



Faculty of Computer Science and Information Technology

**DETECTION AND CLASSIFICATION OF EXTERNAL BOVINE DISEASES
USING DEEP LEARNING**

Muhamad Faaris Bin Jamhari

**Bachelor in Computer Science with Honours
(Computational Science)
2025**

**DETECTION AND CLASSIFICATION OF EXTERNAL BOVINE DISEASES
USING DEEP LEARNING**

Muhamad Faaris Bin Jamhari

**This project is submitted in partial fulfilment of the requirements for the degree
of Bachelor of Computer Science with Honours (Computational Science)**

**Faculty of Computer Science and Information Technology
UNIVERSITI MALAYSIA SARAWAK
2025**

UNIVERSITI MALAYSIA SARAWAK

THESIS STATUS ENDORSEMENT FORM

TITLE: DETECTION AND CLASSIFICATION OF EXTERNAL BOVINE DISEASES USING DEEP LEARNING

ACADEMIC SESSION: 2-2024/20

MUHAMAD FAARIS BIN JAMHARI

hereby agree that this Thesis* shall be kept at the Centre for Academic Information Services, Universiti Malaysia Sarawak, subject to the following terms and conditions:

1. The Thesis is solely owned by Universiti Malaysia Sarawak
2. The Centre for Academic Information Services is given full rights to produce copies for educational purposes only
3. The Centre for Academic Information Services is given full rights to do digitization in order to develop local content database
4. The Centre for Academic Information Services is given full rights to produce copies of this Thesis as part of its exchange item program between Higher Learning Institutions [or for the purpose of interlibrary loan between HLI]
5. ** Please tick (√)

☐

CONFIDENTIAL

(Contains classified information bounded by the OFFICIAL SECRETS ACT 1972)

☐

RESTRICTED

(Contains restricted information as dictated by the body or organization where the research was conducted)

☒

UNRESTRICTED



(AUTHOR'S SIGNATURE)

Validated by



(SUPERVISOR'S SIGNATURE)

Permanent Address

No 653, Jalan Allamanda 5,
Taman Bandar Jaya, 97000
Bintulu, Sarawak

Date: 28/07/2025

Date: 30 Jul 2025

Note * Thesis refers to PhD, Master, and Bachelor Degree

** For Confidential or Restricted materials, please attach relevant documents from relevant organizations / authorities

DECLARATION

I hereby declare that this project altogether with all its contents is none other than that of my work, with consideration of exception of reference or acknowledgement is not made explicitly in test, nor has any part had been written for me another person.



Signature

Name: MUHAMAD FAARIS BIN JAMHARI

Matric Number: 78196

Date: 17/01/2025

Faculty of Computer Science and
Information Technology,
Universiti Malaysia Sarawak

ACKNOWLEDGEMENT

First and foremost, I extend my deepest gratitude to Allah, the Almighty, for His blessings, wisdom, and protection throughout this journey. His guidance has been my source of strength, composure, and determination, enabling me to complete this project successfully.

I would also like to express my sincere appreciation to my supervisor, Associate Professor Dr. Chiew Kang Leng, for his invaluable guidance, support, and encouragement throughout the course of this Final Year Project. His constructive feedback, insightful advice, and unwavering patience have been instrumental in shaping the progress and outcome of this research.

My heartfelt thanks go to my family and friends, who stood by me with unwavering support. Their words of encouragement, thoughtful advice, and emotional backing have been a source of comfort and motivation, especially during challenging moments.

Lastly, I am deeply thankful to Universiti Malaysia Sarawak (UNIMAS) for granting access to essential research facilities and resources. In particular, I extend my gratitude to the Faculty of Computer Science and Information Technology for their continuous support, which has been pivotal in facilitating this academic journey.

To all who contributed directly or indirectly to this achievement, thank you. Your kindness and support have been invaluable.

ABSTRACT

Livestock diseases pose significant challenges to agricultural productivity, often requiring timely and accurate diagnosis to mitigate economic losses and improve animal welfare. Traditional diagnostic methods, reliant on veterinary expertise, can be time-intensive and inaccessible in remote areas. This project focuses on developing a YOLOv11-based deep learning model to detect and classify visible external bovine diseases. A curated and augmented dataset ensures diversity and robustness, while training on Google Colab leverages cloud-based computational resources to optimize precision, recall, and mean Average Precision (mAP). The resulting model aims to provide a reliable and efficient foundation for disease detection, offering potential for future system integration to support farmers with rapid and accurate diagnostics.

Keywords: Bovine diseases, deep learning, YOLOv11, object detection, image classification

ABSTRAK

Cabaran utama kepada produktiviti pertanian ialah penyakit haiwan ternakan, dimana proses diagnosis yang jitu diperlukan untuk mengurangkan kerugian ekonomi dan menaik taraf kebajikan haiwan. Walaubagaimanapun, kaedah diagnostik tradisional oleh pakar yang berpengalaman boleh memakan masa dan sukar untuk diakses di kawasan terpencil. Oleh yang demikian, projek ini menumpu kan fokus kepada Pembangunan model “Deep Learning” berasaskan YOLOv11 untuk mengesan dan mengklasifikasikan penyakit lembu, yang menunjukkan gejala fizikal. Data yang dikumpul disusun menjalankan “augmentation untuk memastikan kepelbagaian dan ketahanan, Selain itu, latihan menggunakan Google Colab dapat memanfaatkan sumber pengkomputeran yang berasaskan “cloud-based” untuk mengoptimumkan “precision”, “recall”, dan “mean Average Precision” (mAP). Model yang dihasilkan bertujuan menyediakan asas yang boleh dipercayai dan cekap untuk mengesan penyakit dan menawarkan potensi dalam mengintegrasikan sistem pada masa hadapan agar petani mendapatkan diagnostik yang pantas dan tepat.

Kata Kunci: Pengesanan penyakit, “deep learning”, YOLOv11, pengesanan objek, klasifikasi imej

TABLE OF CONTENTS

DECLARATION	i
ACKNOWLEDGEMENT	iii
ABSTRACT	iv
ABSTRAK	v
TABLE OF CONTENTS	vi
LIST OF TABLES	ix
LIST OF FIGURES	x
LIST OF ABBREVIATIONS	xi
CHAPTER 1: INTRODUCTION	1
1.1 Introduction	1
1.2 Problem Statement	2
1.3 Objectives	2
1.4 Brief Methodology	2
1.5 Project Scope	3
1.6 Significance of Project	3
1.7 Project Schedule	4
1.8 Expected Outcome	5
1.9 Project Outline	5
1.10 Summary	6
CHAPTER 2: LITERATURE REVIEW	8
2.1 Introduction	8
2.2 Overview of Bovine Disease Detection	8
2.2.1 Machine Learning Disease Detection	9
2.2.2 Deep Learning Disease Detection	9
2.3 Related Works On Disease Detection	10

2.3.1 Convolutional Neural Networks (CNNs).....	10
2.3.2 Hybrid Model (CNN-SVM).....	11
2.3.3 You Only Look Once (YOLO)	12
2.4 Comparison on Reviewed Disease Detection Works.....	12
2.5 Model Comparison.....	17
2.6 YOLO Family of Models	19
2.6.1 Versions.....	19
2.6.2 Variants	20
2.6.3 Size	21
2.7 Summary	22
CHAPTER 3: METHODOLOGY	24
3.1 Introduction	24
3.2 Dataset Preparation	24
3.2.1 Collection	25
3.2.2 Annotation.....	25
3.2.3 Organization	26
3.2.4 Export Dataset.....	26
3.3 Training	26
3.4 Inferencing	27
3.5 Deployment	28
3.6 Summary	28
CHAPTER 4: IMPLEMENTATION AND DISCUSSION.....	30
4.1 Overview	30
4.2 Dataset.....	30
4.2.1 Collection and Preparation	30
4.2.2 Preprocessing	31
4.2.3 Curated Dataset	32

4.3 Dataset Imbalance Mitigation Strategy	33
4.4 Training	36
4.5 Optimisation and Finetuning.....	38
4.6 Testing and Inference	39
4.7 Performance Evaluation	39
CHAPTER 5: CONCLUSION AND FUTURE WORK	42
5.1 Overview	42
5.2 Achievements	42
5.3 Limitations	43
5.4 Future Works.....	43
5.5 Conclusion.....	44
REFERENCES.....	45
APPENDIX A	48
Prediction Example on Validation Set	48
APPENDIX B	50
Github RepositoryLink.....	50

LIST OF TABLES

Table 2.1: Summary of reviewed related works on disease detection	13
Table 2.2: Comparison of models used in the reviewed related works.....	17
Table 2.3: Breakdown of different YOLO sizes and its performance metrics.....	21
Table 2.4: Pros and cons in different sizes of YOLOv11	22
Table 4.1: Annotations in collected images	32
Table 4.2: Augmentations applied to all dataset in the multi model approach	34
Table 4.3: Summary of the dataset to be used in multi model approach	35
Table 4.4: Final Hyperparameters Used In Training.....	38
Table 4.5: Performance summary on the trained models.....	40
Table 5.1: Achievements summary	42
Table 5.2: Limitations of this project	43

LIST OF FIGURES

Figure 1.1: FYP 1 project schedule	4
Figure 2.1: Latency performance on different models and sizes	19
Figure 2.2: Breakdown of different YOLO variants	20
Figure 3.1: General flow of the proposed approach.....	24
Figure 3.2: Proposed dataset preparation process flow	25
Figure 3.3: Proposed training process flow.....	27
Figure 4.1: Implemented Dataset Preparation Flow.....	30
Figure 4.2: Class Label Distribution Pie Chart for Baseline Dataset.....	33
Figure 4.3: Process flow of the multi model approach	34
Figure 4.4: Model Training and Development Process Flow	36

LIST OF ABBREVIATIONS

No	Abbreviations	Full Sentence
1.	AI	Artificial Intelligence
2.	API	Application Programming Interface
3.	BRD	Bovine Respiratory Disease
4.	CNN	Convolutional Neural Network
5.	COCO	Common Objects in Context
6.	CPU	Central Processing Unit
7.	DD	Digital Dermatitis
8.	DL	Deep Learning
9.	FL	Front Left
10.	FMD	Foot and Mouth Disease
11.	FPS	Frames Per Second
12.	FR	Front Right
13.	GAN	Generative Adversarial Network
14.	GPU	Graphics Processing Unit
15.	IBK	Infectious Bovine Keratoconjunctivitis
16.	LSD	Lumpy Skin Disease
17.	ML	Machine Learning
18.	ONNX	Open Neural Network Exchange
19.	R-CNN	Region based Convolutional Neural Network
20.	RL	Rear Left
21.	ROI	Region Of Interest
22.	RR	Rear Right
23.	SGD	Stochastic Gradient Descent
24.	SVM	Support Vector Machine
25.	VGG	Visual Geometry Group
26.	YOLO	You Only Look Once

CHAPTER 1: INTRODUCTION

1.1 Introduction

The health and well-being of livestock are critical to the agricultural economy, with Bovinae being a major contributor to the dairy, beef, and leather industries (Kappes et al., 2023). Among the key challenges facing the livestock industry is the timely and accurate diagnosis of diseases, particularly those with visible external symptoms. Undiagnosed or misdiagnosed diseases in Bovinae can lead to significant economic losses due to reduced productivity, high treatment costs, and potential contagion (Jaime Betancur Hurtado & Giraldo-Ríos, 2019). In recent years, advances in artificial intelligence (AI) and deep learning have opened up new possibilities for automating disease detection and minimizing reliance on specialized human intervention (Neethirajan, 2023).

Despite progress in veterinary medicine, real-time detection and diagnosis of diseases in bovine species, such as cattle, remains a significant challenge, especially in resource-limited areas where access to veterinary services is cumbersome (Fuentes et al., 2022). Many external diseases in cattle, such as Lumpy Skin Disease (LSD), Foot and Mouth Disease (FMD), and Ringworm, exhibit visible symptoms, but timely detection often depends on skilled veterinarians or farmworkers. Delayed detection and misdiagnosis contribute to the spread of infections and increase the difficulty in effective treatment (Kappes et al., 2023). Utilizing AI including computer vision tools could provide much-needed solution to bridge this gap (Neethirajan, 2023).

Current disease detection approaches are largely manual and require veterinary expertise to detect symptoms. While some automated tools have been developed, they often focus on monitoring behaviour or internal physiological signs, rather than leveraging visible symptoms for diagnosis (Fuentes et al., 2022). Moreover, these tools may lack the accuracy or robustness to handle various diseases simultaneously or generalize across different environments (Fuentes et al., 2022). For example, traditional image processing techniques used in tools such as OpenCV may not have the level of sophistication needed for complex disease classification, and existing machine learning models may lack generalization or precision when applied to different bovine diseases.

This research proposes the development of a deep learning-based object detection model using architectures such as You Only Look Once (YOLO) to automate the detection and

classification of bovine diseases based on visible symptoms. By leveraging a curated dataset of annotated images for common bovine diseases like LSD, FMD, Ringworm, and others, this model aims to achieve accurate, near real-time classification. The proposed system would provide a practical and accessible tool for farmers and livestock owners to detect diseases earlier and with greater accuracy, reducing reliance on immediate veterinary services and improving disease management outcomes (Neethirajan, 2023).

1.2 Problem Statement

Access to veterinary services is often limited and slow, leading to delays in the diagnosis, treatment, and prevention of diseases, including zoonotic diseases, which can have catastrophic consequences. Early detection of diseases is critical to mitigating these risks, but it relies on expert knowledge and training that may not always be readily available in rural or resource-constrained settings. Livestock farmers face significant challenges in identifying and managing diseases promptly, resulting in economic losses and potential public health concerns.

Existing computational techniques for automated disease detection, such as Convolutional Neural Networks (CNNs) and Support Vector Machines (SVMs), have been explored to address these challenges. However, these methods have limitations: CNNs typically require extensive, high-quality datasets to achieve accurate predictions and are not optimized for real-time applications, while SVMs struggle to handle complex image data, particularly when localization or multi-object detection is required (Mia et al., 2018). These shortcomings highlight the need for an efficient, accurate, and near real-time disease detection system tailored to the livestock industry.

1.3 Objectives

1. To investigate the process of detecting and classifying bovine diseases based on visible symptoms.
2. To develop a custom-trained deep learning object detection model for bovine disease detection and classification.
3. To evaluate the performance of the proposed model.

1.4 Brief Methodology

This project aims to develop a deep learning based system for detecting and classifying visible bovine diseases using the YOLO model. The methodology is divided into several phases:

1. **Data Collection:** A dataset of bovine diseases (e.g., LSD, FMD, Ringworm, IBK) will be curated, annotated, and augmented to improve model robustness. Image will be labelled based on the type of disease.
2. **Model Development:** YOLO will be trained and fine-tuned on this dataset to detect and classify diseases based on visible symptoms. Hyperparameter tuning and data augmentation will be applied to enhance model performance.
3. **Performance Evaluation:** The model will be tested on an unseen dataset, and performance will be assessed based on accuracy, precision, and recall.
4. **Deployment:** The trained model will be deployed through Roboflow Universe (Roboflow, n.d.), enabling users to access the disease detection system via a web-hosted interface. This platform facilitates near real-time inference and easy integration, allowing users to upload bovine images and receive instant health assessments directly through the hosted model.

1.5 Project Scope

This project aims to develop a deep learning model for the accurate detection and classification of bovine diseases characterized by visible external symptoms, including Lumpy Skin Disease (LSD), Foot and Mouth Disease (FMD), Ringworm, and Pink Eye/Infectious Bovine Keratoconjunctivitis (IBK). A custom dataset will be created and annotated to encompass images reflecting these symptoms, with data augmentation techniques applied to enhance dataset diversity and robustness. Using this dataset, a deep-learning object detection model, such as YOLO, will be trained and fine-tuned to optimize detection accuracy and performance. The model's effectiveness will be evaluated on unseen test data. Finally, the trained model(s) performance will be examined and analyzed.

1.6 Significance of Project

This project holds significant value as a research-driven initiative aimed at advancing the application of deep learning in livestock disease detection. Bovine diseases can result in substantial economic losses, particularly for farmers in resource-limited regions where veterinary services are not readily accessible. By focusing on the development of an accurate, real-time object detection model using YOLO, this study contributes to the growing body of research in agricultural AI and computer vision.

The system's potential to deliver rapid and preliminary diagnostics through mobile devices offers a cost-effective solution that supports early intervention and disease

management. This is especially critical for improving herd health, minimizing productivity loss, and curbing the spread of infections. Beyond immediate applications, the project also aims to support long-term benefits by contributing to a scalable dataset and a reproducible model architecture that can serve as a foundation for future research and system improvements. Ultimately, the integration of this model into a practical tool can bridge the gap between research innovation and real-world agricultural challenges.

1.7 Project Schedule

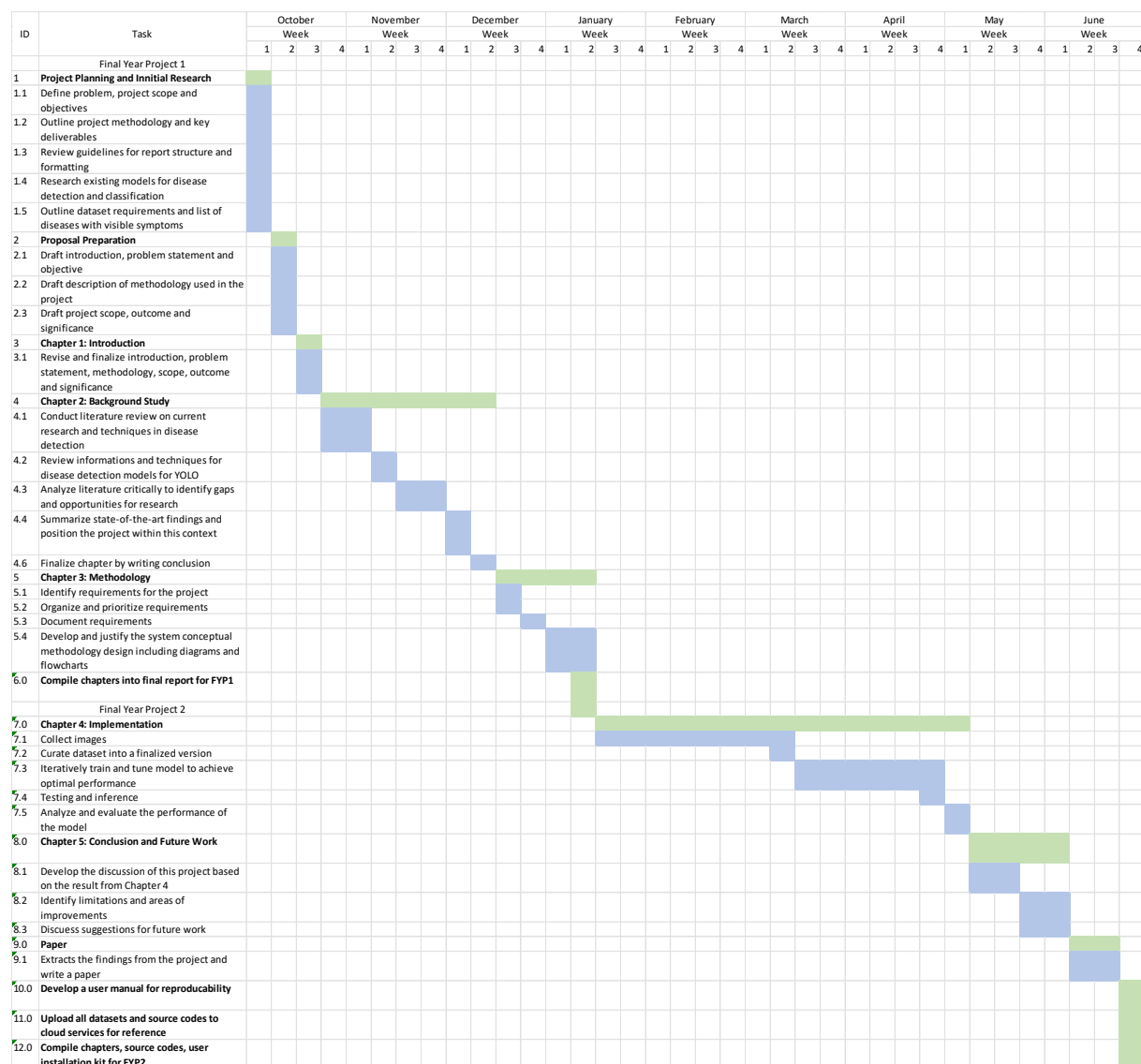


Figure 1.1: *Final Year Project timeline*

Figure 1.1 illustrates the timeline of this project.

1.8 Expected Outcome

The expected outcome of this project is a deep learning-powered application capable of detecting and classifying various bovine diseases based on visible external symptoms with high accuracy and speed. Deployed as a mobile application, it will allow users, such as farmers and livestock managers, to upload images of Bovine and instantly receive a disease assessment. This solution will streamline disease detection, reduce dependency on veterinary presence, and provide insights for proactive herd management, improving overall efficiency and accessibility of bovine healthcare in resource-limited settings.

The expected outcome of this project is the development of a deep learning-based object detection model capable of accurately detecting and classifying various bovine diseases based on visible external symptoms. The model, built using the YOLO architecture, will be custom-trained on a curated dataset comprising images of diseases such as LSD, FMD, Ringworm, and IBK. This dataset, which will be collected, annotated, and augmented specifically for this study, will serve as a significant contribution to academic research in the field of livestock disease detection using computer vision. It will provide a valuable foundation for future studies aiming to improve or expand on bovine health monitoring systems.

The trained model is expected to demonstrate high recall, and precision capabilities. Once developed, the model will be deployed via a mobile-accessible platform, enabling users, particularly farmers and livestock managers, to upload images of bovine animals and receive instant disease assessments. This solution will streamline the detection process, reduce dependency on immediate veterinary services, and support proactive disease management, especially in remote or resource-limited agricultural settings. By achieving these outcomes, the project aims to fulfill its objectives while also supporting ongoing research and innovation in AI-driven livestock health solutions.

1.9 Project Outline

The organization of this report is divided into five main chapters:

Chapter 1: Introduction

This chapter provides an overview of the project, including the background, problem statement, objectives, and significance. The need for automated disease detection in livestock, specifically for bovine diseases with visible symptoms, is discussed, as well as the potential benefits of a deep-learning approach.

Chapter 2: Literature Review

The literature review explores previous works in disease detection using computer vision, with a focus on existing models for object detection and classification in agricultural contexts. Comparative analysis of various techniques, such as traditional image processing and modern deep learning models, highlights limitations in current solutions and provides a rationale for using YOLO-based architectures in this project.

Chapter 3: Methodology

This chapter describes the design and development process of the proposed tool. Details include dataset collection and annotation, model selection and training, and the application of data augmentation. The training, validation, and testing procedures for the deep learning model are outlined, along with the metrics and criteria used for evaluation. The deployment of the model is also discussed.

Chapter 4: Implementation and Discussion

In this chapter, the model's performance and evaluation metrics, such as accuracy and precision, are presented and analysed. Comparison to existing methods is provided where applicable. Any limitations observed during testing and areas for improvement are also discussed.

Chapter 5: Conclusion and Future Work

The final chapter summarizes the project outcomes, contributions, and potential impacts on livestock disease management. Limitations encountered and suggestions for future research and development, such as expanding the dataset or exploring new architectures, are also addressed.

1.10 Summary

This chapter provides an overview of the project, outlining its objectives, methodology, and scope. This chapter introduces the motivation behind the project, by highlighting the significance of an automated bovine disease detection using deep learning methods such as computer vision as an impactful approach. This chapter serves as a foundation, setting the stage for the background study, requirement analysis and design, which will be explored in later chapters.

CHAPTER 2: LITERATURE REVIEW

2.1 Introduction

The literature review is a critical component of this study, providing an organized and synthesized presentation of previous studies relevant to the detection of visible external diseases in bovine animals using advanced computer vision techniques. This chapter explores the progression of disease detection methods, from traditional manual approaches to automated approach powered by machine learning (ML) and deep learning (DL) technologies. By analysing existing models, methodologies, and their applications, this review seeks to establish a strong foundation for the project's proposed solution.

The importance of this review lies in its ability to identify gaps in current research and justify the adoption of specific techniques. Although numerous ML and DL methods have been applied to livestock disease detection, their performance often varies depending on factors such as dataset availability, accuracy, real-time or near real-time inferencing ability, and computational efficiency. These considerations are crucial for selecting a model that aligns with the project's objective of providing a fast and accurate diagnosis system for farmers.

2.2 Overview of Bovine Disease Detection

Livestock diseases pose a significant threat to global food security, economic stability, and public health. These diseases not only result in direct losses due to reduced productivity but also incur indirect costs through treatment and preventive measures. Furthermore, many livestock diseases, including zoonotic ones, have the potential to spread to humans, exacerbating their impact on communities and economies worldwide (Mekonnen et al., 2021).

Traditionally, the detection of diseases in livestock relies primarily on visual inspection by farmers. However, more advanced and accurate diagnosis requires the expertise of veterinarians and laboratory testing. While these methods have been the gold standard, they often face several limitations, including dependence on veterinary expertise, time consumption, and varying presentation of the diseases, which can lead to inaccuracy of the diagnosis itself. These has lead to the emergence of automated systems, in which the possibility of Computer Vision and Artificial Intelligence methods can mitigate the limitations of traditional methods (Neethirajan, 2023).

Machine Learning (ML) and Deep Learning (DL) have emerged as transformative tools in agricultural and veterinary health, offering automated, efficient, and precise solutions for disease detection. These approaches analyze images and data to detect, classify, and localize

diseases, significantly improving diagnostic accuracy and reducing reliance on manual inspection. By applying various computer vision tasks, including object detection, localization, and classification, ML and DL methods provide insights into diagnosing and managing livestock diseases effectively.

The following subsections will delve deeper into existing machine learning and deep learning methodologies for disease detection, discussing their contributions to the field and their relevance to this project's model selection.

2.2.1 Machine Learning Disease Detection

Machine Learning models, such as Support Vector Machines (SVMs), have been widely applied in image classification tasks across various domains. SVMs are particularly effective in solving classification problems by identifying the optimal hyperplane that separates different classes in the feature space, even in high-dimensional datasets. They utilize kernel techniques to transform input data into higher-dimensional spaces, enabling better separation of classes in non-linearly separable scenarios (Nithyashree, 2024).

Shoaib et al. (2023) emphasized the potential of ML methods like SVMs in detecting diseases in animals and plants, offering early and accurate identification that is critical for effective management. This application is particularly valuable in agricultural and veterinary settings. Bello et al. (2021) demonstrated the ability of traditional ML models to classify extracted features into categories, such as healthy or diseased, using structured datasets of plant or animal images.

2.2.2 Deep Learning Disease Detection

Deep Learning models, particularly Convolutional Neural Networks (CNNs), have revolutionized image analysis by learning hierarchical feature representations directly from raw pixel data. This capability allows CNNs to excel in object detection and classification tasks, capturing intricate patterns and features indicative of specific diseases (Purva Huigol, 2024)

Advanced CNN-based architectures, such as Faster R-CNN and Mask R-CNN, enhance detection capabilities by incorporating instance segmentation. This enables the model to predict segmentation masks for each region of interest (ROI), which is especially beneficial in scenarios where multiple diseases may manifest on a single plant or animal. Instance segmentation

improves the localization of diseased regions, offering greater diagnostic precision (Bello et al., 2021).

Shoaib et al. (2023) highlighted the versatility of DL models in both plant and animal health monitoring. For instance, CNNs can analyze images of cattle to detect visible disease symptoms or assess crops for early signs of infection. These applications demonstrate the significant impact of DL in advancing disease detection technologies and enabling timely intervention.

2.3 Related Works On Disease Detection

Deep learning and machine learning family of models including Support Vector Machines (SVMs), Convolutional Neural Networks (CNNs) and You Only Look Once (YOLO) have been extensively applied to disease detection, offering significant advancements in accuracy and efficiency. This section reviews recent studies in this field, highlighting their methodologies, achievements, and limitations.

2.3.1 Convolutional Neural Networks (CNNs)

Naikar et al. (2023) developed a deep learning model to detect Lumpy Skin Disease (LSD) in cattle using transfer learning. Their study employed DenseNet121, VGG16, and VGG19 architectures, trained on a dataset of 1,024 images pre-augmentation and 4,759 images post-augmentation. DenseNet121 achieved the highest performance, with an accuracy of 99.10% and F1-scores of 0.99 for LSD detection. The study excelled in model performance but was limited to a single disease, highlighting a gap in addressing diverse conditions.

Shinde et al. (2024) presented a system for automatic livestock disease detection using CNNs with transfer learning. The study addressed multiple diseases, including Foot-and-Mouth Disease (FMD), Infectious Bovine Keratoconjunctivitis (IBK), and LSD, using a dataset of 2,000 sourced from veterinary databases and agricultural organizations. A fine-tuned VGG16 model achieved an overall accuracy of 88.14%, with FMD classification demonstrating the highest F1-score of 0.85. While the study effectively addressed multi-disease detection, limitations included a relatively small dataset and challenges in generalizability across diverse environments.

Lake et al. (2022) combined deep learning and expert systems to create a diagnostic tool for LSD, ringworm, and warts. Their approach used a CNN trained on a dataset augmented from 208 to 370 images, achieving 95% accuracy within 50 epochs. The study reported F1-

scores of 0.959 for LSD, 0.965 for ringworm, and 0.945 for warts, demonstrating the model's high classification performance. However, the dataset's limited size and dependency on specific image sources restricted its applicability to broader scenarios.

Rony et al. (2021) explored early detection of external cattle diseases, including FMD, LSD, and IBK, using deep learning. Their dataset comprised 600 images pre-augmentation and 6,000 post-augmentation. The CNN algorithm achieved 92% accuracy, with F1-scores of 0.91 (FMD), 0.89 (LSD), and 0.90 (IBK). Among alternative architectures tested, Inception-V3 outperformed others with a 95% accuracy. This study demonstrated strong multi-disease classification capabilities, although limitations in dataset diversity posed challenges for generalization.

Gupta et al. (2023) developed an AI-based system for the early detection of Infectious Bovine Keratoconjunctivitis (IBK) to prevent outbreaks and improve animal welfare. Using Regional Convolutional Neural Networks (RCNNs) implemented in PyTorch, the system detected diseases in cattle muzzles and classified them based on image inputs. The dataset included 3,000 images of healthy beef cattle and 1,500 images of diseased cattle, with specific cases of IBK and bovine respiratory disease. The model achieved a sensitivity of 99.4% and specificity of 97.6%, identifying 169 out of 170 IBK cases before veterinarians. However, the system's exclusive training on beef cattle limited its generalizability to dairy cattle. Biological variations in muzzle symptoms also posed a challenge for more precise diagnostics.

2.3.2 Hybrid Model (CNN-SVM)

Mehta & Khurana (2024) proposed a hybrid CNN-SVM model for detecting FMD, Mastitis, and Bovine Respiratory Disease (BRD). A dataset of 1,500 images was used, and the hybrid model achieved an impressive accuracy of 91%, outperforming the standalone CNN model with 85.2% accuracy and SVM model with 82.3% accuracy. The hybrid approach also demonstrated a high recall of 90.5%, emphasizing its improved feature extraction and classification accuracy. Despite its success, the study's dataset size and lack of variability in disease categories, limited the model's applicability in a more diverse conditions with varying symptoms.

Mehta and Khurana in 2024 proposed a hybrid CNN-SVM model for detecting Foot and Mouth Disease, Mastitis, and Bovine Respiratory Disease. The model was trained on a dataset comprising 1,500 images and achieved an impressive accuracy of 91 percent,

outperforming the standalone CNN with 85.2 percent accuracy and SVM with 82.3 percent accuracy. Furthermore, the hybrid model demonstrated a high recall of 90.5 percent, highlighting its enhanced feature extraction and classification capabilities. However, the study's effectiveness was limited by the small dataset size and lack of variability, which reduced the model's applicability in more diverse real-world scenarios.

2.3.3 You Only Look Once (YOLO)

Shahbaz et al. (2024) developed a two-stage vision system to detect digital dermatitis (DD) lesions in dairy cattle, aimed at preventing lameness. The first stage employed YOLOv5 for hoof detection, and the second stage classified DD lesions into six categories, including different lesion stages. The dataset comprised images of hooves from various farms, labelled for front-right (FR), front-left (FL), rear-right (RR), and rear-left (RL) hooves, as well as the lesion stages. YOLOv5s achieved an average precision of just under 0.99 for RR and RL hooves and above 0.90 for FR and FL hooves. However, lesion classification precision was comparatively lower, exceeding 70% for most categories. Strengths included high precision in hoof detection and early intervention potential, but challenges such as misclassification between lesion stages and dataset size impacted performance.

2.4 Comparison on Reviewed Disease Detection Works

CNNs, SVMs, YOLO family models have shown promising results in detecting and classifying diseases in, particularly in the context of medical imaging and computer vision applications. The comparison of related works that has been reviewed are listed in Table 2.1.

Table 2.1: *Summary of reviewed related works on disease detection*

Citation	Algorithm	Dataset Characteristics	Image Processing	Real Time Operation	Strength	Limitations	Accuracy
Naikar et al. (2023)	DenseNet121, VGG16, VGG19, CNN	Approximately balanced dataset of 4759 images (2465 LSD, 2294 normal)	Data augmentation (random flips, rotations, cropping, resizing, color jittering)	The system processes images in real-time by leveraging pre-trained models and a user-friendly GUI that allows users to upload images and receive instant classification results	High accuracy, early detection, user-friendly interface	No mention	DenseNet121: 99.10%, VGG16: 97.85%, VGG19: 95.69%, CNN: 91.38%
Shinde et al. (2024)	VGG16 CNN	2000 images (1600 training, 400 testing)	Data augmentation (translation, rotation, zoom, flipping)	The system processes images in real-time by leveraging pre-trained VGG16 model and a user-friendly interface	High accuracy, effective disease detection, user-friendly interface	No mention	88.14%

				that allows users to upload images and receive instant classification results			
Lake et al. (2022)	CNN	Images collected from various sources, transformed and augmented	Data augmentation (flipping, zoom, shear, rotation)	The system processes images in real-time by leveraging a hybrid diagnosis system that integrates image classification results, location, and text information	Rapid and effective diagnosis, user-friendly interface	No mention	95%
Rony et al. (2021)	CNN	600 images (250 FMD, 200 LSD, 150 IBK)	Data augmentation (shifting, resizing, rotating, flipping, shearing,	The system processes images in real-time by leveraging pre-trained models and a user-friendly interface that allows	High accuracy, early detection, user-friendly interface	Limited raw data, potential decrease in accuracy and increase in loss during iterations	CNN: 95%, VGG-16: 95%, Inception-V3: 98%

			zooming, brightness and contrast enhancement)	users to upload images and receive instant classification results			
Mehta & Khurana (2024)	Hybrid Approach: CNN for feature extraction, SVM for classification	1500 images of cow diseases (FMD, mastitis, BRD)	Data augmentation (flipping, zoom, shear, rotation)	The system processes images in real-time by leveraging a hybrid diagnosis system that integrates image classification results, location, and text information	High accuracy, effective disease detection, user-friendly interface	No mention	Hybrid model: 91%, CNN: 85.2%, SVM: 82.3%
Gupta et al. (2023)	RCNN	3000 healthy, 1500 sick (800 IBK, 600 BRD, and 100 Foot rot, uterus infection etc).	Data augmentation (flipping, zoom, shear, rotation)	The system processes images in real-time by leveraging a hybrid diagnosis system that integrates image classification	High sensitivity (99.4%) and specificity (97.6%) in early detection of IBK	Limited to beef cattle, requires additional training for dairy cattle	Sensitivity: 99.4%, Specificity: 97.6%

				results, location, and text information			
Shahbaz et al. (2024)	YOLO	Images collected from four different farms, filtered for high quality	Data augmentation (flipping, zoom, shear, rotation)	The system processes images in real-time by leveraging a two-stage vision system that detects hooves and then classifies DD lesions using a user-friendly interface, through a single forward pass during inferencing.	High precision in hoof and lesion detection, early intervention, user-friendly interface	Limited to dairy cattle, requires high-quality images	Hoof detector: 96% precision, Lesion detector: 75% precision

2.5 Model Comparison

Table 2.2: *Comparison of models used in the reviewed related works*

Features		CNNs	SVMs	YOLO
Dataset Requirements	Strengths	Performs exceptionally with large, diverse datasets. Transfer learning can reduce data needs (Naikar et al., 2023)	Performs well on smaller datasets. Requires less data compared to DL models (Mehta & Khurana, 2024)	Moderate dataset requirements. Pre-trained weights enable fine-tuning with smaller datasets (Shahbaz et al., 2024)
	Limitations	Requires substantial labeled data to achieve high accuracy.	Performance is highly dependent on quality of extracted features.	May require additional data for complex multi-object scenarios.
Accuracy	Strengths	High accuracy in classification and detection tasks. Specialized architectures like DenseNet121 achieve 99.10% accuracy (Naikar et al., 2023)	Reliable accuracy for simpler tasks. Achieves competitive results when hybridized with DL models (Mehta & Khurana, 2024)	High accuracy in object detection and localization. YOLOv5 and YOLOv8 excel in precision for disease detection (Sapkota & Karkee, 2024)
	Limitations	Accuracy may drop with insufficient dataset diversity or complex overlapping objects.	Limited accuracy in complex image analysis and multi-class tasks.	May underperform in pixel-level detail compared to segmentation-specific models like Mask R-CNN.

Computational Efficiency	Strengths	Optimized architectures (e.g., MobileNet) reduce computational load during inference (Lake et al., 2022)	Lightweight and computationally efficient. Suitable for systems with limited hardware (Yang et al., 2021)	Optimized for real-time performance. Balances accuracy and efficiency with moderate computational needs (Sapkota & Karkee, 2024)
	Limitations	Training is computationally intensive and may require GPUs for efficiency.	Struggles with scalability for complex datasets or real-time tasks	May require more resources than SVMs for large-scale datasets or fine-tuning.
Real-Time Inferencing	Strengths	Fast inference for lightweight CNNs like MobileNet. Ideal for non-time-critical applications.	N/A.	Unmatched real-time capability. Single-stage design enables rapid detection, e.g., YOLOv8 processes images in ~3.3ms (Sapkota & Karkee, 2024)
	Limitations	Multi-stage architectures (e.g., Faster R-CNN) are unsuitable for real-time tasks (Bello et al., 2021)	Not designed for real-time tasks due to slower classification speeds.	May sacrifice some accuracy compared to multi-stage detection models.

performance through innovative enhancements that surpass previous versions on benchmark datasets COCO (Common Objects in Context). COCO is a widely used dataset in the field of computer vision, containing over 200,000 labelled images across 80 object categories, designed to facilitate the evaluation of object detection, segmentation, and image captioning algorithms. The latest iteration, YOLOv11, represents a significant leap forward, offering superior accuracy, faster inference times, and advanced capabilities such as instance segmentation for RGB images (Sapkota & Karkee, 2024). These improvements make YOLOv11 an ideal choice for this project, as its cutting-edge features align with the requirements for accurate detection, localization, and classification of visible bovine diseases. By leveraging YOLOv11, this project can benefit from robust real-time performance and precision diagnostics.

2.6.2 Variants

Model	Filenames	Task	Inference	Validation	Training	Export
YOLO11	yolo11n.pt yolo11s.pt yolo11m.pt yolo11l.pt yolo11x.pt	Detection	✓	✓	✓	✓
YOLO11-seg	yolo11n-seg.pt yolo11s-seg.pt yolo11m-seg.pt yolo11l-seg.pt yolo11x-seg.pt	Instance Segmentation	✓	✓	✓	✓
YOLO11-pose	yolo11n-pose.pt yolo11s-pose.pt yolo11m-pose.pt yolo11l-pose.pt yolo11x-pose.pt	Pose/Keypoints	✓	✓	✓	✓
YOLO11-obb	yolo11n-obb.pt yolo11s-obb.pt yolo11m-obb.pt yolo11l-obb.pt yolo11x-obb.pt	Oriented Detection	✓	✓	✓	✓
YOLO11-cls	yolo11n-cls.pt yolo11s-cls.pt yolo11m-cls.pt yolo11l-cls.pt yolo11x-cls.pt	Classification	✓	✓	✓	✓

Figure 2.2: Breakdown of different YOLO variants

Figure 2.2, also sourced from Ultralytics (2024), highlights the specialized variants introduced in YOLOv11, each designed to excel in specific computer vision tasks by building on the advancements of its predecessors. These variants include detection, instance segmentation, pose/keypoint detection, oriented detection, and classification. This versatility allows YOLOv11 to cater to a wide spectrum of applications, from real-time detection to more complex segmentation and localization tasks. For this project, which focuses on the detection and localization of visible bovine diseases, the Detection variant (YOLO11) is the most

appropriate. It enables the model to efficiently handle tasks such as identifying, localizing, and classifying disease-affected regions, ensuring optimal performance aligned with the project's requirements.

2.6.3 Size

Table 2.3: *Breakdown of different YOLO sizes and its performance metrics*

Model	size (pixels)	mAP ^{val} 50-95	Speed CPU ONNX (ms)	Speed T4 TensorRT10 (ms)	params (M)	FLOPs (B)
YOLO11n	640	39.5	56.1 ± 0.8	1.5 ± 0.0	2.6	6.5
YOLO11s	640	47.0	90.0 ± 1.2	2.5 ± 0.0	9.4	21.5
YOLO11m	640	51.5	183.2 ± 2.0	4.7 ± 0.1	20.1	68.0
YOLO11l	640	53.4	238.6 ± 1.4	6.2 ± 0.1	25.3	86.9
YOLO11x	640	54.7	462.8 ± 6.7	11.3 ± 0.2	56.9	194.9

Table 2.3 listed the detection variant of YOLOv11, which is available in multiple sizes—Nano (n), Small (s), Medium (m), Large (l), and Extra-Large (x)—distinguished by the suffix in their names. Each size is tailored for specific use cases and computational capabilities, offering flexibility based on project requirements. Key considerations when selecting the model size include the project's objectives, computational constraints, and dataset characteristics.

For this project, where real-time or near real-time performance is preferred, smaller models like YOLOv11n or YOLOv11s are more suitable. These models provide faster inference speeds and lower computational demands, making them ideal for deployment on portable devices like mobile phones or edge devices that farmers can use for disease detection. In resource-constrained environments, the smaller memory footprint and computational efficiency of these models ensure seamless operation without compromising usability.

Dataset size and quality also play a crucial role in model selection. Smaller models are less prone to overfitting when trained on limited datasets, making them a better choice in scenarios where data is scarce. Conversely, larger models such as YOLOv11l or YOLOv11x excel when trained on large and diverse datasets, offering higher accuracy and better generalization. However, these larger models come with increased computational requirements, making them more suitable for server-based or high-performance GPU deployments.

Ultimately, the choice of model size involves balancing inferencing speed, accuracy, computational resources, and dataset limitations. By carefully aligning these factors with the project's goals, an optimal YOLOv11 variant can be selected to achieve efficient and reliable disease detection.

Table 2.4: *Pros and cons in different sizes of YOLOv11*

Model Size	Pros	Cons
Nano/Small	Fast, resource-efficient, suitable for real-time and mobile applications	Lower accuracy and feature extraction capabilities
Medium	Balanced speed and accuracy, good for moderate hardware and datasets	Slightly slower inference than small models
Large/X-Large	High accuracy, handles complex tasks with overlapping objects	Requires substantial computational resources and larger datasets

Table 2.4 provides comparison of varying sizes of YOLOv11 models, where different sizes are tailored to different computational and application requirements. Larger models are suited for complex tasks due to their higher accuracy, but they come with trade-offs, such as the need for substantially larger datasets and heavier computational power during training and inference. Conversely, smaller models are more efficient in resource-constrained environments and perform well with limited datasets, making them ideal for real-time or near real-time applications. For this project, the Small size of YOLOv11 will be utilized for efficient and accurate bovine disease detection.

2.7 Summary

This literature review has examined the progression of livestock disease detection methods, from traditional manual techniques to state-of-the-art computer vision approaches. While traditional methods often rely on time-consuming and expertise-dependent processes, modern machine learning and deep learning models have introduced transformative capabilities, offering greater accuracy, efficiency, and scalability.

Among these advancements, the YOLO framework stands out as a robust solution for real-time detection and localization of diseases. Its ability to process multiple disease manifestations within a single image, coupled with its relatively low computational cost, makes it an ideal choice for this project. The latest iteration, YOLOv11, further enhances these capabilities with improved performance and reduced latency, addressing the specific requirements of this application.

By leveraging YOLO's strengths, this project aims to develop an efficient and accessible diagnostic tool that can detect visible bovine diseases accurately and in near real-time. This solution has the potential to significantly improve animal health outcomes by enabling early detection and timely intervention, ultimately contributing to more sustainable and effective livestock management practices.

CHAPTER 3: METHODOLOGY

3.1 Introduction

The methodology for this project follows a systematic approach, starting from dataset preparation to deploying the trained YOLO model onto edge devices. Figure 3.1 provides a simplified workflow outlining the four main processes: Dataset Preparation, Training, Running Inference, and Deployment. This chapter focuses on detailing each stage, starting with Dataset Preparation, as it is the foundation for developing an effective YOLO model.

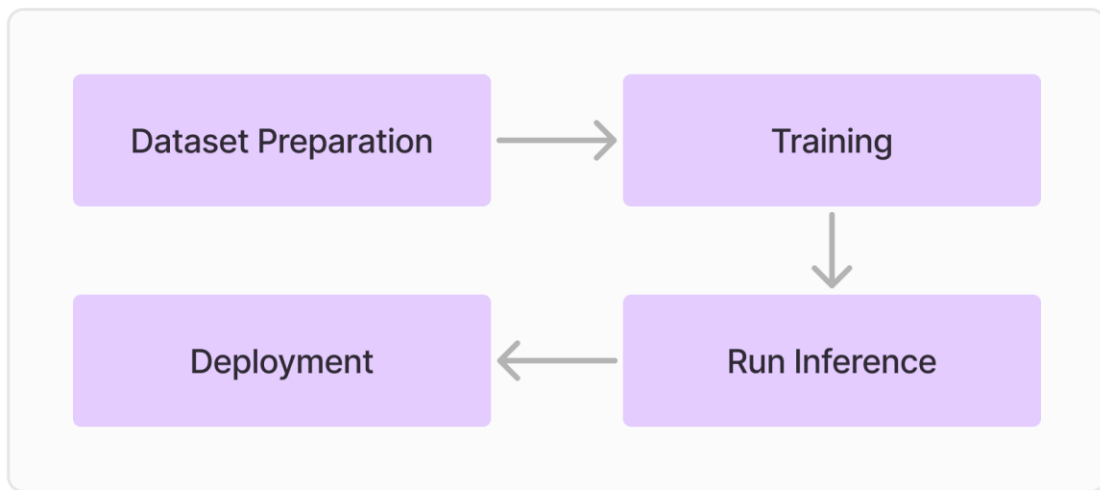


Figure 3.1: *General flow of the proposed approach*

3.2 Dataset Preparation

The process of dataset preparation is crucial to ensure the YOLO model achieves high accuracy and performance. Figure 3.2 illustrates the dataset preparation workflow, which consists of four key steps, Collection, Annotation, Organization, and Exporting Data. Each step is elaborated below.

Roboflow (n.d.) facilitates this workflow, where it is used due to its powerful features and ease of use. Its intuitive interface simplifies dataset creation and management, while the annotation tools reduce the time and effort needed for labelling. The platform has the option to automate the process of preprocessing and augmentations, while ensuring dataset quality and diversity. Its ability to integrate and merge existing datasets further enhances efficiency, while flexible export options allow for seamless integration with various training pipelines. By centralizing these processes, Roboflow (n.d.) minimizes errors, such as annotation mistakes or format inconsistencies, ensuring a robust and well-prepared dataset for this training.

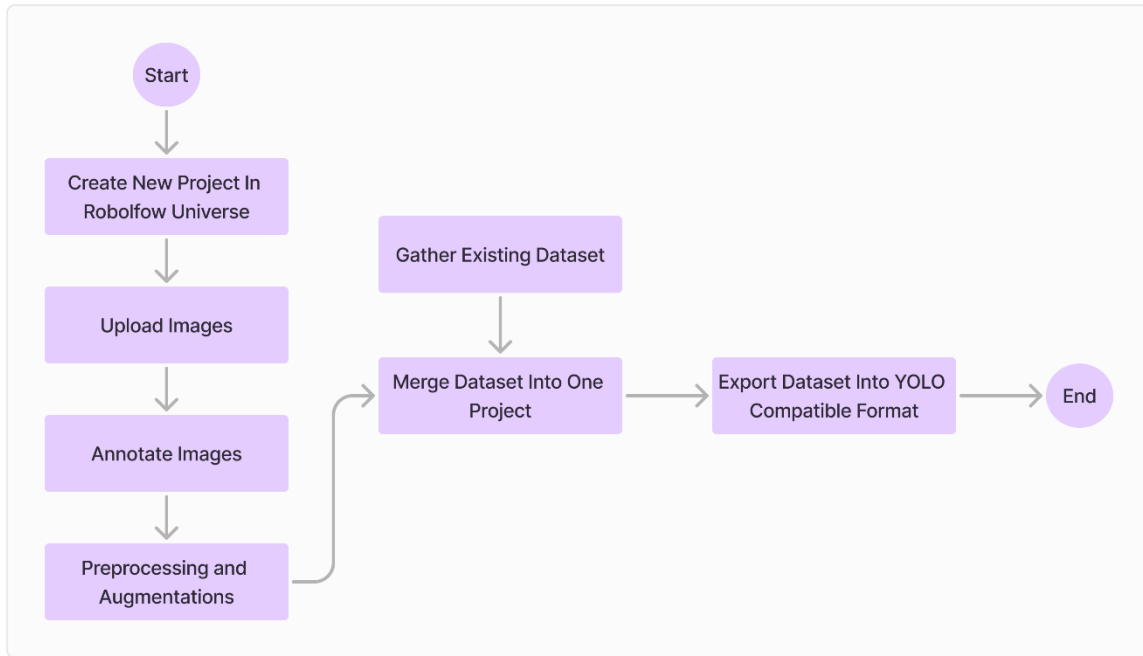


Figure 3.2: *Proposed dataset preparation process flow*

3.2.1 Collection

The first step in dataset preparation involves gathering images from various sources. The images for this project were collected from a range of sources, including raw field-captured data and pre-existing datasets obtained from platforms such as Kaggle (n.d.) and Roboflow (n.d.). This approach ensures a diverse and representative dataset, improving the robustness of the model. The variety in image sources helps the YOLO model generalize effectively for real-world applications.

3.2.2 Annotation

After collecting the images, annotation is performed to label areas of interest. This step is essential for creating the labeled datasets required for supervised learning in object detection. The Roboflow (n.d.) platform was chosen for its efficient annotation tools and user-friendly interface.

Using Roboflow (n.d.), a project workspace was set up as an Object Detection type. This allowed bounding boxes to be drawn around areas of interest in each image. The annotation process involved manually defining rectangular regions around visible signs of diseases such as lesions, swelling, or other abnormalities on the cattle. Roboflow's intuitive interface

streamlined the annotation process, while features like automatic bounding box suggestions helped improve efficiency and consistency.

The resulting annotated dataset serves as the foundation for training the object detection model, ensuring it can accurately identify and classify disease-related features.

3.2.3 Organization

Once annotated, the dataset is organized to prepare it for model training. This involves splitting the dataset into three subsets: training, validation, and testing.

- Training set: Used to train the YOLO model by learning patterns and features in the data.
- Validation set: Used to tune hyperparameters and prevent overfitting during training.
- Testing set: Used to evaluate the model's performance after training is complete.

These subsets usually are split based on the original dataset, not the augmented size. Hence to ensure quality trained model with adequate amount of validation and test set, the dataset will be divided into 80% training, and 10% each for validation and train set. Proper organization ensures that the model can generalize well to unseen data while maintaining robust performance metrics.

3.2.4 Export Dataset

The final step in dataset preparation is exporting the curated dataset in a format compatible with the YOLO framework. Roboflow (n.d.) simplifies this step by supporting various export options, including YOLO-specific formats. This ensures that all annotations, such as bounding boxes and object classes, are structured appropriately for seamless integration into the YOLO training pipeline.

3.3 Training

The training process for the YOLO model follows a structured workflow that includes setting up the training environment, configuring training parameters, and iterating the process to achieve optimal performance metrics. This phase is visually outlined in Figure 3.3 and ensures efficient and systematic training of the model.

The training phase aims to develop a high-performing YOLOv11 model capable of real-time disease detection with high accuracy, precision, mean Average Precision (mAP), and recall. This process will involve iterative adjustments to achieve optimal performance metrics.

To initiate training, the project will use Google Colab as the primary environment due to its accessibility and ability to provide cloud-based computational resources, including an NVIDIA Tesla T4 GPU. The training will leverage preprocessed datasets prepared in Roboflow (n.d.), augmented to address size limitations and class imbalances. A minimum dataset size of 1,000 images per disease class will ensure robustness and diversity.

Training parameters, such as batch size, learning rate, and epochs, will be configured iteratively. The model's performance will be continuously evaluated against metrics like mAP@50%, precision, and recall. These iterative cycles will include hyperparameter tuning to refine the model until the desired performance is achieved. The final trained model will be exported in formats compatible with deployment, ensuring readiness for real-world application. This phase is visually outlined in Figure 3.3 and ensures efficient and systematic training of the model.

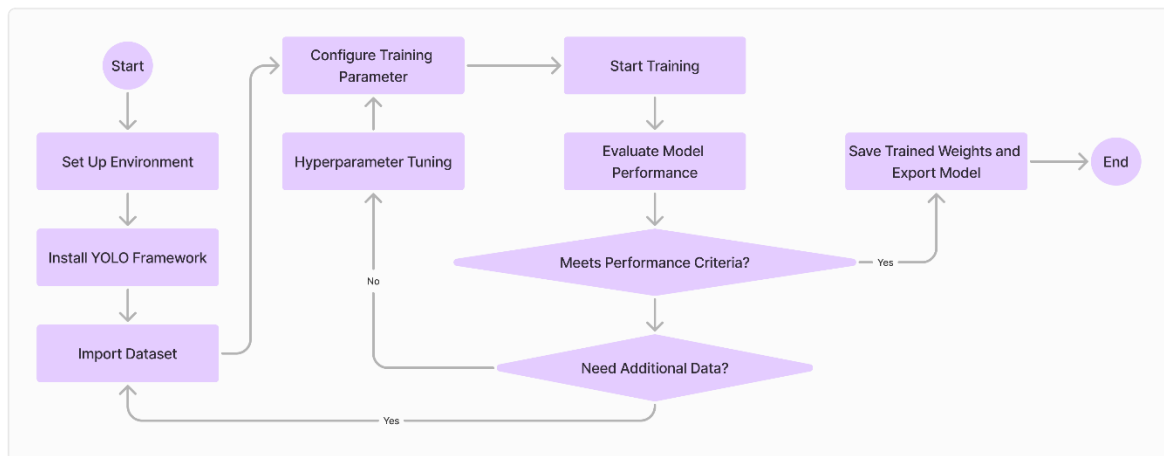


Figure 3.3: *Proposed training process flow*

3.4 Inferencing

Inferencing involves testing the trained YOLOv11 model on unseen images and videos to evaluate its accuracy, and speed. Initial inferencing will be conducted in Google Colab (Google, n.d.).

The inferencing process will use data that mimics real-world conditions to ensure robustness. In this phase, the system will detect and localize visible external diseases, producing

outputs such as bounding boxes, confidence scores, and class labels. While real-time applications such as live video feed processing are not the primary focus at this stage, tools like OpenCV may be integrated to test dynamic frame-by-frame predictions.

Future testing may extend to dedicated devices, such as mobile phones or microcontrollers (e.g., ESP32 or Raspberry Pi), depending on their feasibility and compatibility. The primary goal is to validate the model's real-time detection capabilities and identify potential areas for further optimization.

3.5 Deployment

After training and evaluation, the developed YOLOv11 models were deployed using Roboflow's hosted model services (Roboflow, n.d.). This method provides a cloud-based solution for storing, managing, and serving models, eliminating the need for manual hosting infrastructure or local deployment on high-performance machines. Upon export from the training environment, the models were uploaded directly to Roboflow (n.d.), making them accessible via web APIs and ready for integration with frontend applications or further testing environments.

This deployment strategy was chosen for its simplicity, scalability, and support for collaborative development. By using Roboflow's hosting capabilities (Roboflow, n.d.), the project ensures that the trained models are readily accessible and easily maintained. This also enables future extension of the system into mobile applications or web-based interfaces, allowing users such as farmers and livestock managers to perform disease detection through an internet-connected device or, eventually, in an offline-capable solution.

Although a standalone application was not developed in this phase, the choice of Roboflow (n.d.) as the deployment platform maintains flexibility and readiness for future system development. This approach supports continued research and potential translation into practical diagnostic tools in agriculture, particularly in areas with limited veterinary access.

3.6 Summary

This chapter outlines the methodology for developing a YOLO-based model aimed at detecting visible bovine diseases, structured across four key phases. The first phase, Dataset Preparation, involved collecting images from diverse sources and annotating them using Roboflow (n.d.). These images were then divided into training, validation, and testing subsets in an 80:10:10 ratio, and exported in a YOLO-compatible format to facilitate seamless training

integration. In the Training phase, the model was trained using Google Colab, leveraging resources such as the NVIDIA Tesla T4 GPU. Throughout training, key parameters were iteratively fine-tuned to optimize performance metrics such as precision, recall, and mean Average Precision (mAP), while data augmentation techniques were applied to mitigate dataset limitations and enhance variability.

The third phase, Inferencing, focused on testing the trained model on unseen data within the Google Colab environment. Although inferencing was conducted in a controlled setting, considerations were made for future deployment on mobile devices, with performance indicators like frames per second (FPS) and latency proposed to evaluate real-time viability. Finally, the Deployment phase involves utilizing Roboflow's hosted model services, offering a straightforward and scalable solution for sharing and integrating the model. This setup allows for future development into user-friendly applications that enable accessible agricultural disease detection through the cloning of models and datasets. Altogether, this methodology provides a comprehensive and efficient pipeline tailored to the practical needs of the agricultural sector.

CHAPTER 4: IMPLEMENTATION AND DISCUSSION

4.1 Overview

This chapter presents the implementation and discussion of the disease detection model for bovine animals, by leveraging YOLOv11. This model is to accurately detect diseases including LSD, FMD, IBK, and Ringworm.

4.2 Dataset

4.2.1 Collection and Preparation

A new project was first established on (Roboflow, n.d.) as the centralized platform for organizing, curating, and managing the dataset. Images used for this dataset were sourced from multiple origins to ensure a diverse and representative collection. These sources included extracted frames from videos, publicly available datasets such as those found on (Kaggle, n.d.) and (Roboflow, n.d.), as well as scattered images retrieved from various internet sources. The overall process of collection, refinement, and annotation is illustrated in Figure 4.1.

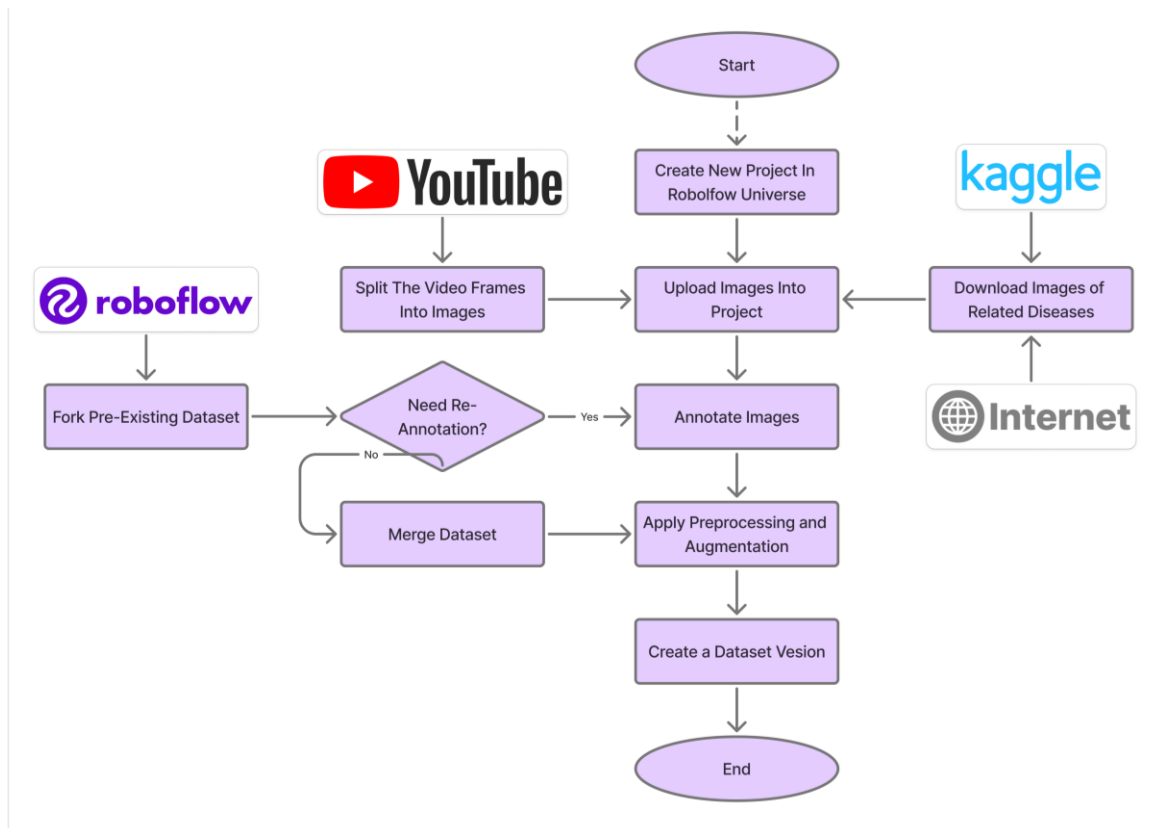


Figure 4.1: *Implemented Dataset Preparation Flow*

To ensure the quality and usability of the dataset for training robust models, several key criteria were considered during image selection. First and foremost, all images were manually

screened for sufficient resolution and clarity, filtering out those that were blurry, noisy, or pixelated. This is critical to ensure that the visual features associated with clinical signs of disease can be recognized clearly by both human annotators and the machine learning models.

Furthermore, the contextual realism of each image was taken into account. Images were preferred if they were captured in practical field conditions such as pastures, barns, or holding pens. Though clinical or artificial settings were minimized, it is still important to give more context of the disease signs. This is to promote generalizability and performance in real-world deployments. Images that were excessively cropped or tightly framed were avoided, as such framing can obscure important contextual cues or lead to incomplete feature representation.

Diversity of the dataset was another central objective. Images were collected across a range of angles, viewpoints, backgrounds, and lighting conditions, to better reflect the variety encountered in real-life scenarios. This includes both full-body views and close-up shots that emphasize visible signs of disease on the skin, improving the model's ability to detect fine-grained features at varying scales.

Consistency in annotation was treated with high priority. Images were annotated using precisely tight bounding boxes that fully enclosed the visible disease signs while excluding unnecessary background or unrelated body parts. Ambiguous or uncertain regions were omitted from annotation to maintain data integrity. For externally sourced images that were already annotated, such as those from other datasets, all bounding boxes were carefully reviewed. Incorrect class labels, box coordinate errors, or annotation format mismatches were rectified to maintain consistency across the dataset.

4.2.2 Preprocessing

A set of preprocessing steps was applied uniformly across all images. This includes auto-orientation, to correct for any metadata-related rotation issues, ensuring consistent image alignment. Resizing to a fixed resolution (640×640), which is a common practice in computer vision tasks like object detection. It ensures uniform input dimensions for the model, simplifies batch processing, and helps the model learn spatial patterns more efficiently. Linear contrast stretching, which expands the dynamic range of pixel intensities. This enhances the visibility of features in underexposed or low-contrast areas of the image without introducing artificial distortions. It is particularly helpful in medical or agricultural imagery, where subtle visual cues may be crucial.

4.2.3 Curated Dataset

Upon completion of the dataset, the gathered images totalled in 8,282. These images were consisting of images of disease and non-disease type. In one image, there might be one or more labels of the same or different classes, hence Table 4.1 illustrates the summary of the curated dataset focusing on the label counts instead of image counts.

Table 4.1: *Annotations in collected images*

Dataset Name	Class Type	Class Name	Labels
Bovine Disease Detection	Non-Disease	Healthy Skin	4,483
		Healthy Eye	615
		Healthy Mouth	286
	Disease	LSD	1,961
		FMD	276
		IBK	616
		Ringworm	2,159
	Total		10,397

This dataset, however, exhibits a significant imbalance in the number of samples across different disease classes. Such imbalance is undesirable for training a computer vision model, as it can lead to biased learning and reduced overall performance. Nevertheless, this imbalanced dataset will serve as a baseline for benchmarking purposes. In the following sub-sections, strategies to mitigate the impact of dataset imbalance will be discussed.

4.3 Dataset Imbalance Mitigation Strategy

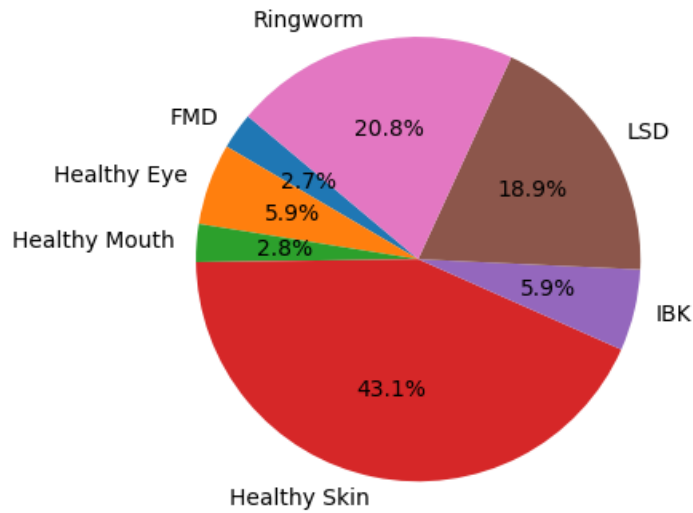


Figure 4.2: *Class Label Distribution Pie Chart for Baseline Dataset*

The curated dataset reveals a striking imbalance between some classes as revealed by Figure 4.2. Diseases such as IBK and FMD, which primarily manifest on the head, are represented in only a few hundred samples. In contrast, diseases affecting the body, such as LSD and Ringworm, are represented in the thousands. This disparity presents an opportunity to split the dataset and train separate models for each region, potentially improving performance by reducing class imbalance effect

To address this, a multi-model approach is proposed. It involves training three distinct models. The first model detects and classifies body parts (head or body), and based on the detected region, the appropriate disease detection model is applied. Cropped and zoomed images of the head or body are then passed to their respective specialized models to perform disease detection. Finally, the predictions are mapped back onto the original image for visualization. This process is illustrated in Figure 4.3. The labels for the body part detection model were created as an extension of the original dataset by manually annotating head and body regions from existing images. As a result, the total number of labels differs from the original disease dataset.

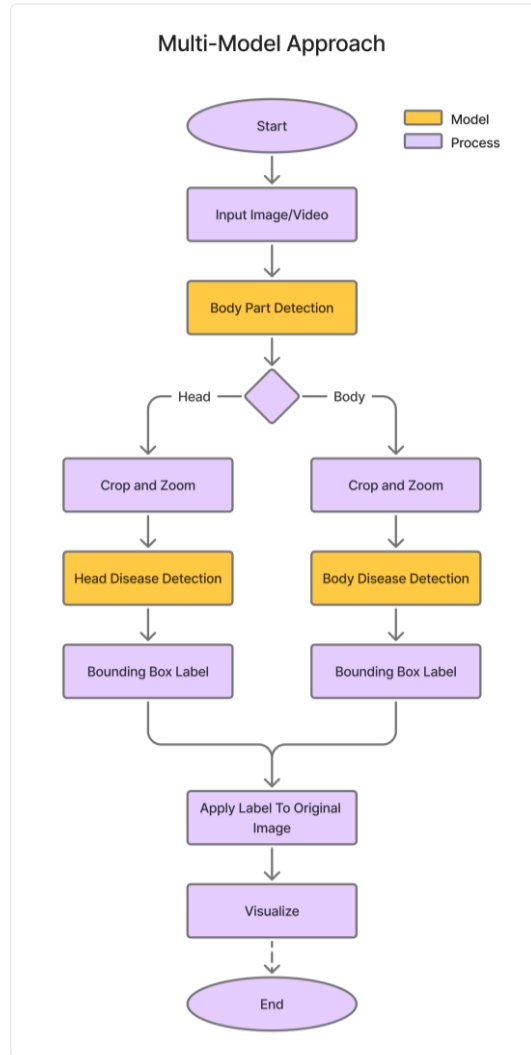


Figure 4.3: *Process flow of the multi model approach*

In preparation of this multi model approach, three dataset versions have been derived from the baseline dataset. To further enhance the model generalization, a set of data augmentation techniques has been applied. These augmentations introduced controlled distortions in the dataset, allowing the model to learn more robust features representations under diverse conditions. The augmentation was performed using Roboflow’s (Roboflow, n.d.) built-in augmentation pipeline. Table 4.2 summarizes the augmentations applied

Table 4.2: *Augmentations applied to all dataset in the multi model approach*

Augmentation Type	Range/Setting	Purpose
Horizontal Flip	Enabled	Simulate left-right orientation changes

90° Rotation	Clockwise, Counter-Clockwise	Mimic varying camera angles
Rotation	Between -15° and +15°	Introduce slight camera tilt
Shear	±10° Horizontal, ±10° Vertical	Distort perspective slightly to reflect real-world variance
Saturation Adjustment	Between -25% and +25%	Handle different lighting or contrast environments
Brightness Adjustment	Between -20% and +20%	Simulate variations in ambient lighting
Exposure Adjustment	Between -10% and +10%	Compensate for under/overexposure

Upon completion of splitting the original baseline dataset and applied data augmentations, the final version of the three datasets can be summarized in Table 4.3

Table 4.3: *Summary of the dataset to be used in multi model approach*

Dataset Name	Class Name	Pre-Augmentation	Post-Augmentation
Bovine Body Part	Head	3,506	17,530
	Body	3,591	17,955
Bovine Head Disease Detection	Healthy Eye	615	3,075
	Healthy Mouth	286	1,430
	FMD	276	1,380
	IBK	616	3,080
Bovine Body Disease Detection	Healthy Skin	4,483	22,415
	LSD	1,961	9,805
	Ringworm	2,159	10,795

	Total	17,493	87,465
--	-------	--------	--------

The training, optimisation and finetuning will be discussed in the next sub-section, where the performance will also be compared to the baseline model.

4.4 Training

Training a model in this project followed a systematic development flow. This approach allows for development of multiple models, trained on varying dataset, allowing for development of different models for different purposes, which were illustrated in Figure 4.4.

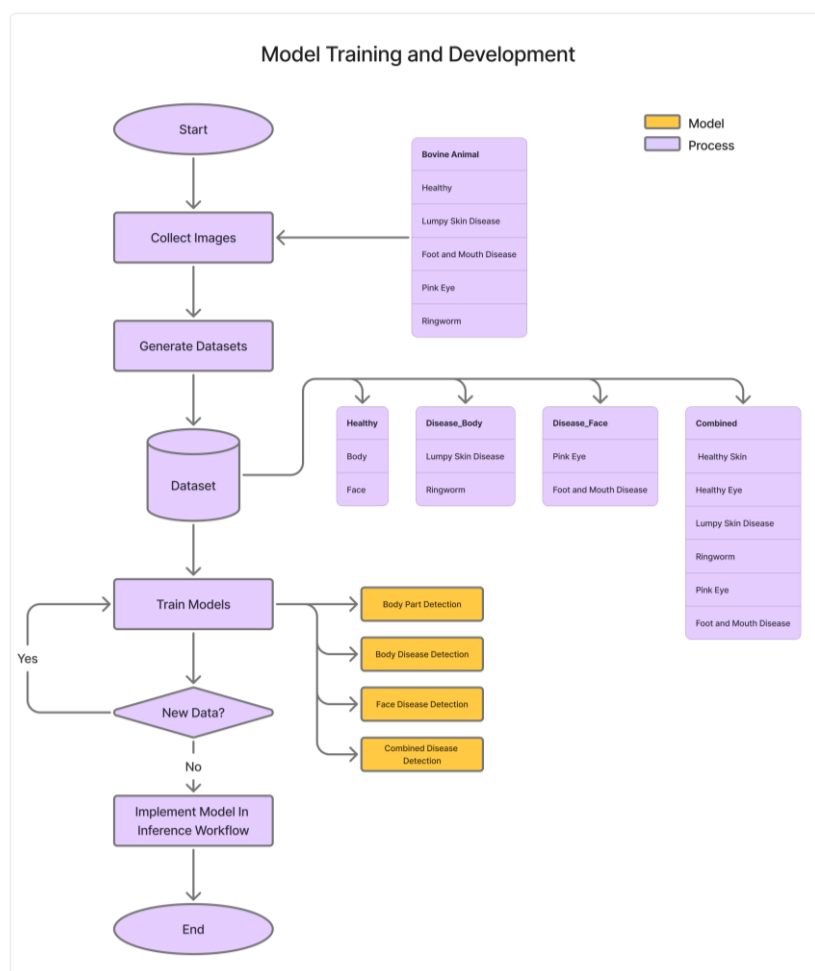


Figure 4.4: *Model Training and Development Process Flow*

Upon setting up a Google Colab Notebook (Google, n.d.) environment, including the required dependencies, the custom dataset that has been curated are imported using Roboflow's

API to give access to the dataset without explicit downloading and uploading into the environment, streamlining the process for any further iterations of the dataset.

To initiate model training, a Google Colab Notebook (Google, n.d.) environment was set up due to its support for GPU-accelerated computation. The necessary dependencies were installed, including Roboflow's Python package. This facilitates direct dataset import into the environment using Roboflow's API to give access to the dataset without explicit downloading and uploading into the environment, streamlining the process for any further iterations of the dataset. This improves efficiency and ensures consistency across multiple development iterations.

Below is the code snippet used to import the dataset into the Colab workspace:

```
from roboflow import Roboflow
rf = Roboflow(api_key="api_key")
project =
rf.workspace("project_id").project("project_path
version = project.version(1)
dataset = version.download("yolov11")
```

Once the dataset is successfully imported and configured, the model training process can begin. Training is conducted using the YOLOv11 architecture. Parameters such as model size (e.g., yolov11s.pt), learning rate schedule (cos_lr), image resolution (imgsz=640), number of epochs, and early stopping patience are defined prior to training.

The following command was used to start the training process:

```
!yolo task=detect mode=train model=yolov11s.pt
data={dataset.location}/data.yaml epochs=100
patience=10 cos_lr=True imgsz=640 plots=True
```

After training is completed, the model is ready for deployment. To streamline this step, the trained model can be uploaded directly back to Roboflow using their deployment API. This facilitates further evaluation, sharing, or integration into application interfaces.

Deployment can be executed with the following code:

```
project.version(dataset.version).deploy(model_type="yolov11", model_path=f"{HOME}/runs/detect/train/")
```

This end-to-end workflow ensures that model training is both reproducible and adaptable, allowing future enhancements and extensions, such as retraining on updated datasets or integrating with edge devices.

4.5 Optimisation and Finetuning

To optimize the model's performance for accurate and reliable detection of bovine diseases, several hyperparameter tuning strategies were explored. The tuning process was primarily focused around optimizer selection, as it significantly affects other training dynamics such as weight updates, convergence rate, and learning rate behavior. Manually adjusting these parameters in isolation often led to unstable performance and slower convergence.

Multiple configurations were tested, varying the number of epochs, learning rates (lr0, lrf), early stopping patience, and optimizer types. After some experimentation, a final configuration was selected based on its consistent performance across mean Average Precision (mAP), precision, and recall, as well as reasonable training duration. The YOLO framework's automatic parameter configuration was retained for several settings, which proved effective in maintaining efficiency and performance.

To avoid overfitting, early stopping with a patience value of 10 was implemented, allowing the training to terminate automatically if performance plateaued over several epochs. Table 4.4 below summarizes the final hyperparameter configuration used in this project:

Table 4.4: *Final Hyperparameters Used In Training*

Hyperparameter	Value	Description
Epochs	100	Sufficient for convergence
Cosine Learning Rate	True	Helped achieve smooth and effective learning rate decay
Optimizer	AdamW	Outperformed SGD and Adam in terms of performance and learning stability

Patience	10	Prevented overfitting and unnecessary training beyond plateaued progress
Other parameters	Auto	As per YOLO's internal optimization logic; Manual tuning led to performance drops and longer training time

This configuration was selected not only for its strong detection metrics but also for its scalability and practicality in real-world deployment scenarios. By focusing on the optimizer, which governs the learning dynamics across the entire model, this project achieved a more robust, stable, and efficient training pipeline.

4.6 Testing and Inference

To test the model output, the following code snippet has been used to predict using the model on some test images.

```
!yolo task=detect mode=predict
model={HOME}/runs/detect/train/weights/best.pt
conf=0.25 source="{dataset.location}/test/images"
save=True
```

4.7 Performance Evaluation

To assess the effectiveness of the proposed approach, both the baseline model and the multi-model system were evaluated on a standardized unseen test set. Three key performance metrics were used: Precision, Recall, and mean Average Precision (mAP@50), which collectively reflect the model's ability to accurately detect, localize, and classify diseases in bovine images.

The baseline model was trained on a combined dataset containing all disease classes, while the multi-model system consists of three models: a body-part classifier, a head disease detector, and a body disease detector. The performance of each individual model was recorded, and a weighted average based on the number of samples in each model's dataset was used to

derive a comparable metric against the baseline. The weighted average of the three performance metrics employs the following formula:

Weighted Average Precision

$$= \frac{(P_{body_part} \times N_{body_part}) + (P_{head} \times N_{head}) + (P_{body} \times N_{body})}{N_{body_part} + N_{head} + N_{body}}$$

Where:

P = Precision

N = Number of samples

This formula to calculate the weighted average for precision were also used to calculate recall, and mAP@50.

Table 4.5: *Performance summary on the trained models*

Dataset Type	Dataset Name	Precision	Recall	mAP@50
Combined (baseline)	Bovine Disease Detection	86.90%	83.30%	89.30%
Separate (multi-model approach)	Bovine Body Part	95.40%	91.20%	94.80%
	Bovine Head Disease Detection	88.90%	79.80%	85.50%
	Bovine Body Disease Detection	85.50%	84.20%	88.80%
	Weighted Average	89.86%	86.59%	90.90%

As shown in the table above, the multi-model approach yielded a higher overall weighted performance across all three metrics, with notable improvements in precision and mAP@50. The body part detection model, in particular, achieved excellent performance due to the distinct visual features between the head and the body of the animal. These conditions are generally well-represented in the dataset and present clear, distinguishable patterns that facilitate accurate detection.

However, the head disease detection model, while still effective, recorded slightly lower recall and mAP values. This could be attributed to the relatively subtle nature of some head-based disease symptoms, such as those observed in IBK, which often require close inspection or appear less pronounced in early stages. Additionally, certain classes like FMD and healthy mouth had fewer training samples, which may have limited the model's ability to generalize across variations.

The baseline model, while showing strong performance, reveals the strong ability of YOLOv11 to perform well, even with considerably high degree of dataset imbalance. However, it is inherently tasked with learning a broader set of features from a more diverse dataset. This can dilute the model's focus on finer disease-specific patterns, potentially leading to lower performance compared to a specialized multi-model configuration.

By training dedicated models for specific anatomical regions, the multi-model system demonstrates its capacity to better learn contextual features and deliver enhanced classification outcomes. These results affirm the potential of targeted model specialization in improving disease detection accuracy.

In conclusion, the multi-model approach exhibits a meaningful performance gain over the baseline model and offers a promising direction for further refinement and practical deployment, especially in real-world scenarios where precision and reliability are essential.

CHAPTER 5: CONCLUSION AND FUTURE WORK

5.1 Overview

This final chapter summarizes the major achievements of the project in relation to the original objectives, highlights the key findings, discusses the encountered limitations, and provides recommendations for future work to further improve the current system.

5.2 Achievements

The goals of this research were guided by three main objectives mentioned in Chapter 1. Table 5.1 summarizes each objective alongside its corresponding achievements.

Table 5.1: *Achievements summary*

Objectives	Achievements
To investigate the process of detecting and classifying bovine diseases based on visible symptoms.	Successfully investigated a multi-model deep learning pipeline that includes region-of-interest (ROI) detection (body and head) followed by localized disease classification. Successfully investigated and developed a DL model to detect bovine diseases, using a single model approach, and multi-model approach for better localization and dataset imbalance mitigation.
To develop a custom-trained deep learning object detection model for bovine disease detection and classification.	A YOLOv11-based model was trained on a curated dataset consisting of over 8,000 annotated images. Multiple models were developed: a baseline model, body part detection model, head disease detection model and body disease detection model.
To evaluate the performance of the proposed model.	Evaluation using metrics such as precision, recall, mAP@50, and confusion matrix. The performance of all models was compared, and the multi-model approach showed better results, especially in managing data imbalance and ROI-specific features.

5.3 Limitations

While the project achieved its goals, several limitations were encountered during the development and evaluation phases. These limitations are summarized in Table 5.2.

Table 5.2: *Limitations of this project*

Limitations	Description
Dataset-related	The dataset had significant class imbalance, particularly for underrepresented classes such as IBK and FMD.
	The available images varied in quality and resolution, which may have affected the model's learning capacity.
Model-related	Certain diseases, such as Ringworm, can appear on both the head and body. This created ambiguity in the ROI classification and affected the disease-specific detection accuracy.
	Although the multi-model approach improved performance in some cases, it introduced additional complexity and longer inference time due to the sequential nature of multiple detection stages.
Limited Number of Training Runs	While multiple training runs (e.g., 20 iterations) are typically conducted to obtain a more reliable average performance and account for variance in model initialization and data shuffling, this project was constrained by computational resources and time, preventing full execution of this approach.

5.4 Future Works

To further augment the current system's capabilities and address identified limitations, several promising avenues for future development are envisioned. Among these is the expansion and diversification of the dataset, recognizing its foundational role in deep learning model performance. This entails collecting a greater volume of images, with a particular emphasis on augmenting data for underrepresented disease classes, thereby fostering enhanced model generalization. Complementary to this, we propose experimenting with synthetic data generation techniques, such as Generative Adversarial Networks (GANs). This approach holds

the potential to significantly improve model robustness by simulating rare disease manifestations and effectively balancing class distributions.

Additionally, future iterations of this project should incorporate multiple training runs, ideally 20 or more, to account for randomness introduced during model initialization and data shuffling. Averaging the results from these runs would provide a more statistically robust evaluation of model performance, offering deeper insights into the reliability and consistency of the results.

Beyond model enhancement, another critical area for future development lies in improving the project's practical usability through model deployment. This facilitates integration of the trained models into a mobile-friendly application or a web interface, enabling offline disease detection. This would empower farmers in remote locations with immediate diagnostic capabilities. Furthermore, considering the model's development, there is significant potential for larger-scale adaptation through edge deployment. This includes establishing a real-time monitoring system for practical, real-world scenarios such as farms and ranches, leveraging devices like the Raspberry Pi. Such an adaptation would streamline the automated disease detection process, ultimately amplifying the model's effectiveness and the overall impact of this project.

5.5 Conclusion

This study successfully developed and evaluated a deep learning-based object detection model to identify and classify external bovine diseases using visible symptoms. By adopting a multi-model approach and applying strategies to mitigate dataset imbalance, the project demonstrates a practical solution with promising results for disease identification tasks in the livestock industry. With further improvements in dataset quality, model design, and deployment strategies, this work lays the foundation for future scalable applications of AI in veterinary diagnostics and agricultural technology.

REFERENCES

- Bello, R.-W., Sufril Azlan Mohamed, A., & Zawawi Talib, A. (2021). Enhanced Mask R-CNN for herd segmentation. *International Journal of Agricultural and Biological Engineering*, 14(3), 238–244. <https://doi.org/10.25165/j.ijabe.20211404.6398>
- Fuentes, S., Gonzalez Viejo, C., Tongson, E., & Dunshea, F. R. (2022). The livestock farming digital transformation: implementation of new and emerging technologies using artificial intelligence. In *Animal Health Research Reviews* (Vol. 23, Issue 1). <https://doi.org/10.1017/S1466252321000177>
- Google. (n.d.). *Google Colaboratory*. Retrieved June 19, 2025, from <https://colab.google/>
- Gupta, S., Kuehn, L. A., & Clawson, M. L. (2023). Early detection of infectious bovine keratoconjunctivitis with artificial intelligence. *Veterinary Research*, 54(1), 122. <https://doi.org/10.1186/s13567-023-01255-w>
- Huigol, P. (2024, November 28). Top 4 pre-trained models for image classification with Python code. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2020/08/top-4-pre-trained-models-for-image-classification-with-python-code/>
- Hurtado, J. B., & Giraldo-Ríos, C. (2019). Economic and Health Impact of the Ticks in Production Animals. In *Ticks and Tick-Borne Pathogens*. <https://doi.org/10.5772/intechopen.81167>
- Kaggle. (n.d.). *Kaggle: Your Machine Learning and Data Science Community*. <https://www.kaggle.com>
- Kappes, A., Tozoneyi, T., Shakil, G., Railey, A. F., McIntyre, K. M., Mayberry, D. E., Rushton, J., Pendell, D. L., & Marsh, T. L. (2023). Livestock health and disease economics: a scoping review of selected literature. In *Frontiers in Veterinary Science* (Vol. 10). <https://doi.org/10.3389/fvets.2023.1168649>
- Lake, B., Getahun, F., & Teshome, F. T. (2022). Application of Artificial Intelligence Algorithm in Image Processing for Cattle Disease Diagnosis. *Journal of Intelligent Learning Systems and Applications*, 14(04), 71–88. <https://doi.org/10.4236/jilsa.2022.144006>

- Mehta, S., & Khurana, S. (2024). Integrating Convolutional Neural Networks and Support Vector Machines for Cattle Disease Classification. *2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT)*, 1–5. <https://doi.org/10.1109/ICEEICT61591.2024.10718546>
- Mekonnen, S. A., Gezehagn, A., Berju, A., Haile, B., Dejene, H., Nigatu, S., Molla, W., & Jemberu, W. T. (2021). Health and economic burden of foodborne zoonotic diseases in Amhara region, Ethiopia. *PLOS ONE*, *16*(12), e0262032. <https://doi.org/10.1371/journal.pone.0262032>
- Mia, M. R., Hossain, S. A., Chhoton, A. C., & Chakraborty, N. R. (2018). A comprehensive study of data mining techniques in health-care, medical, and bioinformatics. In *International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2)*. IEEE. <https://doi.org/10.1109/IC4ME2.2018.8465626>
- Naikar, M. L., Nandeppanavar, A. S., & Kudari, M. (2023). Lumpy Skin Disease Detection Using GUI Based Deep Learning Model in Cattle. *2023 Third International Conference on Ubiquitous Computing and Intelligent Information Systems (ICUIS)*, 182–187. <https://doi.org/10.1109/ICUIS60567.2023.00039>
- Neethirajan, S. (2023). Artificial Intelligence and Sensor Innovations: Enhancing Livestock Welfare with a Human-Centric Approach. *Human-Centric Intelligent Systems*, *4*(1), 77–92. <https://doi.org/10.1007/s44230-023-00050-2>
- Nithyashree. (2024, December 5). Image classification using machine learning. *Analytics Vidhya*. <https://www.analyticsvidhya.com/blog/2022/01/image-classification-using-machine-learning/>
- Roboflow. (n.d.). *Roboflow: Machine Learning Made Easy*. <https://www.roboflow.com/>
- Rony, M., Barai, D., Riad, & Hasan, Z. (2021). Cattle external disease classification using deep learning techniques. In *2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT)* (pp. 1–7). IEEE. <https://doi.org/10.1109/ICCCNT51525.2021.9579662>
- Sapkota, R., & Karkee, M. (2024). Comparing YOLO11 and YOLOv8 for instance segmentation of occluded and non-occluded immature green fruits in complex orchard environment. <https://doi.org/10.48550/arXiv.2410.19869>

- Shahbaz, A., Zhang, W., & Smith, M. (2024). A two-stage approach using YOLO for automated assessment of digital dermatitis within Dairy Cattle. *2024 IEEE 22nd World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, 000417–000422. <https://doi.org/10.1109/SAMI60510.2024.10432745>
- Shinde, S., Himpalnerkar, A., Shendurkar, S., Deshmane, S., & Jadhav, S. (2024). Cattle Disease Detection using VGG16 CNN Architecture. *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 1–6. <https://doi.org/10.1109/ICCCNT61001.2024.10724717>
- Shoaib, M., Shah, B., El-Sappagh, S., Ali, A., Ullah, A., Alenezi, F., Gechev, T., Hussain, T., & Ali, F. (2023). An advanced deep learning models-based plant disease detection: A review of recent research. *Frontiers in Plant Science*, 14. <https://doi.org/10.3389/fpls.2023.1158933>
- Ultralytics. (2024, September 30). *Ultralytics YOLO11*. <https://docs.ultralytics.com/>
- Yang, Q., Zhang, H., Xia, J., & Zhang, X. (2021). Evaluation of magnetic resonance image segmentation in brain low-grade gliomas using support vector machine and convolutional neural network. *Quantitative Imaging in Medicine and Surgery*, 11(1), 300–316. <https://doi.org/10.21037/qims-20-783>

APPENDIX A

Prediction Example on Validation Set



Figure A.1: Prediction on Body Disease Detection

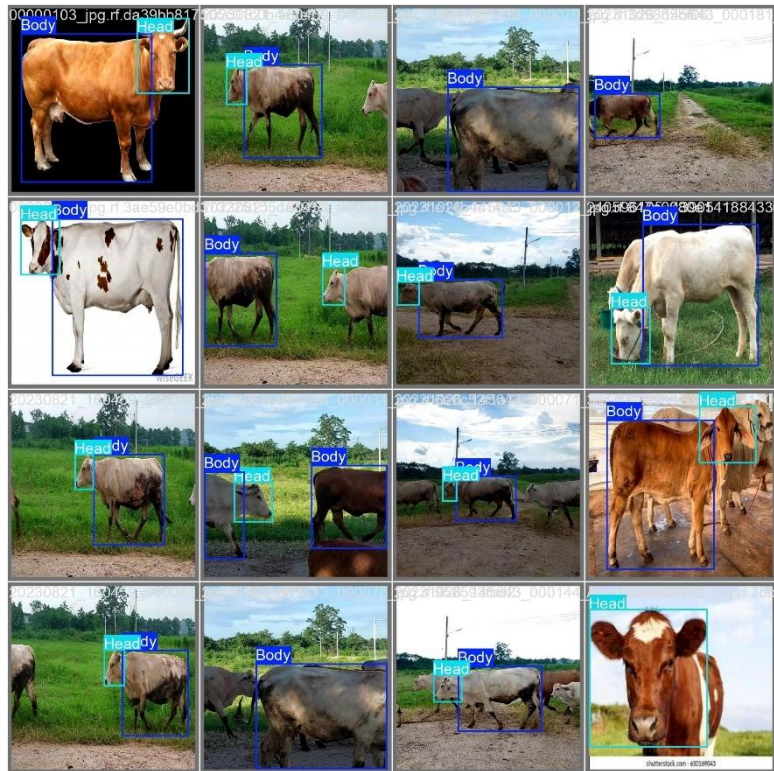


Figure A.2: Prediction on Body Part Detection

APPENDIX B

Github RepositoryLink

The Nootbook that was used in Google Colab to develop the model and the dataset link to the dataset including steps to recreate the project can be accessed through the following Github repository:

<https://github.com/faariS-yntax/Detection-and-Classification-of-External-Bovine-Diseases-Using-Deep-Learning>