical imaging applications; breast cancer (Su et al., 2018), lung cancer (Hua et al, 2015 & Kumar et al 2015), Alzheimer disease ) Suk HI & Shen D, 2013). Deep learning has been useful not only useful for diagnostics but can also predict disease recurrence, progression, and patient outcome (Nirschl et al 2017).

More recently, there has been an increase in the studies of disease detection of plants, animals, and humans using deep convolutional neural networks. Visual recognition is a challenging task for humans but also difficult for an automated system as different images have different properties (Caldera, Rassau and Chai, 2018). Images have different illumination intensity when they are taken and also the position, scale, view or background. The automation system that must build must be able to take into account the previously mentioned views to correctly and accurately classify the image (Lu et al., 2018).

2.2. Convolutional neural networks for Image Classification

In recent years, CNN has been used in the areas of image classification, speech recognition, language processing (Su et al, 2015), (Bishop, 2006), (LeCun et al, 1989) and (Somoyan, 2014 ). These models have outclassed humans in image recognition (Ciregan et al 2012) and also
the recent advancement in computing powers and architecture of convolutional neural networks (Devin et al. 2016).

2.3. Case studies

(Zin et al., 2018) presented a new way of using deep learning. The system uses a deep convolutional network over the cropped cow pattern images for the identification of the individual cow pattern. The system got an accuracy of 86.8% for automatically detecting and cropping of cow's body region and 97.01% for individual cow’s identification.

(Machado, Mendoza and Corbellini, 2015) investigated the detection of the Bovine Viral Diarrhea Virus (BVDV) in cattle using the machine learning random forest (RF) prediction model and performed the variable importance analysis to identify factors associated with BVDV. (Machado, et al, 2015) noted that RF provides a predictive power that can be used to identify potential risk factors for BVDV.

(Esteva et al., 2017) examined the utilization of deep neural systems for the distinguishing proof of skin malignant growth. In the examination, the deep learning learning calculations and dermatologists were researched whether they could recognize harmful versus favourable sores of epidermal (keratinocyte carcinoma contrasted with kindhearted seborrheic keratosis) or melanocytic (dangerous melanoma contrasted with amiable nevus) root. (Esteva et al., 2017) showed the adequacy of deep learning in dermatology, a system that we apply to both general skin conditions and explicit tumors. However, (Esteva et al., 2017) noted that automated images classification is quite difficult because of the fine-grained variability in the appearance of the skin lesions.. (Codella et al., 2017) also evaluated melanoma skin cancer by segmenting lesion and detecting the affected area. The framework performed better giving a precision of 75% contrasted with 70% from eight authority dermatologists (Codella et al., 2017).

Various examinations have been done in the detection of diseases in different crops. (Tm et al., 2018)(Amara, Bouaziz and Algergawy, 2017b)(Mohanty, Hughes and Salathé, 2016) (Kamilaris et al., 2017) built up a CNN based framework that distinguishes 13 sorts of diseases out of the healthy ones in every five crops utilizing images downloaded from the web and the accuracy was 96.3%.

(Mohanty, Hughes and Salathé, 2016) assessed the presentation of two CNN designs AlexNet and GoogleNet to recognize 26 infections remembered for 14 harvests utilizing Plant Village
Dataset. In this study, Right images of diseased and heath plants were utilized. Mohanty noted that preparing a model takes a great deal of time and requires a PC that has great high-specialized specifications. It was also emphasised that the model must be able to be used on a cell phone. Hughes, D.P.; Salathe, M, 2016 likewise utilized the Plant Village dataset however utilizing VGG-16 model prepared with move figuring out how to identify gasp infections.

(Gutstein, 2017) evaluated the performance of three deep learning meta-architectures Faster Region-based Convolutional Neural Network (Faster R-CNN), Region-based Fully Convolutional Network (R-FCN), and Single Shot Multibox Detector (SSD) in the detection of diseases and pests in tomato plants using images captured in-place by camera devices with various resolutions. Deep learning meta-architectures demonstrated that 9 different categories of plant and pests diseases including complex Intra and inter-class can be identified.

(Tm et al., 2018) presented a way of detecting tomato leaf disease using convolutional networks. In this study, Prajwala used a slight variation of the convolutional neural network called LeNet to detect and identify diseases in tomato leaves. In this study, the accuracy of 94 to 95% was achieved in detecting tomato leaf diseases.

(Wallelign, 2017) investigated the early detection of diseases in Soybean plant disease using a convolution neural network. In this study Lenet architecture was used to perform the soybean plant disease classification. Walleligh et al managed to achieve 99.32% classification accuracy in disease detection of soybean.

(Ramcharan et al., 2017) researched the deep learning approach for image-based cassava disease identification. Convolutional systems were utilized to distinguish three diseases and two types of pest damage. The best-trained model accuracies were 98% for darker leaf spot (BLS), 96% for red parasite harm (RMD), 95% for green bug harm (GMD), 98% for cassava dark coloured streak infection (CBSD), and 96% for cassava mosaic malady (CMD).

(Amara, Bouaziz and Algergawy, 2017b) investigated the detection of banana diseases (banana Sigatoka, banana speckle) using convolutional neural networks LeNet architecture. (Amara, Bouaziz and Algergawy, 2017b) demonstrated the effectiveness of classification of the images that have different resolutions, size, pose orientations complex backgrounds.

Kernamy et al,2018 demonstrated the use of deep learning the classifying of images of age-related macular degeneration and diabetic macular edema comparable to the performance of
human experts detection of the diseases with an accuracy of 96.6%. Kernamy et al., 2018 also demonstrated the diagnosis of pediatric pneumonia using chest X-ray images.

Relatively there is very little published work on the detection and diagnosis of diseases in livestock using convolutional neural networks. However, extensive research has been conducted on plants and humans.

2.4. Convolution

Convolution is the central concept behind the convolution neural network (CNN). Convolution is a mathematical operation that combines two sources of information (element-wise multiplication between two matrices) to produce new sources of information. Specifically, it involves the multiplication of the matrices one known as the kernel to the input tensor to produce sources of information known as feature maps. (Fandango, 2018).

Convolution mathematical function can be written as:

\[
(f * g)(n) = \sum_{m} f(m)g(n - m)
\]

The convolution operation is commutative in nature.

\[
(f * g)(n) = \sum_{m} f(n - m)g(m)
\]

For example, if Image A convolves with the kernel filter K and a feature map F is produced. The mathematical expression is given by the following equation.

\[
F(i, j) = (A * K)(i, j) = \sum_{m} A(m, n) \sum_{n} K(i - m, j - n)
\]

The equation can be written as because of the commutative in nature.

\[
F(i, j) = (A * K)(i, j) = \sum_{m} A((i - m, j - n)) \sum_{n} K(m, n)
\]

![Figure 1 Convolution Operation](image)
2.4.1. Convolutional neural networks

Yann LeCunn is the pioneer of Convolutional neural networks in which he builds the first CNN called Lenet in 1988 which was used for character recognition tasks like reading zip codes and digits. Convolutional Neural Networks are inspired by biological visual cortex. The visual context has small regions of cells that are sensitive to specific regions of the visual field as shown in figure 2. For example, some neurons fire when there are exposed to vertical edges, some when shown diagonal edges. This concept forms the basis of ConvNets (Wani et al., 2020).

Convolutional Neural Networks is one of the most fascinating architecture in deep learning, especially how the model helps in learning the keep features. CNN has yielded remarkable results in image recognition in agriculture, automation, and robotics. Some applications use Computer Vision and CNN to solve complex tasks of plant disease identification (Kamilaris et al., 2017).

![Figure 2 Conceptual analog between real neurons A and artificial neurons B Lee et al 2017](image)

2.4.1. What is a Convolutional neural network?

CNN is a feed-forward network neural network that is generally used to analyse visual images by processing data with a grid-like topology. CNN is also known as a ConvNet. Convolutional neural networks consist of two stages, which are feature extraction, and the classification stage. Feature extraction consists of the convolutional layers, max-pooling and the classification stages which is composed of one or more fully connected layers followed by a softmax function.
2.4.2. How CNN recognises images?

![Figure 3: Conceptual analog between real neurons A and artificial neurons B Lee et al 2017]

Convolutional neural networks accepts images as an input and the pixel of the image as input in the form of arrays as shown by Figure 3.

2.4.3. Hidden layer

Feature extraction is done in the hidden layers by performing calculation and manipulation this is the part that somewhat variously revamps images until we get a few information that is easy to read for the neural system. This layer utilizes a matrix filter to play out a convolutional activity to make patterns in the image Convolution intends to coil or to twist so the data will be twisted around and alter it use the operation to detect the pattern example. They are numerous shrouded layers like the Convolution layer, ReLu layer, Pooling Layer that performs include extraction from an image. ReLu activation layer is applied to a convolution layer to get a rectified feature map of the image. The pooling layer utilizes multiple filters to recognize edges, corners of an image. The higher the number of hidden layers the deeper the network as shown in figure 4. Fully connected layer that identifies the object in an image.
2.4.4. Output layer

Different layers in convolutional neural networks
i. Convolutional layer
ii. ReLu Layer
iii. Pooling Layer
iv. Fully connected layer

2.5. Feature extraction

A convolutional layer has several filters that perform the convolutional operation. Every image is considered as a matrix of pixel values as shown in Figure 5. Different filters are chosen depending on the area of interest under investigation. Consider the following 5*5 image whose pixel value are only 0 and 1.

![Convolved feature](image)

**Figure 5: Convolved feature**

Filters deals with different parts of the images depending on the applications eg edges, corners
2.5.1. ReLu Layer

Once the feature maps are extracted, the next step will be to over to the ReLU layer. The ReLu Layer performs an element-wise operation and sets all the negative pixels to 0 and this introduces nonlinearity to the network. The output after the ReLu operation will be the rectified feature map.

2.5.2. Pooling layer

The rectified pooling feature map now goes through the pooling layer. Pooling is a downsampling operation that reduces the dimensionality of the feature map as shown in figure 6. In the Max pooling, we are looking at the maximum value from the rectified feature map. Pooling layer uses different filters to identify different parts of an image like edges, corners.

![Rectified feature map](image)

Max pooling with 2*2 filters and stride 2

Pooled feature map

Max (3,4,1,2)=4

Figure 6 Pooling layer

2.5.3. Flattening

Flattening is the process of converting all the resultant dimensional arrays from the pooled map feature into a single continuous linear vector as shown in figure 7. The pooled feature map is a 2*2 matrix and after flattening it is converted into 1*4 linear vector. The Flattening matrix from the pooling layer is fed as input to the fully connected layer to classify an image.

![Flattening](image)

Figure 7 Flattening
2.5.4. Fully connected layer

In a fully connected layer, each neuron from the previous layer is connected every neuron in the next layer and every value contributes to the prediction of how strongly a value belongs to a particular class. The output of a fully connected layer is fed into a classifier, which outputs class scores. Softmax and Support Vector Machine are some of the classifiers used in ConvNets. Softmax produces probabilities for each class whilst the support vector machine produces class scores and the one with the highest probability is treated as the correct class.

![Fully connected layer](image)

2.6. Classification

Classification in deep learning involves that data under consideration belongs to one type of class or another. Classification means to identify, categorise and recognise the class or label of the images under investigation, in a machine, learning classification falls under the arm of supervised learning where the training samples are labelled. In classification problems, the training dataset is provided with their features or inputs and also the outputs provided (Fandango, 2018). The training data set is then trained to learn the featured on the training data set and parameters of the model is computed. The training model is then used to find its labels on the new test data.

Classification can be binary or multiclass depending on the problem under investigation. Binary class means that there are two distinct labels to distinguish the class for example the cattle have foot and mouth or the cattle do not have foot and mouth. Multi-class means that the data under investigation has many classes for example if a study is looking at the investigation of the probabilities of the foot and mouth five classes, drooling, teat lesion, feet lesion, gum lesion and also tongue lesion. Figure 9 shows multi classifications of distinguishing animals from features on the ear e.g kangaroo, rabbit, wallaby and bandicot.
2.6.1. Activation functions

The activation function is used to change or transform the activation level of neurons into an output. The output of each Convolutional layer is fed into an activation function as shown in figure 10. Different types of activation functions change the output of our neuron, these are, sigmoid, tanh, softmax, Relu (rectified linear unit).

\[
z = b + \sum_{i=1}^{n} w_i x_i
\]

Figure 10  Activation functions

2.6.2. Sigmoid function

It is a commonly used function that gives between 0 and 1. It is commonly used in models in which we predict probability as an output. Sigmoid function squashes the input in the range between 0 and 1. It is also an S-shaped curved as shown in figure 11.

It is mathematically represented as ;
2.6.3. Tanh

The hyperbolic function that is s-shaped and its output range is from -1 to 1 as shown in figure 13. It is similar to the sigmoid curve however, Tanh maps negative functions correctly to the negative and also zero inputs to zero.

2.6.4. Softmax function (Exponential function)

Softmax function comes at the end of a classification network. The example in figure 14 for shows output vector of scores A B and C. The score in the diagram are not scored but logarithmic of likelihoods. They need to be turned in probabilities. The input vectors enter the softmax classifier and output vector in range of 0-1 (probabilities) and the output adds up to 1.
The sum of a softmax function adds up to 1. The softmax function is given by a mathematical expression in equation 6. Softmax is mostly used for multi-classification problems since it outputs probability distribution for the classes under investigation (Fandango, 2018).

\[ s(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \] \hspace{1cm} \text{(6)}

Softmax is used for multiclass classification whereas sigmoid is used for binary class classification. The softmax function is a generalization of the sigmoid function that converts an n-dimensional vector \( z \) of arbitrary real values to an n-dimensional vector \( \sigma(z) \) of real values in the range \((0, 1)\) that add up to 1 (Fandango, 2018).

2.6.5. Relu (Rectified Linear Unit)

Neural networks train much fast with Relu than with sigmoid and tanh (Wani et al., 2020). Relu computes activation function at threshold zero as shown in figure 15. It is mathematically represented as:

\[ f(x) = \max(0, x) \] \hspace{1cm} \text{(7)}

2.7. Optimization algorithms

The main objective of optimisation functions is to minimize the loss function or objective functions. Good objective functions make the objective function being close to zero as fast as
possible after a couple of epochs. Some of the most commonly used optimisation algorithms are Adam, RMS prop, Ada Delta.

2.7.1. RMS prop
RMS prop corrects the vanishing learning rate in the AdaGrad. It is an adjusted adaptation of the AdaGrad that discards of history from the past and present the exponential moving normally. The RMS prop optimiser works as follows:

The magnitude of all weights is set first;

The maximum and minimum permissible loads are set ∆max and ∆min, after each iteration, it indications of the present angle and past are the equivalent and increasing the learning rate by a factor of 1.2 i.e. ρ=ρ+1.2 the update becomes if the indication of the current gradient and the previous are different, at that point decline the learning by a factor of 0.5 i.e ρ=ρ-0.5

- if the sign of the current gradient and the previous gradient are different, then decrease the learning by a factor of 0.5 i.e ρ = ρ − 0.5

2.7.2. Adaptive Moment Estimation (Adam)
Adam is an adaptive optimisation method that utilises the blessings of AdaGrad and RMS prop. It is also comparable to RMS prop and AdaDelta in the experience that it saves an exponentially decaying average from the until now squared gradients. Typical batch sizes used in the optimises is from 50 to 256 and they are elements that are viewed in the resolution of one-of-a-kind batch sizes. Large batch sizes furnish greater accurate consequences, however, this comes at a fee of requiring high reminiscence for computation. Small batch sizes require a smaller mastering price to preserve the high variance in the estimate of the gradient. This will increase the training time due to the fact of the decreased learning rate.

2.8 Artificial Intelligence
The ability of a machine to imitate intelligent human behaviour. In the past years, image recognition using artificial intelligence (AI) with machine learning has dramatically improved and been increasingly applied to diagnostic imaging in various medical fields (Hirasawa et al., 2018). The medical fields include the detection and diagnosis of diabetic retinopathy, skin cancer detection, diagnostic in radiation oncology.
(Smith, 2019) highlighted that artificial intelligence will bring more quick applications will be to improve exactness data about what's going on the farms by improving what is being identified and estimated. (Sung, 2018) also alluded that advancement in artificial intelligence and big data will take into account precision agriculture, for example, yield checking, diagnosing bug bugs, estimating soil dampness, diagnosing harvest time, and observing harvest well-being status. Specifically, the Internet of things (IoT) will measure the temperature, dampness, and measure of daylight underway ranches, making it workable for remote control through cell phones. It won't just lift the production of the ranches yet additionally add to their worth. Thus Artificial Intelligence, Deep Learning and Machine learning will play a major role in the development of Industry 4.0. Figure 16 shows that artificial intelligence includes machine learning and also deep learning.

![Figure 15: Artificial Intelligence, Machine learning and deep learning](image)

In recent years, there has been an increased use of artificial intelligence in the different field technology with so much hype about self-driving cars, chats box and virtual assistant’s making human jobs under threat were most of the economic activity will be handled by robs or AI agents (Ramos et al., 2017). Artificial intelligence has been studied widely since started since the 1950s were researches explicitly thought that human intelligence can be surpassed by trained a model with large data set given rules for manipulating the data.
2.8.1. Machine learning

Application of AI that allows a system to automatically learn and improve from experience. Figure 16 shows that machine learning is a subset of artificial intelligence. Machine learning works with a huge amount of structured data but Deep learning algorithms can work with a huge amount of structured and unstructured data. Machine learning algorithms cannot perform complex operations. As the amount of data increases the performance of the Machine learning algorithm decreases so to make sure the performance of a model is good we need Deep learning. The advent of technology has enabled deep learning to take less computational time through the use of general-purpose computing (GPU) (Lee et al., 2017).

2.8.2. Deep Learning

Deep learning is a subfield of machine learning that deals with algorithms inspired by the structure of the brain. Deep learning is a representation learning approach that learns informative features directly from data (Zhou et al., 2017). The primary difference between deep learning and machine is that deep learning neural networks and it is suitable for handling large amounts of structured data and some other difference shown in figure 18. Machine learning the feature extraction is done by the Data Scientists manually whilst deep learning it is done automatically using neural networks.
Deep learning is a hierarchically based feature extractor so that initial few layers will extract some simple key features e.g. if we take the example of faces it will learn some edges and if we get into deeper layers it will learn some features such as nose, lips, and finally some object.

The word deep refers to how the successive layers learn about the features so that it can reproduce meaningful representations of the data. As the number of layers the increases the depth of the model increases.
Deep refers to the way to the learning of keys features from successive layers and increasing the representations of meaningful data. The more the number of layers the deeper the network and also network with fewer layers is referred to as shallow network as shown in figure 20. In recent years, deep learning has made a breakthrough in near-human image classification, speech recognition, handwriting transcription, improved machine translation, text to speech conversion, near-human autonomous driving, improved search results on the web, improved ad targeting, as used by Google, Baidu and Bing. (Chollet, 2018)

2.8.3. Neural Networks
The term neural network is a reference to neurobiology but some of the central concepts of deep learning were drawn from understanding the brain. A neural network is a computing model whose layered structure resembles the network structure of neurons in the brain, with layers of connected nodes. It can learn from data, so it can be trained to recognize patterns, classify data, and forecast future events. A neural network breaks down your input into layers of abstraction. It consists of an input layer, one or more hidden layers, and an output layer. The layers are interconnected via nodes or neutrons with each layer using the output of the previous layer as its input. Its main function is to receive a set of inputs, perform calculations and then use the output to solve the problem.

2.8.4. How deep learning works?
Deep learning is used in mapping input feed into a sequence of layer transformation and in these layer transformations, key features from the input are learned by adjusting weights in the
layers so that the network can correctly map the input to their associated output targets as shown in figure 21. Learning thus can be defined as a process of finding the right weights adjustments in the layers such that the input can be correctly mapped to the associated values. A deep learning network contains many parameters and tuning the parameters so that the input is correctly mapped to the associated target is an iterative process (Wani et al., 2020).

![Diagram of a neural network parametrized by its weights](image1)

**Figure 20:** A neural network parametrized by its weights

![Diagram of a loss function](image2)

**Figure 21:** A loss function measures the quality of network output

It is of paramount importance to control the output of a neural network by measuring how far this output is far from what is expected as shown in figure 22. This is done loss function of the network. The loss function takes predictions of the network and the true target (what you wanted the network to output) and computes a distance score, capturing how well the network has done.

This score is used as feedback into the network to adjust the weights in the direction that will lower the score. This adjustment is done by the optimizer which implements backpropagation. (Chollet, 2018) optimizers determine how the learning proceeds in a network. Figure 23 how back propagation in a network.
There is a training loop that is created by backward propagation through several iterations conducted in the network weights will be updated in the direction that blows the loss score. A network with a low score output is close to the target in a trained network. (Chollet, 2018). Different types of loss functions are used in deep learning functions e.g. mean squared error (L2), cross loss entropy.

2.8.5. Mean squared error

It is used to calculate the mean average error of individually and is mathematically described by

$$ E = \frac{1}{n} \sum_{i=1}^{n} e_i^2 $$

$e_i$ represents the individual error of the $i$th output

$$ e_i = \text{target}(i) - \text{output}(i) $$
During their training process, the loss function is used to calculate the error at the output layers and its derivative gradient propagated back into the network.

2.8.6. Cross loss Entropy

It is loss function mostly used in regression and classifications problems. It is mathematically represented by:

\[ H(y) = -\sum_i y_i' \log(y_i) \]

Where \( y_i \) is the target label and \( y_i' \) is the output of the classifier. It is mostly used when the output is a probability distribution thus is mostly used with the softmax classifier. (Wani et al., 2020).

2.8.7. Bench making datasets

A huge dataset is required to get better performances in image classifications. However, in some applications like the medical field, it is difficult to get datasets due to privacy issues. Different institutions have been collecting data and designing their datasets which can be utilised in different applications these are ImageNet, Canadian Institute for Advanced Research, CIFAR-10, CIFAR 100, Open Images, Caltech 101 and Caltech 256, Stanford Dog dataset and MNIST. Preparing these datasets is labouring, time-consuming and these datasets have made deep learning to yield remarkable results in different applications.

2.8. ImageNet:

ImageNet: With more than 14 million hand-commented on high-goals hued pictures spreading over 20,000 classifications, this is the best quality level visual dataset. It was intended for use in visual item recognizable proof assignments by the software engineering division at Princeton College in 2009. From that point forward, this dataset (in its cut adaptation of 1,000 non-covering classes) has been utilized as the premise of the ImageNet Enormous Scope Visual Acknowledgment Challenge (https://arxiv.org/abs/1409.0575).

2.9.1. 80 Million Tiny Images dataset:

As the name proposes, this MIT dataset contains 80 million pictures gathered from the web and labelled to more than 75,000 distinctive non-theoretical English nouns. This dataset additionally shapes the reason for different other broadly utilized datasets, including the CIFAR datasets.
CIFAR-10: Created by the Canadian Institute for Advanced Research, CIFAR-10 is one of the most generally utilized datasets for AI (ML) exploration. This dataset contains 60,000 low-goals pictures crossing across 10 non-covering classes. CIFAR-100 from a similar research gathering, this dataset contains 60,000 pictures equally spread across 100 distinct classes.

Caltech 101 and Caltech 256: These datasets contain clarified pictures spreading over 101 and 256 classifications individually. Caltech 101 contains around 9,000 pictures, while Caltech 256 contains near 30,000. Stanford Dog dataset, this is a fascinating dataset explicit to various dog breeds. It contains 20,000 colour pictures spreading over 120 diverse dog breeds. MNIST, One of the most popular visual datasets ever, MNIST has become the de facto, Hello, World dataset for ML lovers. It contains more than 60,000 hand-named digits (zero to nine).

2.9.2. Transfer learning

Transfer learning is the application of skills, knowledge and or attitudes that were learned in one situation to another learning situation. (Perkins, 1992). Human beings use transfer learning since they were born e.g if you ride a bicycle it is easier for you acquired to use skills from a bike to ride a motorcycle

(Kim et al, 2015) highlighted that it is very difficult to achieve impressive performance when you have a small dataset thus to overcome these limitation transfer learning methods are used Convolutional can be trained from scratch, feature extraction from a pre-trained network and also fine-tuning from a pre-trained model.

Training from scratch a CNN model directly extracts features from raw images. During the training, the process is related features are extracted. Feature extraction is mostly used in computer vision tasks such as classification, object detection, and recognition. (Krishna & Kalluri, 2019)

Training of huge datasheet requires large computational power and also takes a lot of time to train the network up to several weeks. (Fandago, 2018) also highlighted that without transfer learning it will difficult to train very huge models as this would require several days or even months.

Train the new network with pre-trained weights can speed up the learning process (Krishna & Kalluri, 2019). In deep learning, some architectures are trained with a large volume of data and learn model weight and bias during training. Pre-trained architecture can be used to
train other networks by transferring weights to test network models. (Pan, SinnoJialin, & Qiang Yang, (2010)). Pre-Trained models have been used in object detection, text translation and also machine translation (Fandago, 2018). Training a new network with pre-trained weights can accelerate the learning procedure. (Krishna and Kalluri, 2019).

ImageNet is a picture database sorted out as indicated by the WordNet chain of importance and they are 100k synset and 1,000 human-commented on images for each synset. ImageNet just stores the references to the images while the images are put away on the web. In most deep learning research papers scientists train the model from the dataset discharged as a major aspect of the ImageNet's Large Scale Visual Recognition Challenge (ILSVRC) to characterize the dataset into 1,000 classifications. Contingent upon the sort of research 1000 number of classes can be changed to the necessary number of classes under scrutiny.

The most pre-trained classification models trained on the ImageNet-1K dataset are Alexnet, Visual Geometry Group (VGG) 16, VGG19, GoogleNet, MobileNet, ResNet and Lenet. Deep learning models like VGG, Google net, ResNet has been trained on the ImageNet data set to identify images and also additional layers can be added on top of these models to identify more complicated structures. Table 1 shows Pretrained models.

Table 1 Pre-trained Classification Models

<table>
<thead>
<tr>
<th>CNN Architecture</th>
<th>Year</th>
<th>Developed</th>
<th>Number of Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>LeNet</td>
<td>1998</td>
<td>YannLeCun et al</td>
<td>60000</td>
</tr>
<tr>
<td>Alexnet</td>
<td>2012</td>
<td>Alex K et al</td>
<td>62.3 million</td>
</tr>
<tr>
<td>VGGNet</td>
<td>2014</td>
<td>Simoyan, Zisserman</td>
<td>138 million</td>
</tr>
<tr>
<td>Google</td>
<td>2014</td>
<td>Google</td>
<td>4 million</td>
</tr>
<tr>
<td>ResNet</td>
<td>2015</td>
<td>Kaiming He</td>
<td>25 million</td>
</tr>
</tbody>
</table>

2.9.3. AlexNet:

This is the system that can be credited for opening the conduits. Planned by one of the pioneers of deep learning, Geoffrey Hinton and group, this system decreased the best five blunder rate to simply 15.3%. It was likewise one of the primary models to use GPUs for accelerating the learning procedure.
2.9.4. **VGG-16:** 
The system from Oxford's Visual Geometry Group is a standout amongst other performing structures, broadly utilized for benchmarking different plans. VGG-16 uses a straightforward design dependent on 3 x 3 convolutional layers stacked one over the other (multiple times), trailed by a maximum pooling layer to accomplish imposing execution. This model was prevailing by a marginally progressively complex model named VGG19.

2.9.5. **Inception:** 
Inception model is also called GoogleNet, this system was presented in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2014, and accomplished a main five blunder pace of 6.67%. It was one of the principal models to accomplish close human execution. The oddity behind this system was the utilization of a commencement layer, which includes the connection of various measured parts at a similar level.

2.9.6. **ResNet:** 
Introduced by Microsoft Research Asia, the residual network (ResNet) was a novel architecture utilizing batch normalization and skipping connections to achieve a top-five error rate of just 3.57%. ResNet was introduced as an innovative solution to the “vanishing gradient” problem. Neural train the network via backpropagation, while relying on the gradient descent, moving down the loss function to find the weights that minimize it. If there are too many layers, repeated multiplication makes the gradient smaller and smaller, until it “disappears”, causing performance to saturate or even degrade with each additional layer.

ResNet stacks up identity mappings, layers that initially don’t do anything, and skips over them, reusing the activations from previous layers. Skipping initially compresses the network into only a few layers, which enables faster learning. Then, the network trains, all the layers are expanded and the residual parts of the network explore more and more feature space of the source image. ResNet model has 152 layers anas it is deeper than the VGG model.

2.9.7. **MobileNet:** 
This is one of the models that are different from others in the sense that it focuses on the amount of power consumed by the mobile devices whilst, other models are completing in the race of outperforming each other. MobileNet has also been developed by Google to cater to mobile
devices, embedded systems and it has attributes of consuming low power and latency to provide a better experience of the devices of the algorithms. The network utilises the depthwise separable convolutions thereby reducing the number of parameters required to train the model.

There are open-source deep learning libraries that can be used in the training of datasets e.g. Caffe, Microsoft Cognitive Toolkit CNTK, Tensorflow, Theano, and Torch. Models trained on the ImageNet dataset can detect and capture some of the universal features such as curves, edges, and shapes. Some of the features apply to other kinds of datasets. Thus in transfer learning, such a universal model is used in extracting features in the new dataset.

Models trained on large and diverse datasets like ImageNet can detect and capture some of the universal features such as curves, edges, and shapes. Some of these features are easily applicable to other kinds of datasets. Thus, in transfer learning, we take such universal models and use some of the following techniques to fine-tune or retrain them to our datasets.

2.9.8. Feature extraction:

There are two ways to use a pre-trained network: feature extraction and fine-tuning. In Feature extraction, repeal and replace the last layer—This technique is mostly used when the new dataset is almost similar to the one the model was trained on, thus the need only the last layer to be retrained.

Depending on the number of classes of the application under investigation the last layer of a model trained from the ImageNet dataset with 1000 classes can be repealed and replaced with the number of a class under investigation.

The common practice for feature extraction is freezing the layers of a model and then add the last layer that is used for the classification of the image. Freezing the layers means preventing weights from being updated during training. If this is not done the pre-trained model will be changed during training and because of backpropagation the weights from added last layer (dense) will be propagated into the model destroying the previously learned features. The middle diagram in Figure 25 shows a VGG16 model as a feature extraction with some layers frozen.

Fine Tuning. This involves unfreezing some of the top layers used for feature extraction and adding a classifier to the fully connected classifier. Fine-tuning is important as sometimes the
model might have not been previously trained on the problem under investigation thus the need for the model to learn key features from the new images. It also involves turning the hyperparameters (filter size, number of filters, strides, zero paddings) and also changing the loss function, optimizer or changing the learning rate.

Figure 24: Showing a VGG Pretrained model
2.9. Deep learning architectures

There are different deep learning architectures used for various applications these are Lenet-5, AlexNet, Inception V3, ResNet, VGG16, VGG19, MobileNet, and Xception.

Lenet-5-Lenet-5 has 5 layers 3 Convolutional and 2 fully connected layers as shown in figure 26. The engineering layered establishment fundamentals of neural frameworks of stacking convolutions and pooling layers and ending with a fully connected layer. This structure has around 60,000 parameters.

![Architecture diagram for Lenet5](image)

2.10.1. Alex net

The rectified linear unit is used to solve the vanishing gradient problem and Alex was the first network in which the ReLU activation function was used. Alex is composed of many layers, which consist of convolutional, max pooling and fully connected layers. Table 2 shows the layers of the Alex architecture.

<table>
<thead>
<tr>
<th>Layer Name</th>
<th>Input Size</th>
<th>Filter Size</th>
<th>Window Size</th>
<th>Filters</th>
<th>Stride</th>
<th>Padding</th>
<th>Output Size</th>
<th>Feature Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>224x224</td>
<td>11x11</td>
<td>-</td>
<td>96</td>
<td>4</td>
<td>1</td>
<td>55x55</td>
<td>96</td>
</tr>
<tr>
<td>Max Pooling 1</td>
<td>55x55</td>
<td>-</td>
<td>3x3</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>27x27</td>
<td>96</td>
</tr>
<tr>
<td>Conv2</td>
<td>27x27</td>
<td>5x5</td>
<td>-</td>
<td>256</td>
<td>1</td>
<td>2</td>
<td>13x13</td>
<td>256</td>
</tr>
<tr>
<td>Max Pooling 2</td>
<td>27x27</td>
<td>-</td>
<td>3x3</td>
<td>384</td>
<td>1</td>
<td>1</td>
<td>13x13</td>
<td>384</td>
</tr>
<tr>
<td>Conv3</td>
<td>13x13</td>
<td>3x3</td>
<td>-</td>
<td>384</td>
<td>1</td>
<td>1</td>
<td>13x13</td>
<td>384</td>
</tr>
<tr>
<td>Conv4</td>
<td>13x13</td>
<td>3x3</td>
<td>-</td>
<td>384</td>
<td>1</td>
<td>1</td>
<td>13x13</td>
<td>384</td>
</tr>
<tr>
<td>Conv5</td>
<td>13x13</td>
<td>3x3</td>
<td>-</td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>13x13</td>
<td>256</td>
</tr>
</tbody>
</table>
Max Pooling 3

<table>
<thead>
<tr>
<th>Layer</th>
<th>Input Size</th>
<th>Filter Size</th>
<th>Window Size</th>
<th>Filter s</th>
<th>Stride</th>
<th>Padding</th>
<th>Output Size</th>
<th>Feature Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Connected 1</td>
<td>4096 neurons</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fully Connected 2</td>
<td>4096 neurons</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fully Connected 3</td>
<td>1000 neurons</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Softmax</td>
<td>1000 classes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 26: Architecture diagram for AlexNet

2.10.2. VGGNet

(Simonyan and Zisserman, 2015) introduced the VGG Network in their paper titled Very Deep Convolutional Networks for Large Scale Image Recognition. Table 3 and Figure 27 shows the details of the VGGNet 16 architecture.

Table 3: Details of various layers of VGGNet-16

<table>
<thead>
<tr>
<th>Layer Name</th>
<th>Input Size</th>
<th>Filter Size</th>
<th>Window Size</th>
<th>Filter s</th>
<th>Stride</th>
<th>Padding</th>
<th>Output Size</th>
<th>Feature Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>224*224</td>
<td>3*3</td>
<td>-</td>
<td>64</td>
<td>1</td>
<td>1</td>
<td>224*224</td>
<td>64</td>
</tr>
<tr>
<td>Conv2</td>
<td>224*224</td>
<td>3*3</td>
<td>-</td>
<td>64</td>
<td>1</td>
<td>1</td>
<td>224*224</td>
<td>64</td>
</tr>
<tr>
<td>Max Pooling 1</td>
<td>224*224</td>
<td>-</td>
<td>2*2</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>112*112</td>
<td>64</td>
</tr>
<tr>
<td>Conv3</td>
<td>112*112</td>
<td>3*3</td>
<td>-</td>
<td>128</td>
<td>1</td>
<td>1</td>
<td>112*112</td>
<td>128</td>
</tr>
<tr>
<td>Conv4</td>
<td>112*112</td>
<td>3*3</td>
<td>-</td>
<td>128</td>
<td>1</td>
<td>1</td>
<td>112*112</td>
<td>128</td>
</tr>
</tbody>
</table>
Max Pooling 2  112*11 2  2*2  2  0  56*56  128  

<table>
<thead>
<tr>
<th>Conv5</th>
<th>56*56</th>
<th>3*3</th>
<th>-</th>
<th>256</th>
<th>1</th>
<th>1</th>
<th>56*56</th>
<th>256</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv6</td>
<td>56*56</td>
<td>3*3</td>
<td>-</td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>56*56</td>
<td>256</td>
</tr>
<tr>
<td>Conv7</td>
<td>56*56</td>
<td>3*3</td>
<td>-</td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>56*56</td>
<td>256</td>
</tr>
<tr>
<td>Max Pooling 3</td>
<td>56*56</td>
<td>-</td>
<td>2*2</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>28*28</td>
<td>256</td>
</tr>
<tr>
<td>Conv8</td>
<td>28*28</td>
<td>3*3</td>
<td>-</td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>28*28</td>
<td>512</td>
</tr>
<tr>
<td>Conv9</td>
<td>28*28</td>
<td>3*3</td>
<td>-</td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>28*28</td>
<td>512</td>
</tr>
<tr>
<td>Conv10</td>
<td>28*28</td>
<td>3*3</td>
<td>-</td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>28*28</td>
<td>512</td>
</tr>
<tr>
<td>Max Pooling 4</td>
<td>28*28</td>
<td>-</td>
<td>2*2</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td>14*14</td>
<td>512</td>
</tr>
<tr>
<td>Conv11</td>
<td>14*14</td>
<td>3*3</td>
<td>-</td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>512</td>
</tr>
<tr>
<td>Conv12</td>
<td>14*14</td>
<td>3*3</td>
<td>-</td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>512</td>
</tr>
<tr>
<td>Conv13</td>
<td>14*14</td>
<td>3*3</td>
<td>-</td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>512</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Fully Connected 1</th>
<th>4096 neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fully Connected 2</td>
<td>4096 neurons</td>
</tr>
<tr>
<td>Fully Connected 3</td>
<td>4096 neurons</td>
</tr>
<tr>
<td>Softmax</td>
<td>1000 classes</td>
</tr>
</tbody>
</table>

**Figure 27:** Architecture diagram for VGGNet-16

### 2.10.3. Pretrained VGG 16

A VGG 16 network has 16 layers in which weights can be trained. There are two ways to use a pre-trained network that is as a feature extractor or fine-tuning. Feature extraction involved using learned features from the previous network and it identifies key interesting features that will run through a new classifier, which is trained, from scratch. In the VGG Model instead of
using the Image Net classifier with 1000 classes, you can use your own classifier with the number of classes you want. It is advisable that if your new dataset differs from the data trained on the original model you must use the first few layers of the model to do the feature extraction rather using the entire convolutional base (Colleta, 2018). Figure 28 shows the VGG16 model architecture.

Figure 28: VGG-16 Model Architecture

VGG as a fixed image extractor

In transfer learning that are different scenarios in which it can be used depending on how large is your dataset.

i. If a small dataset: fix all weights (treat CNN as a fixed feature extractor) and retrain the only the classifier.

ii. If you have a medium-sized dataset, “fine-tune ” instead: use the old weights as initialization, train full the network or only some of the higher layers

iii. If you have a large medium dataset, “fine-tune” instead: use the old weights as initialization, train full the network or only some of the higher layers

Cheng et al, 2016 investigated transfer learning with convolutional neural networks for the classification of abdominal ultrasound. In the study, 185 abdominal studies were categorized into 11 classes. The pre-trained model of VGGNet was used and weights in the convolutional layers were frozen to serve as feature extractors. The VGG net achieved an accuracy of 77.9% on the test images out of the (1109/1423 images).

Huynh et al, 2016 investigated transfer learning with deep convolutional neural networks using digital mammographic tumor. In the study, 619 lession images were used and a comparison
was conducted using support vector machines based on the CNN extracted image features to
distinguish benign and malignant breast lesion.

2.10.4. Google Net (Inception)

(Szegedy et al., 2015) developed this architecture in a bid to increase the computing resources in a network. The deeper the network the more computing resources are required. This was achieved by increasing the depth and width of a network whilst keeping the computer resources constant. However increasing the depth and the width comes with two drawbacks, the bigger the size the larger the number of parameters and also it will be prone to overfitting and also more computer resources are required. Google LeNet architecture has 22 layers deep network. (Szegedy et al., 2015) designers of Google Net aimed for efficiency and practicality. The resultant benefit of the GoogleNet architecture was that it was 12 times fewer parameters than AlexNet and also significantly accurate than Alex Net. Lower memory use and lower power use made Google Net more suitable for mobile device use.

GoogleNet architecture is a combination of all the convolutions the 1X1, 3X3, 5X5 as input to the next stage instead of picking up them as one by one. Since max-pooling has been successful in the previous architectures, it suggested that adding max-pooling in parallel. In images, correlations tend to be local so we cluster the neurons simply by taking convolutions into local patches. Local clusters can be covered by 1X1 convolutions; also, more spread out clusters by 3X3 clusters and lastly cover even more spread out clusters by 5X5 convolutions as shown in figure 29. All the convolutions are concatenated on the same patch.

![Figure 29: Heterogeneous set of convolutions to cover spread out](image)
Conceptually the inception module is simply the concertation between three convolutions scales a 1X1 convolution, 3X3 convolution, 5X5 convolution and also a 3X3 max pooling.

The practical benefits of using multiple convolutions on a single patch are that visual information is being processed at several scales and then aggregated for the next stage. This improves the discriminatory power of the network. However, there is a big problem that the network has conceived and increased computation several-fold when there is a number of features maps. Thus, 1X1 convolutions used to reduce the dimensionality before the expensive 5X5 convolution and 3X3 convolutions as shown in figure 31 and 32. The inception module gives a controlled option and or computer resources and speed resulting in networks that can be 3 to 10X faster. (Szegedy et al., 2015). Table 4 show the parameters of the layers of GoogleNet architecture.
Figure 32: Inception module with dimension reductions

Table 4: Details of various GoogleNet (Inception)

<table>
<thead>
<tr>
<th>Layer Name</th>
<th>Input Size</th>
<th>Filter Size</th>
<th>Window Size</th>
<th>Filters</th>
<th>Stride</th>
<th>Padding</th>
<th>Output Size</th>
<th>Feature Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution</td>
<td>224*224</td>
<td>7*7</td>
<td>-</td>
<td>64</td>
<td>2</td>
<td>2</td>
<td>112*112</td>
<td>64</td>
</tr>
<tr>
<td>Max Pool</td>
<td>112*112</td>
<td>3*3</td>
<td>-</td>
<td>2</td>
<td>0</td>
<td></td>
<td>56*56</td>
<td>64</td>
</tr>
<tr>
<td>Convolution</td>
<td>56*56</td>
<td>3*3</td>
<td>-</td>
<td>192</td>
<td>1</td>
<td>1</td>
<td>56*56</td>
<td>192</td>
</tr>
<tr>
<td>Max Pool</td>
<td>56*56</td>
<td>3*3</td>
<td>192</td>
<td>2</td>
<td>0</td>
<td></td>
<td>28*28</td>
<td>192</td>
</tr>
<tr>
<td>Inception 3a</td>
<td>28*28</td>
<td>-</td>
<td>-</td>
<td>28*28</td>
<td>-</td>
<td>-</td>
<td>28*28</td>
<td>480</td>
</tr>
<tr>
<td>Inception 3b</td>
<td>28*28</td>
<td>-</td>
<td>-</td>
<td>28*28</td>
<td>-</td>
<td>-</td>
<td>28*28</td>
<td>480</td>
</tr>
<tr>
<td>Max Pool</td>
<td>28*28</td>
<td>3*3</td>
<td>480</td>
<td>2</td>
<td>0</td>
<td></td>
<td>14*14</td>
<td>512</td>
</tr>
<tr>
<td>Inception 4a</td>
<td>14*14</td>
<td>-</td>
<td>-</td>
<td>14*14</td>
<td>512</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception 4b</td>
<td>14*14</td>
<td>-</td>
<td>-</td>
<td>14*14</td>
<td>512</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception 4c</td>
<td>14*14</td>
<td>-</td>
<td>-</td>
<td>14*14</td>
<td>512</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception 4d</td>
<td>14*14</td>
<td>-</td>
<td>-</td>
<td>14*14</td>
<td>512</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception 4e</td>
<td>14*14</td>
<td>-</td>
<td>-</td>
<td>14*14</td>
<td>512</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max Pool</td>
<td>14*14</td>
<td>3*3</td>
<td>2</td>
<td>0</td>
<td>7*7</td>
<td></td>
<td>832</td>
<td></td>
</tr>
<tr>
<td>Inception 5a</td>
<td>7*7</td>
<td>-</td>
<td>-</td>
<td>7*7</td>
<td>832</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception 5b</td>
<td>7*7</td>
<td>-</td>
<td>-</td>
<td>7*7</td>
<td>832</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avg-Pool</td>
<td>7*7</td>
<td>7*7</td>
<td>0</td>
<td>1*1</td>
<td>1024</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dropout 40%</td>
<td></td>
<td>1024</td>
<td>1*1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Linear</td>
<td></td>
<td>1000</td>
<td>1*1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Softmax</td>
<td></td>
<td>1000</td>
<td>1*1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
2.10.5. Resnet

(He et al., 2016) highlighted that deep neural networks are difficult to train thus the investigation of a residual network in a paper titled Deep residual learning for image recognition. Accuracy increases as the depth of a network increases and it saturates when the network converges. However, if the depth is further increased the will start to degrade faster (Wani et al., 2020). Thus (He et al., 2016) proposed a method that provides an optimal deep network through a residual deep learning framework that lets the new layers fit in the residual mapping. The idea in ResNet is that to prevent the gradient from degrading information is let to flow through short connections to reach shallow layers. The shortcut connection does not add extra complexity nor extra parameters. (Atienza, 2018). The skip connections allow the gradient to be propagated to the earlier networks as shown in figure 34.
Given that $H(x)$ is to be fitted in a few stacked layers, where $x$ is the input into the layers. Residual learning is represented by the expression $F(x) = H(x) - x$. It alludes that residual mapping is easier to optimise than original (Wani et al., 2020). The input image of $224 \times 224$ RGB is passed through a stack of convolutional layers and filters of $3 \times 3$. In one of the configurations $1 \times 1$ convolution filters as a linear transformation to the input channels. Table 5 shows the residual network architecture parameters.

<table>
<thead>
<tr>
<th>Layer Name</th>
<th>Input Size</th>
<th>Filter Size</th>
<th>Window Size</th>
<th>Filters</th>
<th>Stride</th>
<th>Padding</th>
<th>Output Size</th>
<th>Feature Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conv1</td>
<td>224*224</td>
<td>7*7</td>
<td></td>
<td>64</td>
<td>2</td>
<td>2</td>
<td>112*112</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>112*112</td>
<td>3*3</td>
<td></td>
<td>64</td>
<td>2</td>
<td>0</td>
<td>56*56</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>56*56</td>
<td>3*3</td>
<td></td>
<td>64</td>
<td>1</td>
<td>1</td>
<td>56*56</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>56*56</td>
<td>3*3</td>
<td></td>
<td>64</td>
<td>1</td>
<td>1</td>
<td>56*56</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>56*56</td>
<td>3*3</td>
<td></td>
<td>64</td>
<td>1</td>
<td>1</td>
<td>56*56</td>
<td>64</td>
</tr>
<tr>
<td></td>
<td>56*56</td>
<td>3*3</td>
<td></td>
<td>64</td>
<td>1</td>
<td>1</td>
<td>56*56</td>
<td>64</td>
</tr>
<tr>
<td>Conv 3_x</td>
<td>56*56</td>
<td>3*3</td>
<td>3*3</td>
<td>128</td>
<td>2</td>
<td>1</td>
<td>28*28</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>56*56</td>
<td>3*3</td>
<td></td>
<td>128</td>
<td>1</td>
<td>1</td>
<td>28*28</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>56*56</td>
<td>3*3</td>
<td></td>
<td>128</td>
<td>1</td>
<td>1</td>
<td>28*28</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>56*56</td>
<td>3*3</td>
<td></td>
<td>128</td>
<td>1</td>
<td>1</td>
<td>28*28</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>56*56</td>
<td>3*3</td>
<td></td>
<td>128</td>
<td>1</td>
<td>1</td>
<td>28*28</td>
<td>128</td>
</tr>
<tr>
<td></td>
<td>56*56</td>
<td>3*3</td>
<td></td>
<td>128</td>
<td>1</td>
<td>1</td>
<td>28*28</td>
<td>128</td>
</tr>
<tr>
<td>Conv 4_x</td>
<td>28*28</td>
<td>3*3</td>
<td>3*3</td>
<td>256</td>
<td>2</td>
<td>0</td>
<td>14*14</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>14*14</td>
<td>3*3</td>
<td></td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>14*14</td>
<td>3*3</td>
<td></td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>14*14</td>
<td>3*3</td>
<td></td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>14*14</td>
<td>3*3</td>
<td></td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>14*14</td>
<td>3*3</td>
<td></td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>14*14</td>
<td>3*3</td>
<td></td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>256</td>
</tr>
<tr>
<td></td>
<td>14*14</td>
<td>3*3</td>
<td></td>
<td>256</td>
<td>1</td>
<td>1</td>
<td>14*14</td>
<td>256</td>
</tr>
<tr>
<td>Conv5_x</td>
<td>14*14</td>
<td>3*3</td>
<td>3*3</td>
<td>512</td>
<td>2</td>
<td>0</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td></td>
<td>7*7</td>
<td>3*3</td>
<td></td>
<td>512</td>
<td>1</td>
<td>1</td>
<td>7*7</td>
<td>512</td>
</tr>
<tr>
<td>Avg-pooling layer</td>
<td>7*7</td>
<td>-</td>
<td>7*7</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>1*1</td>
<td>1000</td>
</tr>
</tbody>
</table>
2.10.6. DenseNet

(Huang et al., 2017) investigated the use of Densely Connected Network in which later is related to each other in a feed-forward plan. Ordinary a convolution arrange with L layers have L affiliations however in DenseNet sort out there are \((L(L+1))/2\) direct associations. In DenseNets maps of all the past layers are used as wellsprings of information and its part maps into the going with layers. The preferred position of DenseNet is that it deals with the issue of vanishing tendency, stimulate feature reuse and it improves the movement of information, gradient all through the framework which makes them less difficult to train. This is made
possible considering the way that every layer has a direct relationship with the hardship work and besides the commitment of the first (Huang et al., 2017).

DenseNet consists of a series of Dense Blocks and in between each block, there is a convolution +Pooling layer as shown in figure 36. DenseNet combines all the feature maps with connection operation and each layer is related to the other layers which maximise the flow of information. This solves the problem of vanishing gradient in the deep network, strengthens feature propagation, encourages feature re-use and reduces model parameters (Huang et al., 2017). DenseNet network consists of the two basic structures the Dense block and transition layer. Each convolution layer in the dense block received the output of all previous layers as input and passes the output as the subsequent convolution layer as shown in figure 37.

The connection of a network can be expressed by the following equation

\[ x_l = H_l([x_0, x_1, x_2, \ldots, x_{l-1}]) \]

In the equation above, \( lx \) speaks to the yield of layer \( l \). \([x_0, x_1, x_2, \ldots, x_{(l-1)}]\) indicates that 0, 1,... \( l-1 \) layer feature maps are associated with the channel measurement.

\( H_l(\cdot) \) is the excitation capacity of the main layer. It is a non-linear change, which is a blended activity, including a progression of BN (Batch Normalization), ReLU, Pooling and convolution tasks. The letter \( k \) speaks to the width of the convolution layer, which is the number of channels.
of the trademark chart yield by every convolution layer in a thick square. All dense blocks yield a k-highlight map after the convolution of each layer, so it is important to utilize a change layer between two thick squares, normally made out of BN-ReLU-Conv-Pooling four sections, to accomplish the reason for lessening the size of the element map.

Since DenseNet makes full use of the yield data of every convolution layer, it can set fewer yield channels, thus reducing the quantity of preparing parameters. The recipe for computing the quantity of trademark channels $C$ after going through dense square layers is as per the following.

$$C_0 = N_H \times k + C_i$$

(Xu et al., 2019) proposed the utilization of DenseNet in the image classification of fundus clinical images. The consequences of the examination indicated that DenseNet image classification improved the exactness of the clinical image pixel order which valuable in the image classification.

(Yaohua and Xudong, 2019) proposed synthetic oil image recognition method using DenseNet. The model extracts the multi-features and improves the accuracy in the classifications and recognition before reusing them in the convolutions. After the denoising and removal of SAR images are inputted into the DenseNet model.

(Wu et al., 2019) investigated the cascaded fully connected DenseNet for automatic kidney segmentation in ultrasound sound images. This study was meant to solve problems of kidney ultrasounds images such as noise, severe redundancy information and low image contrast which makes it difficult to achieve image segmentation with a fully connected DenseNet layer. The study proposed a fine cascaded model which provides coarse image segmentation. The size of the kidney is very small as compared to the size of the ultrasound image thus there is more redundant information and if FC dense is used the segmentation is very low. (Wu et al., 2019) proposed a cascaded FC DenseNet to focus on the inner and the outer contours of the kidney.

(Hai et al., 2019) investigated the use of fully connected DenseNet with automated breast tumor segmentation. It was pointed out that the FC DenseNet connected was used to improve the precision of image segmentation for various sizes and shapes of tumor images without preprocessing and postprocessing.
For each layer, the feature maps of every single going before layer are utilized as sources of info, and its element maps are utilized as contributions to every single consequent layer. DenseNets have a few convincing favourable circumstances: they ease the disappearing gradient problem, reinforce include spread, energize highlight reuse, and considerably diminish the number of parameters. We assess our proposed design on four highly competitive article acknowledgement benchmark assignments (CIFAR-10, CIFAR-100, SVHN, and ImageNet). DenseNets get huge enhancements over the best in class on most of them, while requiring less calculation to accomplish elite. Table 6 shows the dense architecture parameters.

Figure 38: Five Layer Dense block

Figure 39: Architecture diagram for DenseNet
Table 6: Dense Architecture parameters

<table>
<thead>
<tr>
<th>Layers</th>
<th>Output Size</th>
<th>DenseNet-121</th>
<th>DenseNet-169</th>
<th>DenseNet-201</th>
<th>DenseNet-264</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution</td>
<td>112 × 112</td>
<td>7 × 7 conv, stride 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Pooling</td>
<td>56 × 56</td>
<td>3 × 3 max pool, stride 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dense Block (1)</td>
<td>56 × 56</td>
<td>1 × 1 conv 3 × 3 conv × 6</td>
<td>1 × 1 conv 3 × 3 conv × 6</td>
<td>1 × 1 conv 3 × 3 conv × 6</td>
<td>1 × 1 conv 3 × 3 conv × 6</td>
</tr>
<tr>
<td>Transition Layer (1)</td>
<td>56 × 56</td>
<td>1 × 1 conv</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dense Block (2)</td>
<td>28 × 28</td>
<td>1 × 1 conv 3 × 3 conv × 12</td>
<td>1 × 1 conv 3 × 3 conv × 12</td>
<td>1 × 1 conv 3 × 3 conv × 12</td>
<td>1 × 1 conv 3 × 3 conv × 12</td>
</tr>
<tr>
<td>Transition Layer (2)</td>
<td>28 × 28</td>
<td>2 × 2 average pool, stride 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dense Block (3)</td>
<td>14 × 14</td>
<td>1 × 1 conv 3 × 3 conv × 24</td>
<td>1 × 1 conv 3 × 3 conv × 32</td>
<td>1 × 1 conv 3 × 3 conv × 48</td>
<td>1 × 1 conv 3 × 3 conv × 64</td>
</tr>
<tr>
<td>Transition Layer (3)</td>
<td>14 × 14</td>
<td>2 × 2 average pool, stride 2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dense Block (4)</td>
<td>7 × 7</td>
<td>1 × 1 conv 3 × 3 conv × 16</td>
<td>1 × 1 conv 3 × 3 conv × 32</td>
<td>1 × 1 conv 3 × 3 conv × 32</td>
<td>1 × 1 conv 3 × 3 conv × 48</td>
</tr>
<tr>
<td>Classification Layer</td>
<td>1 × 1</td>
<td>7 × 7 global average pool</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Python programming language used.

Keras: is an easy to utilize neural system library based on Theano or TensorFlow that permits designers to model thoughts very rapidly (Chetlur et al., 2014). Keras gives the greater part of the structure squares expected to construct sensibly refined deep learning models. It additionally accompanies incredible documentation and huge amounts of online assets. It works with python. This structure was utilized alongside the arrangement of loads learned on a huge dataset, ImageNet [https://github.com](https://github.com/). OpenCV—OpenCV is a free library for both scholarly and business utilization that supports and Windows, Linux, Mac OS, iOS and Android, and C++, C, Python and Java interfaces. OpenCV was intended for computational proficiency and with a solid spotlight on continuous applications. Written in upgraded C/C++, the library can exploit multi-centre preparation [http://opencv.org](http://opencv.org).

2.10. Foot and Mouth

Foot-and-mouth Disease (FMD) is an exceptionally infectious livestock disease caused by the Foot and Mouth Disease Virus (FMDV; variety Aphthovirus, family Picornaviridae), (Control, 2012) and ('Foot and Mouth Disease (FMD)', 2015). FMD affects cloven-hoofed domestic and around 70 wild creature species for example African Buffalo, including cows, pigs, and little ruminants (Jamal and Belsham, 2013) and (Infection, 2018). Figure 40 shows the structure of the foot and mouth virus.
There are 7 distinct serotypes (O, A, C, Asia 1, SAT 1, 2, & 3) and there are some subtypes in each serotype. Serotypes O and A were named after their place origin Serotype) originated in the department in France called Oise whilst A from Allemagne in Germany. SAT was also named after a place of origin which is the Southern African Territory (SAT). (Jamal and Belsham, 2013).

The immunity against one serotype (and subtypes) may not protect the animals from other serotypes (and subtypes). Six serotypes have been identified in Africa from the 7 distinct serotypes these are, serotype O, A, C, SAT-1, SAT-2, and SAT-3, respectively, (WRLFMD, 2003–2013; Di Nardo et al., 2011). In East Africa serotypes O, A, SAT, SAT2and SAT 3 are common, whilst in West Africa serotypes O, A, SAT1 and; SAT 2 (Tekleghiorghis et al., 2016)and southern Africa has serotypes SAT-1, SAT-2, and SAT-3 (Brito et al., 2016) &(Babb and Emery, 2018)) as shown in figure 41. Animal importation has caused the spread of SAT1 and SAT 2 which were only common in Africa to other regions like the Middle East (Ahmed et al., 2012).
FAO reports that there has been a high prevalence of Foot and Mouth in many African, the Middle East, Asia, and South America. Europe, North and Central America, Pacific nations. (Knight-Jones, Mclaws and Rushton, 2017) noted that FMD endemic areas contain three-quarters of the world FMD susceptible livestock and the poorest livestock keepers. However,
There has been an improvement of FMD situation in Europe. (Babb and Emery, 2018)

In 2012 the Food Agriculture Organisation (FAO) and the World Organisation for Animal Health developed a 15 years Global Control Strategy to reduce FMD in endemic countries and also maintain the status of FMD free countries. However, it was noted that the endemic may not be eradicated within 15 years but significant progress in reducing the burden caused by FMD. The strategy was developed tapping from the experiences from different regions. (Babb and Emery, 2018)

The strategy includes three components:

a) improving global FMD control;

b) the strengthening of Veterinary Services;

c) the control of other Transboundary Animal Disease (TADs).

A country that is affected by FMD has been banned from the international trade of animals and animal products. In developing countries, the loss involves that of animals, biodiversity, food security and the livelihood of smallholder farmers.

Figure 42: Serotypes in different regions

Serotypes 7, 10 and 3 are in blue. Some cases have an indication of being in red and green.
FAO reports that FMD outbreaks were the disease has been previously eradicated cause losses of approximately USD 1.5 billion per year and also the cost burden in endemic regions is estimated to be more that USD 6.5 Billion a year. (Babb and Emery, 2018). Thus, reducing FMD in endemic countries by a coordinated control strategy at the global and regional level is of shared interest and should be considered a global public good.

2.10.1 Clinical signs of foot and mouth

The incubation period of FMD is between 2-12 days and when it is early stages animals can experience high fever with temperatures 104 -106 °F. Animals also develop blisters in the mouth (tongue, gum, lips ) which later rupture and leave ulcers. Blisters also develop on the teats and feet of animals (Aftosa, 2015).

The clinical signs of FMD include the formation of blisters in the mouth fever, lameness, and excessive salivation (ptyalism), reduced milk yield and fever. The disease is highly contagious that many cattle can be affected at the same time however; they may not show the same clinical signs. (Infection, 2018). However, confirmation of diagnosis can only be done after laboratory tests.

2.10.2 The spread of the disease

Foot and mouth disease is a transboundary disease in which the movement of animals plays a key role in the spread of the disease. Foot and mouth virus is found in the fluids from the vesicle, saliva, milk and cow dung (‘Foot and Mouth Disease Control Strategy for Great Britain’, 2011). Contamination of any of the secretions or excretions of the fluids poses a great danger to other animals. The environmental conditions also determine whether the virus will survive or not as heat and disinfectants kill the virus and cold and darkness keeps the virus alive.

There are several ways in which FMD can be spread through animals;

i. Direct contact with the animals
ii. Aerosols
iii. Contaminated farming equipment, tools, clothing, shoes, water
iv. Semen from infected animals
Intraspecies and interspecies transmission of FMDV

i. Wildlife is involved in the spread of FMDV. Mainly the African buffalo (Syncerus caffer) spreads the (SAT1-3) livestock husbandry varies from region to region depending on the grazing and watering points. Robinson et al. (2011) described the animal movement in Africa as:

ii. Total nomadism in which animals have no permanent residence and it is no activities of cultivation

iii. Semi nomadism in which animals have a permanent residence and cultivation is practised and cattle are herded to far away grazing lands

iv. Transhumance: a permanent place for the cattle exists and cattle are sent to good grazing lands seasonally. Partial nomadism this is when farmers live in the permanent settlement and leave the cattle to graze in the vicinity

The animal movement poses a risk of the contact between cattle and the African buffalo in areas that are near the game parks were animals can meet in the grazing lands, stock rivers, watering grounds. especially during the driest season (Jamal and Belsham, 2013).

2.10.3 Diagnosis of FMD

Currently, foot and mouth are diagnosed by laboratory tests. There are three types of laboratory tests: two detect the presence of virus (antigen Enzyme-linked immunosorbent assay (ELISA) and virus isolation and polymerase chain reaction (PCR) type tests) and one detects the presence of antibody produced by an infected animal in response to infection. (‘Foot and Mouth Disease Control Strategy for Great Britain’, 2011). The lateral flow immunochromatographic (LFI) test has been used in the detection of specific antibodies against FMDV non-structural proteins (Niedbalski, 2016). Neutralisation test is normally performed to detect the structure of the antibodies to the structural proteins and is mostly used import and certification of animal products (Edition, 2012).

The importance of detecting FMDV in the field cannot be overemphasised as (Niedbalski, 2016) highlighted the issue of time needed to take samples to the laboratory as early as 4 to 6 hours of confirmation of FMDV. Thus they are portable mobile platforms of real-time RT-PCR assays used for the detection of FMDV in the field the Cepheid SmartCycler Real-time PCR machine, Enigma FL and LightCycler Nano System. When a country is affected by foot and mouth they are multi challenges that can be faced costs, logistic challenges, repeated
vaccinations and enforcement of outbreak measured. The affected country will face trade restrictions to prevent the spread of the disease into other countries. (Infection, 2018).

2.10.4 Control of FMD
Foot and mouth control requires effective monitoring and early warning systems at local, regional. International cooperation plays also an important part in human capital development in regions lacking the required expertise in fighting the diseases. Continuous improvement of the Veterinarians is also an important component, which can be done through international cooperation. International bodies like FAO/OIE/WHO Global Early warnings help in the dissemination of information about FMD (Babb and Emery, 2018).

Vaccinations are also used in controlling the impact of foot and mouth in endemic countries. The Global Strategy report of 2018 reports that there are scarce vaccines in some developing countries thus support is given for the vaccines to meet the required quantities by OIE. The Global Strategy also highlights each country at the national level must have emergency response teams that deal with Foot and Mouth and at the international level, there is an FAO/OIE Crisis Management Centre –Animal Health (CMC-AH) which caters for emergency services.

Regional groups have developed coordinated Food and Mouth Control programs. The Southern African Development Community control programs have been reported that it has yielded good results however it has been noted that Cape Buffalo (Syncerus caffer) has been the reservoir for the SAT virus and provides a challenge for the management of the disease. (Tekleghiorghis et al., 2016)

2.10.5 Foot and Mouth Outbreaks in Zimbabwe
(Guerrini et al., 2019) carried out a study for the FMD outbreaks in Zimbabwe during the period 1931 to 2016. In the study major FMD drivers were also investigated which are;

i. Distance from the protected areas
ii. Seasons
iii. Water availability
iv. Political and economic
Figure 43: Foot and Mouth Outbreaks in Zimbabwe from 1931 to 2016 (Guerrini et al., 2019)

Figure 43 shows black dots which represent primary outbreaks and protected areas represented as grey areas (this is where African Buffalo resides). Figure 44 shows the numbers of outbreaks from the period 1931 to 2016.

Figure 44: Number of foot and mouth outbreaks during the period 1931 to 2016 (Guerrini et al., 2019)
2.10.6 FMD Control measures
The following control measures are important in reducing the spread of foot and mouth disease:

I. Early detection and reporting of the FMD to limit the spread of the disease

II. Quarantining of the infected animals at the premises where it was detected

III. Containing the spread of the disease by restricting the movement of the animals from the premises.

IV. Vaccination of cattle to eradicate the disease

V. Continuous surveillance in the FMD prone diseases

2.10.7 Foot and Mouth Detection
Early identification of animals affected with FMD is vital in preventing disease outbreaks and diagnosis and control. Animals infected with FMD have skin temperatures rising in the excess of 40°C which can be detected by palpation (Alexandersen et al, 2003). However (Gloster et al., 2011) noted that temperature changes are not conclusive and but helps in identifying animals that require closer examination to detect more definitive signs of signalling testing and laboratory tests.

Infrared thermography has also been reported to be used in detecting heat emitted from the cattle to detect those with the possibility of having FMD. (Schaefer et al., 2004) investigated the use of Infrared thermography in the detection of foot and mouth as a method of identifying animals infected with FMDV. Infrared thermography is a technique that has been used in measuring animal and human body temperatures. (Schaefer et al., 2004) focused on the detection of FMDV in calves by collecting thermographs of the side, hooves, facial areas, ears including the lachrymal gland and tear duct.
Figure 45: Infrared Detection of FMDV in calves (Schaefer et al., 2004) (Rainwater-Lovett et al., 2009) also investigated the use of Infrared Thermography (IRT) in the detection of FMDV before and after the clinical signs have been observed in cattle. In this study, IRT imaging revealed that foot temperatures of animal’s increases in infected animals and the cut off temperature was found to be 34.4 °C. Foot temperatures were chosen ahead of face temperatures as changes were easily observed see figure 45.

Figure 46: Digital and infrared images of cattle without (A) or with (B) fever and note that the lower temperatures (blue-green) in the animal without fever or viremia versus the higher temperatures (orange-red) in the viremic and feverish animal.
2.10.8 EuFmd

Eu Fmd has three pillars, which are to improve, reduce and promote the fighting and mouth

I. Improved readiness for FMD crisis management by members
II. Reduced risk to Members from the European neighbourhood: Progressive Control in neighbouring regions
III. Promote uptake of the global strategy for the Progressive control of FMD

2.11. Economic Impact of Foot and Mouth

Any infection among animals requires a monetary effect on the livestock keepers do as to contain the disease. These impacts can be classified into two groups; direct losses and indirect losses. Direct losses are associated with death, abortion; Indirect losses are linked to the costs associated with preventing the disease i.e vaccines, costs of quarantining animals. For some animal diseases, an accurate estimation of illness influence is problematic due to nonattendance of open data and the variability of creation structures used the world over.

In endemic countries, FMD affects the production of both livestock meats and also milk to both large and smallholder farmers. In Kenya, smallholder farmers account for 70% of the national output. (Knight-Jones, McLaws and Rushton, 2017). The long term effect in FMD endemic countries is reduced animal production and restriction is the trade on animal products (Jamal and Belsham, 2013).
In endemic nations, immediate, obvious FMD creation misfortunes fluctuate and have been estimated in various manners. Both huge also, little pig ranchers and steers property delivering milk are normally the most exceedingly terrible influenced. This influences national yield; in Kenya, smallholders represent 70% of milk creation (FAO, 2011).

In South Sudan, yearly misfortunes coming about because of diminished milk creation and mortality from FMD were evaluated at US$25 per head of cows in the populace (Barasa et al., 2008). In Pakistan flare-up, milk loss production for more than 60 days was put at US$100 per influenced lactating cow (Ferrari et al., 2014). In Turkey evaluated direct costs shifted from the US $152 per influenced dairy yearling to US$294 for an influenced lactating dairy bovine, and about US$200 per influenced creature for meat cows (Knight-Jones, McLaws and Rushton, 2017). € Family unit effect can be progressively significant when put as the level of salary. In Cambodia smallholders, encountering flare-ups had family unit misfortunes of about US$45, with low salary family units losing the biggest rate of salary (12% of yearly pay for the most unfortunate) (Shankar et al., 2012).

2.12. Evaluating classifier Predictions

There is a need to verify how a classifier used for a certain type of application is performing. Depending on the application different classifiers are used in machine learning, deep learning and also artificial intelligence. The confusion matrix and a classification report are used to describe how good a model.

2.13 Confusion Matrix

Confusion Matrix is utilised to portray the exhibition of a classification model. A confusion lattice is a basic method to format what number of anticipated classifications or classes were accurately anticipated and what number of were not. It is utilised to assess the after-effects of a prescient model with a class result that was effectively anticipated as their actual class. To comprehend what's happening inside this confusion matrix of right classes versus inaccurate classes, we first need to comprehend the true positives, true negatives, false positives.

Table 8 shows binary classification, which is the task of classifying the members of a given set of objects into two groups based on whether they have some property, or not. There are four possible outcomes from a binary classifier.
true positive (TP): predicted to be positive and the actual value is also positive.
false positive (FP): predicted to be positive but the actual value is negative.
true negative (TN): predicted to be negative and the actual value is also negative.
false negative (FN): predicted to be negative but the actual value is positive.

Table 7 Confusion matrix

<table>
<thead>
<tr>
<th>(Actual)</th>
<th>(Predicted)</th>
<th>(Predicted)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>True Positive</strong> (TP)</td>
<td><strong>False Positive</strong> (FP)</td>
</tr>
<tr>
<td></td>
<td><strong>False Negative</strong> (FN)</td>
<td><strong>True Negative</strong> (TN)</td>
</tr>
</tbody>
</table>

True Positive TP: cases when classifier predicted TRUE (they have the disease-Foot and Mouth) and the correct class was TRUE (cattle has the disease- Foot and Mouth).

True Negative TN: cases when the model predicted FALSE (no disease-Healthy) and the correct class was FALSE (cattle do not have the disease-Foot and Mouth).

False Positive FP: (Type I error): classifier predicted TRUE but correct class was FALSE (cattle did not have the disease).

False Negatives FN: (Type II error): classifier predicted FALSE (cattle do not have the disease-Foot and Mouth) but they do have the disease.

2.13.1 Accuracy
Determineing accuracy is one of the most common performance metrics used in classification. It is the proportion of the observations predicted correctly over the total number of samples.

Classification Accuracy = \[ \frac{\text{No of samples Predicted correctly}}{\text{Total No of samples}} \]  

\[ \text{Classification Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \]  

true positive rate: TPR = \[ \frac{\text{positive correctly classified}}{\text{total positives}} = \frac{TP}{P} \]  

false positive rate: FPR = \[ \frac{\text{negative incorrectly classified}}{\text{total negatives}} = \frac{FP}{N} \]

An accuracy performance indicator is very good when the data is balanced. When the data is imbalanced, it suffers that a model will have high accuracy but lacking predictive power.
For example, we are trying to predict a disease that occurs in a population of 0.1%. After the model has been trained, an accuracy of 95% is calculated. However, 99.9% of the population does not have the disease and we have created a model that predicts that nobody has the disease, the model will be 4.9% more accurate but it has less predictive power. Thus they are other metrics like precision, recall, F1 score which are used to solve the shortfalls of the classification accuracy.

2.13.2 Precision
Precision represents the portions that every observation predicted to be positive is positive. When the model predicted TRUE class how often was it right

\[
\text{Precision} = \frac{TP}{\text{Total True predictions}} = \frac{TP}{TP + FP}
\]

2.13.3 Recall
The recall represents the proportion that every truly positive observation. It measures the ability of a model to identify a positive class. When the class was TRUE, how often did the classifier get it right? When the recall is very high in a model it will have low power predicting observations in the positive class. (Albon, 2018)

\[
\text{Recall} = \frac{TP}{\text{Actual TRUE}} = \frac{TP}{TP + FN}
\]

- Number of cattle that are diseased with Foot and Mouth = TP + FN
- Number of cattle that are classified as diseased with Foot and Mouth (when they have Foot and Mouth )

2.13.4 F1 Score
F1 is used as a metric to quantify the performance evaluating multiclass classification on imbalanced data. It is the weighted harmonic mean between precision and recall. F1 Score shows the correctness of positive prediction (correct prediction for observation labelled as correctness)

\[
\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

2.13.5 Sensitivity and Specificity
The idea behind sensitivity and specificity is that these measures how well a screening test determines how whether someone or something has a condition. The key thing about sensitivity and specificity they are well known as conditional probabilities. This means the denominator
used to calculate the probability is not the entire group but instead the subset of the group in this case whether or not someone has the disease.

**Sensitivity** is the probability that the screening test is positive given that you have the disease. During testing we need to know how good is our classifier in identifying cattle with foot and mouth disease.

\[
\text{Sensitivity} = P(\text{Test} = \text{Positive}|\text{Foot and Mouth})
\]

\[
\text{Sensitivity} = \text{true positive rate}: \text{TPR} = \frac{\text{positive correctly classified}}{\text{total positives}} = \frac{tp}{tp+fn}
\]

**Specificity** is the probability that the screening test is negative given that you do not have the disease. During the testing we need to know how good our classifier in identifying cattle without foot and mouth.

\[
\text{Specificity} = P(\text{Test} = \text{Negative}|\text{Foot and Mouth} = \text{False})
\]

\[
\text{Specificity} = \text{true negative rate}: \text{FNR} = \frac{\text{negative correctly classified}}{\text{total negatives}} = \frac{tn}{fp+tn}
\]

Just like the hypothesis, testing screening tests can also have errors these errors are False Negative (FN) and False Positive (FP).

**False Negative (FN):** Probability that the test negative given that you have the disease

**False-positive FP:** Probability that the test is positive given that you do not have the disease

The symptom or a test is considered effective in predicting a disease when BOTH the sensitivity and the specificity are high.

0% specificity means that healthy cattle will be labelled as unhealthy (in our case meaning diseased with Foot and Mouth) as shown in figure 48.

100% Specificity means that all healthy cattle are labelled as healthy

0% Sensitivity means that all cattle with Foot and Mouth (unhealthy) are labelled as healthy

100% Sensitivity means that all cattle with Foot and Mouth (unhealthy) are labelled as diseased with Foot and Mouth (unhealthy)
2.13.6 Macro averaging

The measurements used to measure the performance of the model on the test data are micro-average precision and macro-average precision. Micro-averaging considers the whole data as a solitary total outcome. The individual genuine positive qualities and false-positive values are added to give accuracy.

$$ P_{\text{micro}} = \frac{TP_1 + TP_2 + \cdots + TP_k}{(TP_1 + \cdots + TP_k) + (FP_1 + \cdots + FP_k)} $$

$P_{\text{micro}}$ micro average precision

$TP_i$, True Positive count of class $i$

$FP_i$, False Positive count of class $i$

Micro averaging gives every observation an equivalent weight as opposed to giving each class an equivalent weight. This gives the classes with bigger observations more power. To give each class an equivalent weight, we utilize macro averaging.

Macro-averaging accuracy decreases the multi-class expectations to multiple binary predictions of all classes. Macro average precision is used to find out each class performs on a model within a dataset. Micro averaged precision gives an overview of the model over the entire test data.

$$ P_{\text{macro}} = \frac{P_{r_1} + P_{r_2} + \cdots + P_{r_k}}{k} $$

$P_{r_{\text{macro}}}$ - macro average precision

$P_{r_i}$ precision of the classifier for class $i$

$k$ number of classes
2.14.1. Receiver Operating Curve

The Receiver Operating Curve (ROC) is most commonly used in evaluating the performance of a binary classifier at various probability thresholds. ROC compares the presence of true positives to false negatives at every probability threshold. The True positives probabilities will be on the y-axis whilst the true negatives will be on the negative. A classifier that predicts every observation correctly is represented as a solid light grey line in Figure 49. In evaluating a model the closer to the solid line the better the model. The higher the curve the better the model and also the greater the area under the curve (Albon, 2018). The area under the curve evaluates the areas under the ROC curve and the closer is to one the better the model.

![Receiver Operating Characteristic](image)

Figure 49: Receiver operating Curve
CHAPTER THREE: METHODOLOGY

3.1. Introduction

This chapter discusses the specific methods used for the detection of Foot and Mouth. It describes the necessary steps that are conducted for the prediction so that it can be re-implemented or another researcher can improve or compare with another method. Transfer learning was applied to different deep learning architectures (VGG-16, InceptionV3, Inception Resnet, Densenet, and Xception) and an assessment was carried out looking at performance metrics and predicting capability. Figure 50 shows the approach that was used in the study of the Software Engineering model.

3.2. Systems Engineering V System model

![Software Engineering V Model Diagram]

3.2.1. Steps in the detection of Foot and mouth

i. Defining the mission requirements
ii. Getting familiar with the Foot and Mouth disease signs and symptoms
iii. Find out how other researchers/specialists have detected and diagnosed it using clinical tests.
iv. Acquire the required data information
v. Classify the data into the required classes (Foot and Mouth, Healthy)
vi. Plan how to process the information and then right method to part them in train, test and approval information
vii. Plan on how you can map the data information into the deep learning

viii. Use different deep learning models for feature extraction to distinguish between Foot and Mouth and Healthy cattle,

ix. Evaluate and compare the results from different models and choose the best model

3.2.2. Mission requirements

The mission of the study is to detect foot and mouth diseases using deep learning architectures, to assess deep learning architecture model performance for the detection of foot and mouth and also come up with recommendations when taking images for the detection of foot and mouth. The requirement must be able solve the user needs as shown in the software engineering model.

3.2.3. Data Acquisition

In this study, data were acquired from the;

i.) Diseased Cattle downloaded from the Internet

ii.) Diseased Cattle acquired from European Union Foot and Mouth Division (EuFMD) portal

iii.) Diseased Cattle Acquired from Prightbright through reaching an agreement with the University

iv.) Healthy Cattle Images were taken at University farm
The healthy cattle dataset of JPEG images was taken with a Nikon D5300 24-MP digital camera. Foot and mouth images acquired are taken by different devices (cameras, smartphones) in variables conditions and have different resolutions.

### 3.3. Acquisition of Healthy cattle images

#### Day 1

The researcher took images of the mixed breed of cattle whilst they were grazing as shown in figure 52. There were about 300 cattle of mixed breed (Boran pure breed, crossbreeds of Boran, Brahman, Hard Mashona, Holstein, Thuli)) The images focused on the face of the cattle so that the pictures can be analysed to check whether the cattle have drooling or not.

![Figure 52: Images taken whilst cattle were grazing](image)

#### Day 2

20 Cattle of mixed breed (Boran pure breed, crossbreeds of Boran, Brahman, Hard Mashona, Holstein, and Thuli) were selected by the University of Zimbabwe (UZ) farm staff and brought into the concealed holding yards adjacent to the handling facilities at 7 am. Cattle are marshalled into a single file race passing through a weighing station leading to the cattle crush. The Cattle crush allows easy handing of cattle and it is normally used for artificial insemination, castration, dehorning and weighing.

The cattle head is locked by the head bail in a cattle crush and the farm staff uses a nose holder(cattle tongs) to hold the nose so that the mouth can be opened and the researcher
(Kuhamba) takes the images of the region of interest (tongue, gum pad, teats) as shown in figure 53. The duration of the handling procedure and taking images for each cattle was approximately about 5 to 7 minutes.

Day 3

10 Cattle of mixed breed (Boran pure breed, crossbreeds of Boran, Brahman, Hard Mashona, Holstein, and Thuli) were selected by the University of Zimbabwe (UZ) farm staff and brought into the concealed holding yards adjacent to the handling facilities at 7 am. Cattle are marshalled into a single file race passing through a weighing station leading to the cattle crush. Cattle head is locked by the head bail, a rope tied one leg, and the feet are washed so that images can be taken as shown in Figure 54. The duration of the handling procedure and taking images for each cattle was approximately about 5 to 7 minutes.
3.4 Journals and Books

Some of the research information required by the researcher was not locally available, hence the need to use the internet and books as well as journals from platforms like research gate, Google scholar IEEE and Elsevier. Some of the information includes how the deep learning architectures works, foot, and mouth current detection methods. Also, there was a need to consult with veterinary specialists both local and international about foot and mouth disease. The information was an addition to the literature review. Journals and books were also helpful in the derivation of the classification if the disease and also comes with the best model for foot and mouth detection.

3.5 Data Preparation

Images need to be classified into different classes since we are using supervised learning. Datasets are grouped into three which are the Training directory, Validation directory and also the Testing directory. The training and validation directory are further divided into two classes; Foot and Mouth Diseased images and Healthy pictures. The Foot and Mouth directory and Healthy directory are then further divided into five classes of the symptoms as shown in Figure 55 and 56. The Test directory is not categorised into foot and mouth classes. The following are important steps in data preparation selection of area of interest, image augmentation and resizing.
3.5.1. **Image Pre-processing**

i. Selecting the area of interest

ii. Image Augmentation

iii. Introducing diseases in healthy images.

iv. Resizing images

3.5.2. **Selecting the area of interest**

Original images taken by the camera need to be pre-processed by selecting the area of interest that is relevant to the study under investigation. This was done to remove the background interference that could occur when the image is taken, as the cattle will restricting to be held.
3.5.3. Image Augmentation

Images augmentation was performed so as to increase the dataset as a smaller dataset was acquired for this study as shown in Figure 58. Data augmentation is used so as to avoid overfitting due to the few training data samples (Chollet, 2018).
3.5.4. Resizing Images

The image is resized so that many images can be used without losing important features of the image. The images are first resized to 224×224 for VGG net, ResNet and DenseNets architectures. Then again, for the Inception V4 design the images are resized to 299×299 pixels. Normalization of information is done by dividing all the finished by pixels by 255 to make compatible with the network values.

3.6. Foot and Mouth Detection

In this part, a softmax classifier, which consists of a vector of features, outputs the probability of an image belonging to a certain type of class e.g. foot and mouth disease or healthy class. Figure 60 shows the testing procedure to check whether an image displays the foot and mouth symptoms or not.

![Figure 59: Softmax Classification](image-url)
3.7 Bayes optimal Classification

Bayes optimal classifier is utilized to locate the plausible classification (foot and mouth, healthy) of the new instance (test data or images) given the training data. The most plausible classification of the (test data) is acquired by the predictions of all hypothesis, weighted by the posterior probabilities. (Grosan and Abraham, 2011)

Let $V = \{\text{foot and mouth, healthy}\}$

Equation 3.1 shows the conditional probability for new instance $V_j$ given training data $D$ and also hypothesis $H$. $v_j$ is new instance (test data) to be classified, $P(v_j|h_i)$ is the posterior probability for $v_j$ given hypothesis $h_i$, $P(h_i|D)$ is the posterior probability for hypothesis is $h_i$, $P(h_i|D)$ is the posterior probability for hypothesis is $h_i$ given data $D$, $P(v_j|D)$ is the probability that correctly classifies test data $v_j$ and $D$ is the training data.

$$P(v_j|D) \equiv \arg \max_{v_j \in V} \sum_{h_i \in H} P(v_j|h_i)P(h_i|D)$$
Bayes optimal classifier of the test data is the value for which \( P(v_j|D) \) is maximum. In other words bayes optimal selects the maximum values of the test data and classifies whether the cattle is diseased with Foot and mouth or not.

2.14 Training Strategy

In the study we train different models with balanced and imbalanced data, evaluate using a binary classifier and multi classifier. We compute performance metrics for the different scenarios. Since the pictures were taken in the uncontrolled environment the distinctive lighting conditions and background in the preparation pictures may bias the neural network. To test this, the experiment was performed utilizing the grayscale.

![Figure 61: Sample Gray Scale images left side healthy cattle and right gum lesion](image)

3.7.1. Binary Classifier Training

- Detecting foot and mouth using imbalanced datasets using a binary classifier
- Introducing diseases in healthy images so that the datasets can be balanced and test the performance on with different deep learning architectures
- Training from scratch and also fine-tuning the pre-trained models
3.7.2. Multi Classifier Training

- Evaluate the impact of using imbalanced datasets on multi-classifier training
- Introducing diseases in healthy images so that the datasets can be balanced and test the performance on with different deep learning architectures
- Training from scratch and also fine-tuning the pre-trained models

```
Training Images --> Binary Classifier --> Foot and mouth Detector
                     |                  |
                     | Data Augmentation|
                     |                  |
                     | Multi classification |
                     |                  |
                     | Foot and Mouth Identifier |
```

i. Training
ii. Image pre-processing
iii. Image augmentation
iv. Introducing diseases in healthy images to increase the datasets
v. Changing the number of epochs

3.7.3. Testing of images taken by a smartphone

Images taken by an iPhone 6s smartphone will be tested into the algorithm to verify whether the system is working very well. This will help in coming up with recommendations to the app developer and also what the farmer needs to do whether taking pictures investigating a specific foot and mouth symptom.

3.7.4. Deep Learning Models

i. Inception v3
ii. VGG 16
iii. Densenet 201 Model
iv. Resnet 50
v. InceptionResnet

Steps in Building models
i. Load the libraries
ii. Load Data
Steps in fine-tuning a network
Thus the steps for fine-tuning a network are as follow:

Steps in adjusting a network
Along these lines the means for adjusting a system are:
1. Add your custom network over an effectively-prepared base network.
2. Freeze the base network.
3. Train the part you included.
4. Unfreeze a few layers in the base network.
5. Jointly train both these layers and the part you included.

3.7.5. Evaluation of Model

- Accuracy
- Confusion Matrix
- Classification Report

Figure 62: Learning process
3.7.6. **Hardware Consideration**

To perform machine learning and deep learning on any dataset, the software/program requires a computer system powerful enough to handle the computing power necessary. The following hardware considerations were made for this study shown in Table 6. However as the computation of the results was too long, Google Colab was used as it provides free access to processing assets including GPUs. Google Colab enables the execution of self-assertive python code through the cloud and it has more computing power see Table 7 for the specifications.

**Table 8: Experimental Hardware**

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel® Core(TM) i7-8th Generation CPU @ 1.8 GHz processor</td>
</tr>
<tr>
<td>Memory</td>
<td>RAM — 8 GB</td>
</tr>
<tr>
<td>Hard disk</td>
<td>1TB HDD</td>
</tr>
<tr>
<td>Operating system</td>
<td>Windows 10</td>
</tr>
</tbody>
</table>

**Table 9 Summary of FMD distribution by serotype and topotype for Southern Africa,**

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>Intel® Core(TM) i7-8th Generation CPU @ 1.8 GHz processor</td>
</tr>
<tr>
<td>Memory</td>
<td>RAM — 32GB</td>
</tr>
<tr>
<td>GPU</td>
<td>Nvidia K80s, T4s, P4s and P100s.</td>
</tr>
<tr>
<td>Hard disk</td>
<td>68.40GB</td>
</tr>
<tr>
<td>Operating system</td>
<td>Windows 10</td>
</tr>
</tbody>
</table>
### 3.7 Program Verification Methods

**Table 10: Program Verification Methods**

<table>
<thead>
<tr>
<th>Requirement</th>
<th>Verification Items</th>
<th>Verification Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Import Libraries</td>
<td>Check whether libraries have added successfully</td>
<td>Try to use the library if it added successfully you can use it</td>
</tr>
<tr>
<td>Importing Datasets</td>
<td>Is it possible to import the images</td>
<td>Load the datasets and then check the number of images imported from the directories</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Check whether the images have been successfully imported</td>
</tr>
<tr>
<td>Datasets balance</td>
<td>Plot the datasets using matplotlib library</td>
<td>Plot for the datasets and verify whether the is data balanced</td>
</tr>
<tr>
<td>Import a pre-trained model</td>
<td>Import the dataset from the Imagenet database</td>
<td>Display the summary information to verify that the model has been loaded</td>
</tr>
<tr>
<td>Training the model</td>
<td>Check whether it is calculating the metrics</td>
<td>Verify whether the number of epochs is changing when it is calculating</td>
</tr>
<tr>
<td>Evaluating metrics</td>
<td>Check whether the accuracy metric and categorical cross-entropy loss (loss) have been inputted into your code for evaluation of the models.</td>
<td>Cross-check the history of the code and check accuracy metric, categorical cross loss</td>
</tr>
<tr>
<td></td>
<td>Check whether the system is learning features very well</td>
<td>Check whether the training loss, validation loss are close to zero or both accuracy closer to 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Display the graphs and check whether the loss is exponential decaying</td>
</tr>
<tr>
<td>Image Processing</td>
<td>Image Augmentation</td>
<td>Display the images and check whether the images have been augmented</td>
</tr>
</tbody>
</table>
### Evaluating Performance

| To check how many have been correctly predicted as a disease with foot and mouth and how many have not | Display the confusion matrix and analyse the result |
|                                                                                                        | Display the classification report and analyse the result |
|                                                                                                        | Display the Region under the Curve (ROC) |
|                                                                                                        | Display the area under the curve (AUC) |
| To check how many have been correctly predicted as a disease with foot and mouth and how many have not | Display the confusion matrix and analyse the result |
|                                                                                                        | Display the classification report and analyse the result |
| For balanced data use the Precision-Recall                                                         | For imbalanced data use the f1 score |
|                                                                                                        | Display the Region under the Curve (ROC) |
|                                                                                                        | Display the area under the curve (AUC) |
| To check whether the system is predicting foot and mouth or not                                     | Load a test image and run the test |

### 3.8. Risk Management

In carrying out the study there was a need to carry out risk assessments to combat some of the risks. It is always important to assess the risks that can hinder the project as it ensures continuing of the project. Table 11 shows the risks and the counter measures.
<table>
<thead>
<tr>
<th>Risk</th>
<th>Counter Measure</th>
</tr>
</thead>
</table>
| Failure to get images from the Veterinary department | Request from International Organisations dealing with Foot and Mouth (EuFMD, Pirbright Institute)  
Download from the Internet |
| Few images                   | Data Augmentation  
Image Pre-processing of healthy cattle and pre-process them introducing diseases  
Transfer learning            |
| Cattle feet in a muddy farm  | Isolate the cattle and wash their feet before taking pictures                    |
| Large training time required for CPU | Use Google Collabo Graphical Processing Unit (GPU)                                |
| Google Collabo resources are not guaranteed | Use the high-performance computer at the University of Zimbabwe               |
RESULTS AND DISCUSSIONS

4.1. Introduction

In this study, an assessment of appropriate deep learning architecture was conducted for the detection of Foot and Mouth. The assessment was done on distinctive pre-prepared convolutional neural systems, which are Inception, Resnet 152, VGG-16, Densenet 201, and Inception Resnet. All the outcomes as far as accuracies are accounted for in Table 12, Table 13 and Table 14 separately.

<table>
<thead>
<tr>
<th>Model</th>
<th>Layers</th>
<th>Params</th>
<th>Training Accuracy %</th>
<th>Validation Accuracy %</th>
<th>Training loss</th>
<th>Test Accuracy</th>
<th>Test loss</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception V3</td>
<td>41.2M</td>
<td>41.2M</td>
<td>0.9950</td>
<td>0.9525</td>
<td>0.0208</td>
<td>98.44</td>
<td>6.91</td>
</tr>
<tr>
<td>VGGNet</td>
<td>16</td>
<td>119.6M</td>
<td>0.8995</td>
<td>0.9300</td>
<td>0.314</td>
<td>79.69</td>
<td>50.11</td>
</tr>
<tr>
<td>Resnet</td>
<td>50</td>
<td>23.6M</td>
<td>0.9950</td>
<td>0.9850</td>
<td>0.0163</td>
<td>95.31</td>
<td>8.43</td>
</tr>
<tr>
<td>Resnet 152</td>
<td>152</td>
<td>58.5M</td>
<td>0.9825</td>
<td>0.9575</td>
<td>0.0367</td>
<td>100</td>
<td>8.70</td>
</tr>
<tr>
<td>Densenet 201</td>
<td>121</td>
<td>7.1M</td>
<td>0.9900</td>
<td>0.9900</td>
<td>0.0152</td>
<td>96.87</td>
<td>7.05</td>
</tr>
</tbody>
</table>

4.2. VGG16

Figure 63: VGG16 accuracy and loss performance
4.2.2 Inception v3

Figure 64: Inception v3 accuracy and loss performance

4.2.3 Densenet 201

Figure 65: Densenet 201 accuracy and loss performance

4.2.4 Resnet v2 152

Figure 66: Resnet 15v2 accuracy and loss performance
4.2.5 Resnet 50

Figure 67: Resnet50 accuracy and loss performance

4.2.6 InceptionResnetv2

Figure 68: InceptionResnetv2 accuracy and loss performance

4.3. Analysis of evaluation metrics

The results plotted in Fig 63 to Fig 68 shows the training accuracy and validation loss for each different models. Densenet 201 performs better than other models as it has training accuracy values closer to 1 and the lowest training loss closer to zero. VGG 16 performs the worst both the training accuracy and the higher loss as compared to the other deep learning architectures.

Of course, DenseNets, ResNet and Inception V3 performed bearably especially showed up contrastingly comparable to VGG net as depicted in the evaluation on Table 12. This shows that a deeper network performs better than shallow networks. The number of parameters on the more significant learning systems (DenseNets, ResNet and Inception V3) are lessened showed
up diversely comparable to VGG net. This result resembles (Huang et al., 2017) who indicated that increasingly significant models are continuously accurate and powerful to plan.

With DenseNets having a minimal number of parameters. DenseNets 201 has extensively diminished a number of parameters although its like ResNet. DenseNets is 8 times not exactly as ResNet 152 and 16 times, not exactly VGG net. Therefore it's simpler to train DenseNets as contrasted with the remainder of the deep learning architectures examined.

In DenseNet, classifier utilises features of all complex multifaceted levels and this tends to give progressively smooth decision boundaries, unlike other convolutional networks which utilises only the most complex (high level)features. This clarifies why DenseNet performed well ahead of other models regardless that the training dataset was small.

4.4. Evaluation Metrics

In this study evaluation performance metrics are used as validation measures for examination purposes, for example, Recall = \( \frac{TP}{TP+FN} \) and Precision = \( \frac{TP}{TP+FP} \), where TP, FP, and FN represent the true positives, false positives and false negatives.

\[
\text{Accuracy} = \frac{TP+TN}{TP+FN+FP+TN}
\]

19

\[
\text{Precision} = \frac{TP}{TP+FP}
\]

20

\[
\text{Recall} = \frac{TP}{TP+FN}
\]

21

We further utilize broadly utilized evaluation metrics measures, for example,

\[
\text{F1-score} = \frac{2 \times TP}{FP+FN+2 \times TP}, \quad \text{Receiver Operating Characteristics (ROC) and its Area Under Curve (AUC).}
\]

Table 13 shows a comparison of different studies on disease detection and also the evaluation of their performance metrics.
<table>
<thead>
<tr>
<th>Study</th>
<th>Model</th>
<th>Dataset</th>
<th>Performance Evaluation Metrics %</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Accuracy</td>
</tr>
<tr>
<td><strong>Tomato Disease detection</strong> (Brahimi, Boukhalfa and Moussaoui, 2017)</td>
<td>CNN</td>
<td>14828 images 9 diseases</td>
<td>99.18%</td>
</tr>
<tr>
<td><strong>Cassava detection and diagnosis</strong> (Ramcharan et al., 2019)</td>
<td>CNN</td>
<td>720 images 3 diseases</td>
<td>84.70%</td>
</tr>
<tr>
<td><strong>Soya Bean identification</strong> (Walleling, 2017)</td>
<td>CNN</td>
<td>12673 images 3 diseases</td>
<td>99.32%</td>
</tr>
<tr>
<td><strong>Banana leaf disease classification</strong> (Amara, Bouaziz and Algergawy, 2017a)</td>
<td>CNN</td>
<td>3700 images 3 diseases</td>
<td>98.61%</td>
</tr>
<tr>
<td><strong>Malaria Parasite Detection</strong> (Rahman et al., 2018)</td>
<td>VGG16</td>
<td>2756 images 2 classes</td>
<td>97.89%</td>
</tr>
</tbody>
</table>
The performance of the results of our study is comparable to other studies however there is a need for a larger dataset for the identification of individual diseases as it is the least with fewer dataset images.

4.5. Confusion Matrix

Confusion Matrix is utilised to portray the exhibition of a classification model. A confusion lattice is a basic method to format what number of anticipated classifications or classes were accurately anticipated and what number of were not.

**True Positive TP:** cases when classifier predicted TRUE (they have the disease-Foot and Mouth) and the correct class was TRUE (cattle has the disease- Foot and Mouth)

**True Negative TN:** cases when the model predicted FALSE (no disease-Healthy) and the correct class was FALSE (cattle do not have the disease-Foot and Mouth)

**False Positive FP:** (Type I error): classifier predicted TRUE but correct class was FALSE (cattle did not have the disease)
**False Negatives FN:** (Type II error): classifier predicted \textbf{FALSE} (cattle do not have the disease-Foot and Mouth) but they do have the disease

Table 14 Summary of confusion Matrix for different models

<table>
<thead>
<tr>
<th>Model</th>
<th>TP</th>
<th>TN</th>
<th>FP</th>
<th>FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception V3</td>
<td>59</td>
<td>116</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>VGG 16</td>
<td>51</td>
<td>117</td>
<td>3</td>
<td>9</td>
</tr>
<tr>
<td>Densenet 201</td>
<td>60</td>
<td>119</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Resnet 50</td>
<td>59</td>
<td>118</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Resnet 152</td>
<td>59</td>
<td>118</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>InceptionResnetv2</td>
<td>59</td>
<td>116</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

Figure 70: Comparison of the predictions for different deep learning architectures

Densenet 201 model has the highest True Positive and True Negative than the other model Inception v3, Resnet50,152, InceptionResnetv2 has the same False Positive.
Figure 71: VGG16 Confusion Matrix and Classification Report

Figure 72: Inception V3 Model

4.5.1. Densenet 201
Figure 73: DenseNet 201 Confusion Matrix and Classification Report

4.5.2. Resnet 152 v2

![Resnet 152 v2 Confusion Matrix and Classification Report](image)

Figure 74: Resnet 152 v2 Confusion Matrix and Classification Report

4.5.3. Resnet50

![Resnet 50 Confusion Matrix and Classification Report](image)

Figure 75: Resnet 50 Confusion Matrix and Classification Report
4.5.4. Inception Resnetv2

![Inception Resnetv2 Confusion matrix](image)

<table>
<thead>
<tr>
<th>True labels</th>
<th>Predicted labels</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foot_and_Mouth</td>
<td>Foot_and_Mouth</td>
</tr>
<tr>
<td>Healthy</td>
<td>Healthy</td>
</tr>
</tbody>
</table>

InceptionResnetV2 Classification Report

<table>
<thead>
<tr>
<th></th>
<th>precision</th>
<th>recall</th>
<th>f1-score</th>
<th>support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Foot_and_Mouth</td>
<td>0.94</td>
<td>0.98</td>
<td>0.96</td>
<td>68</td>
</tr>
<tr>
<td>Healthy</td>
<td>0.99</td>
<td>0.97</td>
<td>0.98</td>
<td>120</td>
</tr>
</tbody>
</table>

accuracy: 0.97, 188
macro avg: 0.96, 0.97, 0.97, 188
weighted avg: 0.97, 0.97, 0.97, 188

![VGG16 Multiclassification Performance Report](image)

Figure 76: Multiclassification Training and Accuracy Report

There is a poor representation of classes probabilities due to the small samples of diseased foot and mouth which the researcher had difficulty in acquiring as seen from figure 77 and 78.
4.6. **Comparison of Evaluation Metrics**

The idea behind sensitivity and specificity is that these measures how well a screening test determines how whether someone or something has a condition. The key thing about sensitivity and specificity they are well known as conditional probabilities. This means the denominator used to calculate the probability is not the entire group but instead the subset of the group in this case whether or not cattle has the foot and mouth disease.

**Sensitivity** is the probability that the screening test is positive given that cattle have foot and mouth disease.
Specificity is the probability that the screening test is negative given that cattle do not have the foot and mouth disease.

\[
\text{Sensitivity} = \text{true positive rate: } TPR = \frac{\text{positive correctly classified}}{\text{total positives}} = \frac{TP}{TP+FN}
\]

\[
\text{Specificity} = \text{true negative rate: } FNR = \frac{\text{negative correctly classified}}{\text{total negatives}} = \frac{TN}{FP+TN}
\]

Table 12 shows the comparison performance evaluating metrics for different deep learning architecture.

<table>
<thead>
<tr>
<th>Model</th>
<th>Status</th>
<th>Accuracy %</th>
<th>Precision</th>
<th>Recall</th>
<th>F1 Score</th>
<th>Sensitivity</th>
<th>Specificity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception</td>
<td>Foot and Mouth</td>
<td>98.44</td>
<td>0.94</td>
<td>0.98</td>
<td>0.98</td>
<td>0.9833</td>
<td>0.9666</td>
</tr>
<tr>
<td></td>
<td>Healthy</td>
<td></td>
<td>0.99</td>
<td>0.97</td>
<td>0.98</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VGG 16</td>
<td>Foot and Mouth</td>
<td>79.69</td>
<td>0.94</td>
<td>0.85</td>
<td>0.89</td>
<td>0.85</td>
<td>0.975</td>
</tr>
<tr>
<td></td>
<td>Healthy</td>
<td></td>
<td>0.93</td>
<td>0.97</td>
<td>0.95</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Densenet - 201</td>
<td>Foot and Mouth</td>
<td>93.75</td>
<td>0.98</td>
<td>1.00</td>
<td>0.99</td>
<td>1.0</td>
<td>0.9916</td>
</tr>
<tr>
<td></td>
<td>Healthy</td>
<td></td>
<td>1.00</td>
<td>0.98</td>
<td>1.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resnet-50</td>
<td>Foot and Mouth</td>
<td>100.00</td>
<td>0.97</td>
<td>0.98</td>
<td>0.98</td>
<td>0.9833</td>
<td>0.9833</td>
</tr>
<tr>
<td></td>
<td>Healthy</td>
<td></td>
<td>0.99</td>
<td>0.98</td>
<td>0.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resnet-152</td>
<td>Foot and Mouth</td>
<td>100.00</td>
<td>0.97</td>
<td>0.98</td>
<td>0.98</td>
<td>0.9666</td>
<td>0.975</td>
</tr>
<tr>
<td></td>
<td>Healthy</td>
<td></td>
<td>0.99</td>
<td>0.98</td>
<td>0.99</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Inception-ResnetV2</td>
<td>Foot and Mouth</td>
<td>93.75</td>
<td>0.94</td>
<td>0.98</td>
<td>0.96</td>
<td>0.9833</td>
<td>0.9666</td>
</tr>
<tr>
<td></td>
<td>Healthy</td>
<td></td>
<td>0.99</td>
<td>0.97</td>
<td>0.98</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 79:** Specitivity and Sensitivity
0% specificity means that healthy cattle will be labelled as unhealthy (in our case meaning diseased with Foot and Mouth)

100% Specificity means that all healthy cattle are labelled as healthy

0% Sensitivity means that all cattle with Foot and Mouth (unhealthy) are labelled as healthy

100% Sensitivity means that all cattle with Foot and Mouth (unhealthy) are labelled as diseased with Foot and Mouth (unhealthy).

Densenet 201 outperformed other deep learning architectures in sensitivity which is one of the most important parameters in the medical field as shown in figure 80. Densenet 201 model has the highest performance metric of F1 score which is also important the medical field.

![Evaluating Performance measures](image)

**Figure 80:** Evaluating Performance for different deep learning architectures

4.7. Evaluating the performance on test data

4.7.1. ROC and AUC curve graphs

The Receiver Operating Curve (ROC) is most commonly used in evaluating the performance of a binary classifier at various probability thresholds. ROC compares the presence of true positives to false negatives at every probability threshold. Areas under the ROC curve were used to determine the binary classification outcomes of healthy vs foot and mouth diseased
cattle. The area under the curve evaluates the areas under the ROC curve and the closer is to one the better the model. The higher the curve the better the model and also the greater the area under the curve.

Figure 81: VGG16 ROC

Figure 82: Inception v3 ROC

Figure 83: Densenet 201 ROC
### 4.7.2. Resnet 152v2 Classification Report

![Resnet 152v2 ROC](image)

**Figure 85:** Resnet 152v2 ROC

![InceptionResnetv2 ROC](image)

**Figure 86:** InceptionResnetv2
Table 16 ROC and AUC for different deep learning architectures

<table>
<thead>
<tr>
<th>Model</th>
<th>AUC</th>
<th>ROC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inception V3</td>
<td>0.98</td>
<td>0.9750</td>
</tr>
<tr>
<td>VGG 16</td>
<td>0.91</td>
<td>0.9125</td>
</tr>
<tr>
<td>Densenet 201</td>
<td>0.99</td>
<td>0.9958</td>
</tr>
<tr>
<td>Resnet-50</td>
<td>0.98</td>
<td>0.9833</td>
</tr>
<tr>
<td>Resnet152</td>
<td>0.98</td>
<td>0.9833</td>
</tr>
<tr>
<td>InceptionResnetv2</td>
<td>0.98</td>
<td>0.9750</td>
</tr>
</tbody>
</table>

Densenet 201 is a better model as it is closer to 1 than outperforms other deep learning architectures. Inception v3, Resnet50, Resnet152 are comparable but far much better than VGG16.

4.8. Predictions on test images

Prediction probability from the colour images is better than the grayscale and soft-focus images. Soft focus and greyscale images reduce the prediction probability as shown by Figure 80, 81, 83 and 84. Soft focus also causes the wrong prediction as shown in Figure 82 and 84. This shows the colour feature and focus is important to extract important features for classification.
Figure 88: Prediction of Drooling and non-drooling image

Figure 89: Prediction of gum lesion and non-gum lesion images
Figure 90: Prediction of drooling and non-drooling image

Figure 91: Prediction of teat lesion and non lesion
Colour images have more classification accuracy than grayscale and soft-focus images as shown by figure 88-92 thus colour has a significant effect on classification accuracy. In Figure 92 gray scale colour caused a wrong prediction for the tongue lesion. The soft-focus has also a major effect on the classification accuracy as it decreases and also the wrong prediction in figure 90 and 92.

4.8.1. Batch testing

Batch testing was conducted to 37 images of foot and mouth images received from Pirbright institute and the snippet of the result is shown in figure 93.
CHAPTER FIVE: CONCLUSION

This study set out to evaluate how effective is the detection of FMD using different deep learning architectures. Findings of this study’ show that FMD can be detected using deep learning however larger datasets of both FMD and healthy images are required to improve the performance evaluation metrics and also the identification of the disease. There has been a major challenge in acquiring images of FMD diseased cattle resulting in a smaller dataset used for this study. Thus veterinary departments and international organisations across the World must be encouraged to take images and archive of cattle diseased with FMD. The main contribution of this study is that it has set the groundwork for the development of a mobile application (app) that will be used for the detection of FMD. The study also sets out the requirements that will be taken into considerations in the development of the mobile app for the detection of FMD. The mobile app will help farmers in timeously detecting FMD thus reducing the loss of livestock, loss of milk production and also allowing countries to avoid trade restrictions due to FMD disease. Colour information and image focus are important in foot and mouth detection. There is also a need to collect images of different breeds of cattle across the world both diseased and healthy cattle so that the deep learning architecture can be trained to learn features which enhances the mobile app to be utilised across the world. However further work needs to be done in the investigation for the use of deep learning in the diagnosis of FMD. The work has been submitted to the Engineering Applications for Engineering Journal for consideration.
References


Atienza, R. (no date) *Advanced Deep Learning with Keras*.


Gloster, J. *et al.* (2011) ‘Normal variation in thermal radiated temperature in cattle :
implications for foot-and-mouth disease detection’, (July 2010).


Gutstein, S. M. (no date) ‘No Title’.


Lu, L. et al. (no date) Advances in Computer Vision and Pattern Recognition Deep Learning and Convolutional Neural Networks for Medical Image Computing.


DenseNet Convolutional Neural Network. doi: 10.1109/ICRIS.2019.00028.

APPENDICES

Appendix A:

Project plan