### MAJOR PROJECT REPORT

On

### *“*Poultry Disease Diagnosis with a Vision: Leveraging Deep Learning for Automated Image Analysis*”*

*Submitted in partial fulfillment of the requirements for the award of*

### Bachelor of Technology (B.Tech)

In the department of

### Computer Science & Engineering

### 

*Submitted by*:

### Anushka Khatua (UG/02/BTCSE/2020/025)

### Surya Chakraborty(UG/02/BTCSEAIML/2020/001)

### Biswajit Chakraborty (UG/02/BTCSEAIML/2020/003)

*Under the Guidance of*

### Dr. Jhilam Mukherjee

### Assistant Professor

### Dept. Of CSE

### Adamas University

### School of Engineering & Technology

**ADAMAS University, Kolkata, West Bengal**

**Jan 2024 – June 2024**

**CERTIFICATE**

This is to certify that the project report entitled ***“*Poultry Disease Diagnosis with a Vision: Leveraging Deep Learning for Automated Image Analysis*”****,* submitted to the School of Engineering & Technology (SOET), **ADAMAS UNIVERSITY, KOLKATA** in partial fulfillment for the completion of **Semester – 8th**of the degree of **Bachelor of Technology** in the department of **Computer Science & Engineering**, is a record of bonafide work carried out by **Anushka Khatua**, **UG/02/BTCSE/2020/025, Surya Chakraborty, UG/02/BTCSEAIML/2020/001, Biswajit Chakraborty UG/02/BTCSEAIML/2020/025**  under our guidance.

All help received by us from various sources has been duly acknowledged.

No part of this report has been submitted elsewhere for award of any other degree.

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Jhilam Mukherjee**

**(Project Guide)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Mr. Sayantan Singha Roy/ Mr. Aninda Kundu**

**(Project Coordinator)**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Dr. Sajal Saha**

**(HOD CSE)**

### ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be incomplete without the mentioning of the people whose constant guidance and encouragement made it possible. We take pleasure in presenting before you, my project, which is the result of a studied blend of both research and knowledge.

We express our earnest gratitude to our **Dr. Jhilam Mukherjee (Project Guide)**, **Department of CSE**, for their constant support, encouragement and guidance. We are grateful for their cooperation and valuable suggestions.

Finally, we express our gratitude to all other members who are involved either directly or indirectly for the completion of this project.

## DECLARATION

We, the undersigned, declare that the project entitled ‘**Poultry Disease Diagnosis with a Vision: Leveraging Deep Learning for Automated Image Analysis**’, being submitted in partial fulfillment for the award of Bachelor of Engineering Degree in Computer Science & Engineering, affiliated to ADAMAS University, is the work carried out by us.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Anushka Khatua**

**(UG/02/BTCSE/2020/025)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Surya Chakraborty**

**(UG/02/BTCSEAIML/2020/001)**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Biswajit Chakraborty**

**(UG/02/BTCSEAIML/2020/003)**

**ABSTRACT**

Poultry diseases pose a significant threat to the poultry industry, causing substantial economic losses and compromising public health. Accurately predicting and diagnosing poultry diseases is crucial for implementing preventive measures and ensuring the production of safe and healthy poultry products. This paper proposes a poultry disease prediction system utilizing deep learning techniques to analyze poultry images and clinical data for disease detection and classification.

The core of the system will be a pre-trained RetinaNet model. RetinaNet is a convolutional neural network (CNN) architecture specifically designed for object detection tasks. It excels at pinpointing objects of interest within images and assigning class labels to them. In our case, the RetinaNet model will be fine-tuned to identify and classify various poultry diseases based on images captured from poultry farms.

To supplement image-based analysis, the system integrates clinical data, including poultry health records, laboratory test results, and necropsy reports. This clinical data provides valuable context and enhances the system's understanding of disease patterns and their correlation with visual features.

The proposed system employs a data augmentation strategy to artificially expand the dataset and improve the neural network's generalizability. Data augmentation techniques, such as image flipping, rotation, cropping, and noise addition, create new variations of existing images, making the neural network more robust to real-world image variations.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | **TITLE** | **PAGE** |
|  | **TITLE PAGE** | **i** |
|  | **CERTIFICATE** | **ii** |
|  | **ACKNOWLEDGEMENT** | **iii** |
|  | **DECLARATION** | **iv** |
|  | **ABSTRACT** | **v** |
|  | **TABLE OF CONTENTS** | **vi** |
|  | **LIST OF FIGURES** | **viii** |
|  | **LIST OF TABLES** | **x** |
|  | **ACRONYMS** | **xi** |
| **1** | **INTRODUCTION** |  |
|  | 1.1 Background | **1** |
|  | 1.2 Purpose of the project | **2** |
|  | 1.3 Problem Statement | **2** |
|  | 1.4 Objective | **3** |
| **2** | **BACKGROUND** |  |
|  | 2.1 Common Poultry Diseases | **4-5** |
|  | 2.2 Economic Impact of Poultry Diseases | **5** |
|  | 2.3 Impact on Human Health | **5** |
|  | 2.4 Poultry Diseases Detection Methods | **5** |
|  | 2.5 Deep Learning for Poultry Disease Detection | **5** |
|  | 2.6 Previous Works on DL for Poultry Diseases Detection | **6** |
|  | 2.7 Challenges & Future Direction | **6** |
| **3** | **METHODOLOGY** |  |
|  | **3.1** Convolutional Neural Network | **7-10** |
|  | **3.2 ResNet & Retina-Net** | **10-16** |
|  | 3.3 Dropout | **18-19** |
|  | 3.4 YOLOv8 | **19-21** |
|  | 3.5 Grad-CAM | **21-22** |
| **4** | **TECHNOLOGY** |  |
|  | 4.1 Tensor Records | **23-29** |
| **5** | **RESULTS AND DISCUSSION** | **30-37** |
| **6** | **CONCLUSION** | **36** |
| **7** | **FUTURE WORK** | **37** |
| **8** | **REFERENCE** | **38-41** |
| **9** | **PLAGARISM REPORT** | **42** |

**LIST OF FIGURES**

|  |  |  |
| --- | --- | --- |
| **FIGURE** | **TITLE** | **PAGE** |
| **Fig 3.1** | Layers used to build Convolutional Neural Network | **7** |
| **Fig 3.2** | Input to the convolution layer and convolution filter | **8** |
| **Fig 3.3** | Convolution operations and output | **9** |
| **Fig 3.4** | Max pooling operation | **10** |
| **Fig 3.5** | Resnet basic building block | **11** |
| **Fig 3.6** | RetinaNet Architecture | **12** |
| **Fig 3.7** | RetinaNet workflow | **14** |
| **Fig 3.8** | Dropout visualization | **19** |
| **Fig 3.9** | YOLOv8 Architecture | **21** |
| **Fig 3.10** | Grad-CAM Architecture | **22** |
| **Fig 4.1** | The sizes of images in the dataset changed to 224\*224 and 512\*512 | **25** |
| **Fig 5.1** | Training and validation plots for baseline model | **31** |
| **Fig 5.2** | Training and validation plots for the Resnet 50 model | **32** |
| **Fig 5.3** | After successfully predicting the diseases Cocci | **32** |
| **Fig 5.4** | After successfully predicting the diseases New Castle | **33** |
| **Fig 5.5** | After successfully predicting the diseases Salmo | **33** |
| **Fig 5.6** | Epoch vs Accuracy graph of Retina-net | **33** |
| **Fig 5.7** | Epoch vs Accuracy graph of YOLOv8 | **34** |
| **Fig 5.8** | Grad-CAM model of YOLOv8 | **34** |
| **Fig 5.9** | Grad-CAM model of Retina-Net | **35** |

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE** | **TITLE** | **PAGE** |
| **Table-3.1** | Activation function | **17** |
| **Table-4.1** | Dataset splitting for training baseline models | **26** |
| **Table-4.2** | Hyperparameter using Rest-Net | **26** |
| **Table-4.3** | Dataset splitting training, validation and test sets | **27** |
| **Table-4.4** | Hyperparameters for training the Convolutional Neural Network model | **28** |
| **Table-5.1** | Dataset | **30** |
| **Table-5.2** | Training and validation results for baseline model | **31** |

**ACRONYMS**

|  |  |
| --- | --- |
| **Abbreviations** | **Descriptions** |
| ND | Newcastle Disease |
| AI | Avian Influenza |
| IBD | **Infectious Bursal Disease** |
| MD | **Marek's Disease** |
| RNN | Recurrent Neural Networks |
| CNN | Convolutional Neural Networks |
| FPN | Feature Pyramid Network |
| ReLU | Rectified Linear Unit |
| CLI | Command Line Interface |
| CAM | Class Activation Mapping |
| Grad-CAM | Gradient Weight Class Activation Mapping |
| GAP | Global Average Pooling |

**CHAPTER 1**

**INTRODUCTION**

In this chapter we are going to brief our main motive of the project what are the problem statement we had find out by doing survey of previous works and how we are going to find the solution of those problems and also our main motive of this work.

### 1.1 Background

Poultry farming is sub sector of farming it is one of the major practices that Indian farmers are having which involves keeping birds such as chickens for their meat and eggs, ducks, pigeons, and other animals like goat, sheep, etc. for their different uses. India is a country which is mainly depends for their farming habitat which is inherited generation wise as it is one of the largest producer for wheat, rice but apart from that it also 3rd highest producer of egg i.e. broiler egg and 5th largest producer of chicken meat which is increasing by 7.86% to 8% per year. This also helps the economic growth of the country by its poultry contribution.

However, farmers are facing many problems such as various diseases which became very challenging for their farming. Where chickens which is one of the major part of poultry farming are get effected by diseases like salmonella, Newcastle, pox and many more. It often causes both loss for the society as well as for the farmers as it create a loss of huge amount of loss of species which is directly affected the economy and also a huge effect on the farmer as all the production get stopped at a glance and on the society it effect to the health of human being also. In addition to diseases, farmers lack access to reliable information of necessary resources on poultry as only few numbers of extension officers are present on distant locations for consultations and which lead to the lack of awareness campaign which are occurred on animal husbandry and due to less awareness, they need to rely on word of mouth from friends and their ways and tradition as there are no other way exist.

Farmers need assistance and solution to this type of problems to stop the loss percentage so here deep learning module used to give the proper identification of the disease in advance and also the treatment procedure to so that we can counter the maximum number of diseases to give a hope of relief to them.

### 1.2 Purpose of the Project

Our main purpose of this project is to pre detect the diseases with the help of image detection technique by identifying their natural waste and also the condition of the feather and other vulnerability which is observed over different period of time manually but this project aims to check everyday help and using some advanced techniques to stop the loss of the poultry.

### 1.3 Problem Statement

Accurately detecting poultry in various environments using robust and efficient techniques is crucial for various applications, including poultry farming, disease monitoring, and food safety inspection. Traditional poultry detection methods, such as manual observation and video surveillance, are often time-consuming, labor-intensive, and prone to human error. Therefore, there is a need for automated poultry detection techniques that can overcome these limitations and provide accurate and reliable results.

**Key Challenges**

* **Varied Poultry Appearance:** Poultry can exhibit diverse appearances in terms of size, color, and posture, making it challenging for detection algorithms to generalize effectively.
* **Complex Backgrounds:** Poultry are often found in complex and cluttered environments, such as farms, barns, and processing facilities, which can obscure their presence and make detection more difficult.
* **Real-time Requirements:** In certain applications, such as automated sorting and inspection systems, real-time detection is essential for efficient operation.
* **Computational Efficiency:** Poultry detection techniques should be computationally efficient to minimize processing time and resource utilization.

**Potential Solutions**

* **Deep Learning-based Approaches:** Deep learning techniques, particularly convolutional neural networks (CNNs), have demonstrated promising results in object detection tasks. CNNs can effectively extract features from complex images and classify poultry with high accuracy.
* **Transfer Learning:** Transfer learning involves leveraging pre-trained CNN models, trained on large datasets of general images, to reduce the training time and improve the performance of poultry detection models.
* **Data Augmentation:** Data augmentation techniques, such as image flipping, rotation, and cropping, can artificially expand the training dataset, making the detection models more robust to variations in poultry appearance and background conditions.
* **Lightweight Models:** Developing lightweight CNN architectures specifically tailored for poultry detection can reduce computational requirements and enable real-time applications.

### 1.4 Objective

The development of accurate, efficient, and robust poultry detection techniques can significantly impact various industries and applications:

* **Poultry Farming:** Automated poultry detection can improve farm management practices by monitoring poultry health, tracking growth patterns, and detecting potential abnormalities early on.
* **Disease Monitoring:** Automated detection can facilitate early identification and isolation of diseased poultry, preventing the spread of diseases and minimizing economic losses.
* **Food Safety Inspection:** Automated detection can enhance food safety by identifying and removing contaminated or unhealthy poultry products from the supply chain.
* **Environmental Monitoring:** Automated detection can track poultry populations and movements, aiding in conservation efforts and environmental protection.

### CHAPTER 2

### BACKGROUND

In this chapter we are going to brief the work which has done till yet and what are the model they had used for their works.

The poultry industry is a vital sector of the global economy, providing a significant source of protein and income for millions of people worldwide. However, poultry production is susceptible to various diseases that can cause significant economic losses and pose a threat to human health. These diseases can affect poultry of all ages, from chicks to adult birds, and can lead to decreased productivity, increased mortality, and reduced quality of poultry products.

**2.1 Common Poultry Diseases**

Poultry diseases can be caused by various factors, including viruses, bacteria, fungi, and parasites. Some of the most common poultry diseases include:

* **Avian influenza (AI)**: AI is a highly contagious viral disease that can affect various bird species, including poultry. It can cause severe respiratory illness and death in poultry, and some strains of AI can also infect humans.
* **Newcastle disease (ND)**: ND is another highly contagious viral disease that affects poultry. It can cause respiratory problems, neurological signs, and death in infected birds.
* **Infectious bursal disease (IBD)**: IBD is a viral disease that affects the bursa of Fabricius, a vital organ for the development of the immune system in poultry. Infected birds experience immunosuppression, making them more susceptible to other diseases.
* **Marek's disease (MD)**: MD is a viral disease that causes tumors in poultry. It can affect different parts of the bird's body, including the nerves, viscera, and skin.
* **Coccidiosis**: Coccidiosis is a parasitic disease caused by coccidia, which are protozoan parasites that infect the intestinal tract of poultry. It can cause diarrhea, weight loss, and even death in infected birds.

**2.2 Economic Impact of Poultry Diseases**

Poultry diseases have a significant economic impact on the poultry industry. They can cause direct losses due to decreased productivity, increased mortality, and reduced quality of poultry products. Additionally, poultry diseases can lead to indirect losses due to trade restrictions, increased veterinary costs, and consumer concerns about food safety.

**2.3 Impact on Human Health**

Some poultry diseases, such as avian influenza and Newcastle disease, can also pose a threat to human health. These diseases can be transmitted to humans through direct contact with infected birds or through contaminated poultry products. While human cases of these diseases are relatively rare, they can have severe consequences, including respiratory illness, pneumonia, and even death.

**2.4 Poultry Disease Detection Methods**

Traditional methods of poultry disease detection often rely on clinical signs, laboratory tests, and post-mortem examinations. However, these methods can be time-consuming, expensive, and often impractical for large-scale poultry operations.

**2.5 Deep Learning for Poultry Disease Detection**

Deep learning, a subfield of artificial intelligence, has emerged as a promising tool for poultry disease detection. Deep learning algorithms can analyses large datasets of images and other data to identify patterns and make predictions. This makes them well-suited for detecting poultry diseases from images of poultry carcasses, eggs, or even live birds.

**2.6 Previous Works on Deep Learning for Poultry Disease Detection**

Several studies have demonstrated the potential of deep learning for poultry disease detection. For example, one study used a convolutional neural network (CNN) to classify images of poultry carcasses into three categories: healthy, diseased, and decomposing. The CNN achieved an accuracy of 98.9% in classifying healthy carcasses, 96.7% in classifying diseased carcasses, and 95.2% in classifying decomposing carcasses.

Another study used a deep learning model to detect Marek's disease in poultry from images of their eyes. The model achieved an accuracy of 91.3% in detecting Marek's disease lesions in the eyes of infected birds.

These studies demonstrate the potential of deep learning for poultry disease detection. Deep learning algorithms can be used to detect poultry diseases from images of poultry carcasses, eggs, or even live birds. This could lead to the development of more rapid, accurate, and cost-effective methods of poultry disease detection.

**2.7 Challenges and Future Directions**

Despite the promising results of previous studies, there are still challenges that need to be addressed for deep learning to be widely adopted in the poultry industry. One challenge is the need for large datasets of high-quality images of poultry carcasses, eggs, and live birds. These datasets are essential for training deep learning algorithms to accurately detect poultry diseases.

Another challenge is the need to develop deep learning algorithms that are robust to variations in lighting, image quality, and bird breed. These variations can affect the performance of deep learning algorithms and make it difficult to apply them to real-world poultry disease detection scenarios.

Despite these challenges, deep learning has the potential to revolutionize poultry disease detection. With further research and development, deep learning algorithms could be used to develop.

**CHAPTER 3**

**METHODOLOGY**

In this chapter the methodology of the project is suggested what are the process and what are the positive and negative side for deployment of each models and which model can provide better result among comparing with other models.

**3.1 Convolutional Neural Network**

Convolutional neural networks (CNNs) excel at tasks involving spatial data, like images. Unlike traditional neural networks, they exploit the inherent structure of the data by using filters called kernels. These kernels slide across the input, extracting features like edges and textures. This process, called convolution, progressively builds higher-level representations of the data, culminating in accurate classifications or predictions. CNNs' ability to learn these features directly from the data makes them powerful tools for various applications, from image recognition and object detection to medical image analysis and natural language processing.

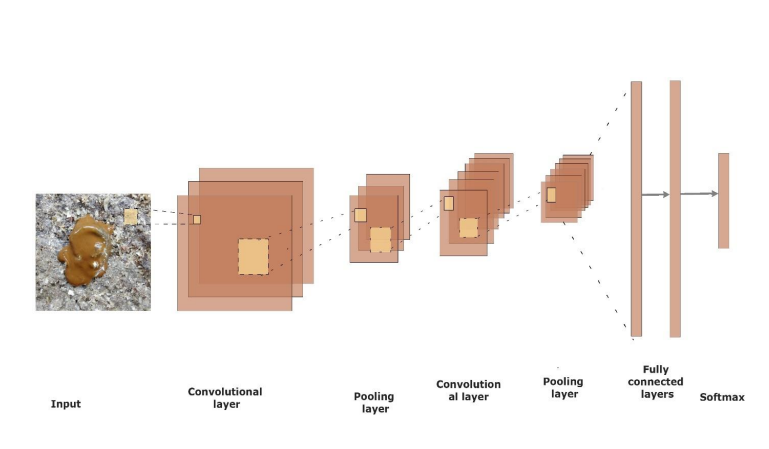


Fig 3.1 : Layers used to build Convolutional Neural Network

**3.1.1 Convolutional layers**

Convolutional layers are a key component of many deep learning architectures, particularly those used for image recognition and natural language processing. They work by applying a filter, or kernel, to an input image, scanning it across the image and computing the dot product with the local image patch at each position. This process is repeated for multiple filters, each extracting different features from the input. The resulting feature maps contain the extracted information, which can then be used by subsequent layers in the network to make predictions. Convolutional layers offer several advantages, including their ability to learn spatial relationships between pixels and their ability to be shared across different parts of the image, which reduces the number of parameters that need to be learned.

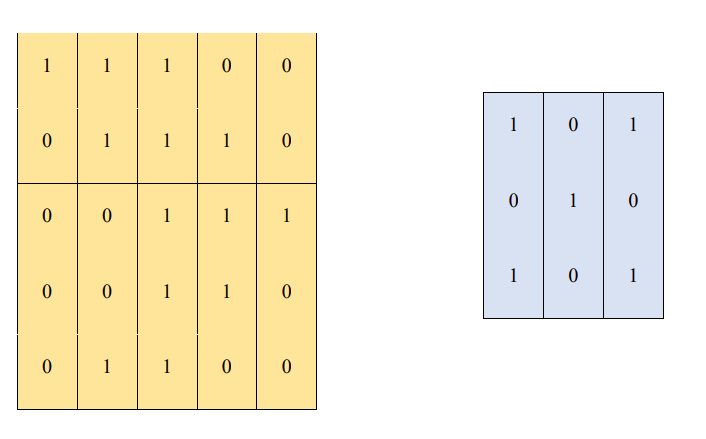


Fig 3.2: Input to the convolution layer and convolution filter

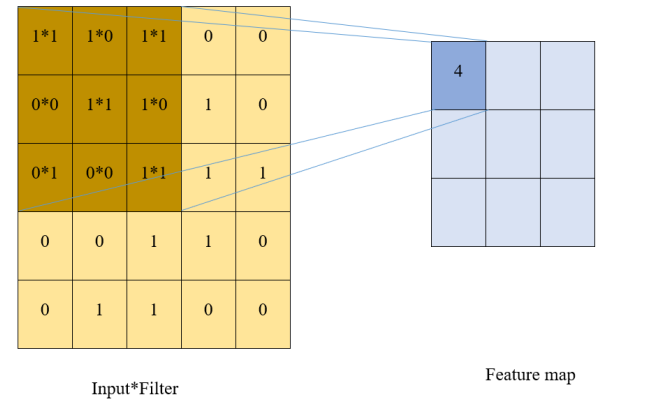


Fig 3.3: Convolution operations and output

**3.1.2 Pooling layers**

Pooling layers are like mini-summaries in a neural network's information processing journey. Imagine a large pool filled with water (data). Pooling layers act like nets that scoop out only the most relevant information (like the average or maximum water level). This helps reduce the network's complexity and computational cost while capturing key features. It's like summarizing a lengthy report into its main points, making it easier to analyze and understand. This allows the network to focus on important information while ignoring less relevant details, leading to better performance and efficiency.

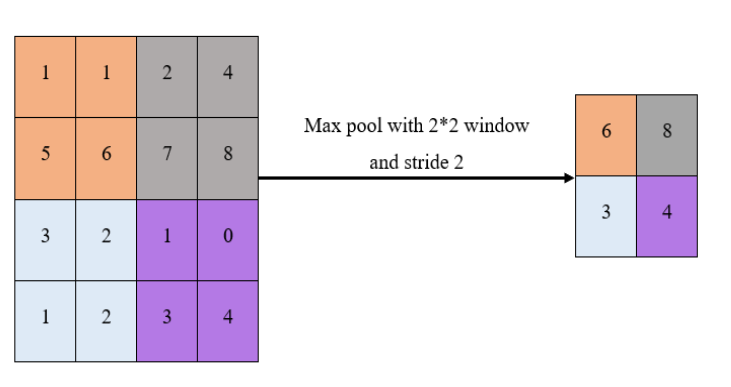


Fig 3.4: Max pooling operation

## 3.1.3 Fine-tuning the Pre-trained Networks

Fine-tuning pre-trained networks is a powerful technique in deep learning that leverages the knowledge gained from vast amounts of data to achieve better results on specific tasks. Imagine a chef with years of experience in various cuisines (pre-trained network). Now, they specialize in making delicious pizzas (fine-tuning) by adapting their skills and knowledge to the specific ingredients and techniques needed for pizza. This targeted approach enables them to quickly master pizza-making and consistently deliver exceptional results. Similarly, fine-tuning pre-trained networks allows us to adapt the general knowledge of the network to a particular domain, leading to improved performance and faster training times compared to training from scratch. It's like giving the network a head start on its journey to becoming an expert in a specific field.

**3.2 ResNet-50**

ResNet-50 is a convolutional neural network (CNN) architecture with 50 layers designed for image recognition. It builds upon the basic CNN structure by introducing residual connections that bypass some layers, allowing the network to learn easier features and alleviate the vanishing gradient problem. This results in superior performance compared to deeper, plain CNNs. ResNet-50 consists of 4 main stages with increasing filter sizes and decreasing feature map sizes. Each stage utilizes residual blocks containing multiple convolutional layers and activation functions, where residual connections merge outputs with earlier layers' outputs. This enables efficient training and feature extraction, making ResNet-50 a popular choice for various vision tasks.

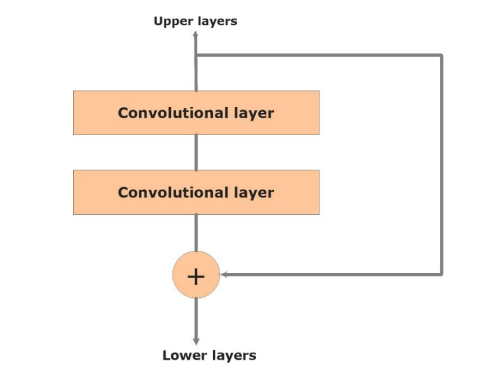


Fig 3.5: Resnet basic building block

**3.2.1 Retina-Net**

RetinaNet, a single-stage object detection framework, was introduced in 2017 and gained significant traction due to its speed and accuracy. It eliminates the need for separate region proposal and classification steps prevalent in traditional two-stage detectors, achieving remarkable real-time performance. The core of RetinaNet lies in its focal loss function, which effectively addresses the class imbalance issue common in object detection tasks. This function assigns higher weights to hard-to-classify examples, focusing the network's attention on those instances during training. Additionally, RetinaNet utilizes a Feature Pyramid Network (FPN) to extract features at multiple scales, enabling it to detect objects of diverse sizes efficiently. Furthermore, the network employs anchor boxes of various sizes and aspect ratios at each pyramid level, ensuring sufficient coverage for potential objects. By combining these innovative components, RetinaNet achieves state-of-the-art object detection performance while maintaining real-time inference speed, making it a highly attractive choice for various applications requiring fast and accurate object detection.

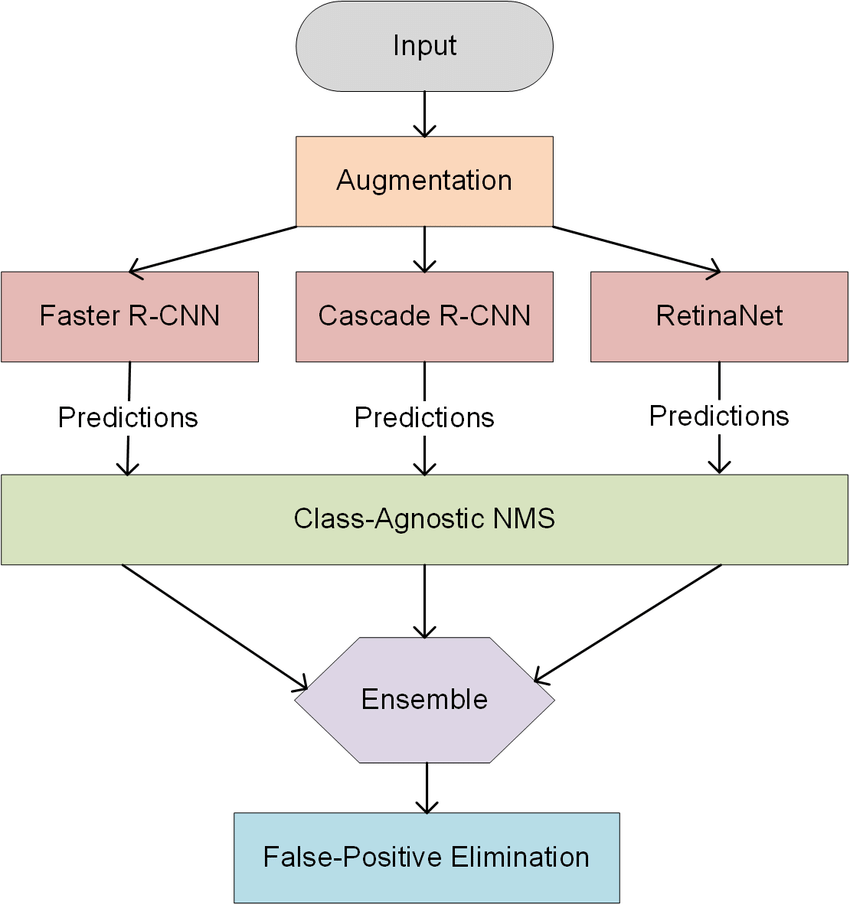


Fig 3.6: Retina-Net Architecture

## 3.2.2 Retina-Net Architecture and Framework

Retina-Net is a single-stage object detection framework that was introduced in 2017. It has gained significant traction due to its speed and accuracy. Unlike traditional two-stage detectors that require separate region proposal and classification steps, Retina-Net performs both tasks simultaneously, achieving remarkable real-time performance.

Here's an overview of the Retina-Net framework and architecture:

**Core Components:**

* **Focal Loss:** This is the key innovation in Retina-Net. It addresses the class imbalance issue common in object detection, where a large number of background examples can dominate the training process and hinder the network's ability to learn from foreground objects. The focal loss function assigns higher weights to hard-to-classify examples, focusing the network's attention on those instances during training. This results in improved accuracy for both positive and negative samples.
* **Feature Pyramid Network (FPN):** Retina-Net utilizes an FPN to extract features at multiple scales. This allows the network to detect objects of diverse sizes efficiently. The FPN combines high-level semantic features with low-level spatial features, creating a rich feature representation for object detection.
* **Anchor Boxes:** RetinaNet employs anchor boxes of various sizes and aspect ratios at each pyramid level. These boxes serve as potential object locations and guide the network to predict bounding boxes around objects. By using a diverse set of anchor boxes, Retina-Net ensures sufficient coverage for potential objects of different sizes and shapes.

**Architecture:**

* **Backbone Network:** The first stage of Retina-Net is a backbone network, typically a convolutional neural network (CNN) pre-trained on an image classification task. This network extracts features from the input image.
* **Feature Pyramid Network (FPN):** The extracted features are then passed through an FPN, which generates feature maps at multiple scales. These feature maps capture different levels of detail, enabling the network to detect objects of various sizes.
* **Task-Specific Networks:** Separate task-specific networks are applied to each FPN level. These networks predict bounding boxes and class labels for potential objects.
* **Focal Loss:** The predicted bounding boxes and class labels are then used to compute the focal loss. This loss function guides the network towards better predictions.

**Inference:**

* During inference, Retina-Net performs non-max suppression to remove redundant bounding box and generate the final set of detections.

**Benefits:**

* **Single-stage:** Retina-Net eliminates the need for separate region proposal and classification steps, making it significantly faster than two-stage detectors.
* **Focal Loss:** This function improves the network's ability to learn from hard-to-classify examples, resulting in better accuracy.
* **FPN:** The FPN allows Retina-Net to detect objects of diverse sizes efficiently.

**Applications:**

Retina-Net's speed and accuracy make it suitable for various applications, including:

* Real-time object detection for autonomous vehicles
* Object detection in surveillance systems
* Robotics and automation

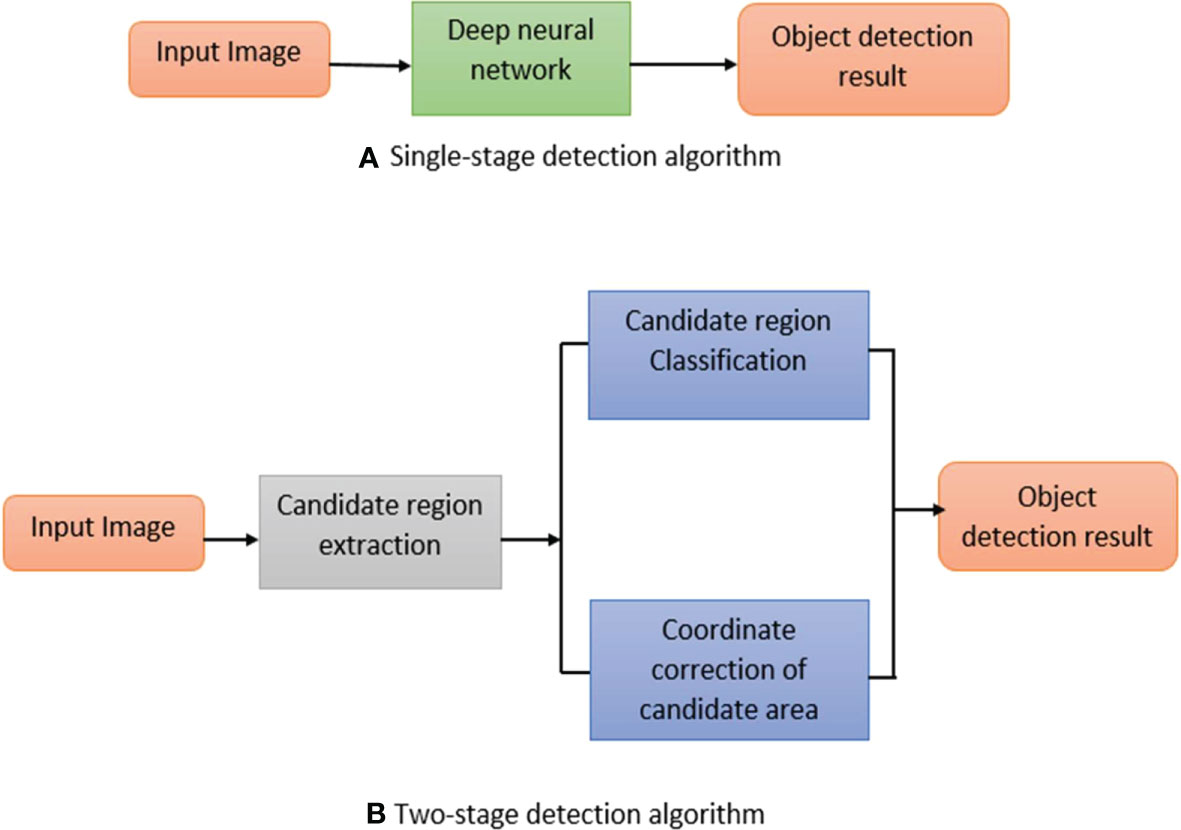


Fig 3.7: Retina-Net workflow

## Non-linearity in Deep Learning

Non-linearity is a crucial aspect of deep learning that allows it to model complex relationships between input and output data. It is achieved through various components:

* **Activation Functions:**

These functions introduce non-linear transformations in the neural network, allowing it to learn complex relationships beyond simple linear functions. Commonly used activation functions include:

* **Rectified Linear Unit (ReLU):** Outputs the input directly if it's positive, otherwise outputs 0. (Image of ReLU activation function)
* **Sigmoid:** Outputs a value between 0 and 1, representing the probability of the input belonging to a specific class. (Image of Sigmoid activation function)
* **Tanh:** Similar to sigmoid but outputs values between -1 and 1. (Image of Tanh activation function)
* **Non-linear Operations:**

Deep Learning algorithms often use non-linear operations like:

* **Matrix Multiplication:** This operation allows the network to combine features from different neurons in a non-linear manner.
* **Element-wise Operations:** Operations like addition, subtraction, and multiplication can be applied to individual elements of a tensor, introducing non-linearity.
* **Network Architecture:**

The architecture of a deep learning model itself can contribute to non-linearity. For example, convolutional layers can learn to extract non-linear features from the input data. Additionally, having multiple hidden layers allows the network to learn increasingly complex relationships between the input and output.

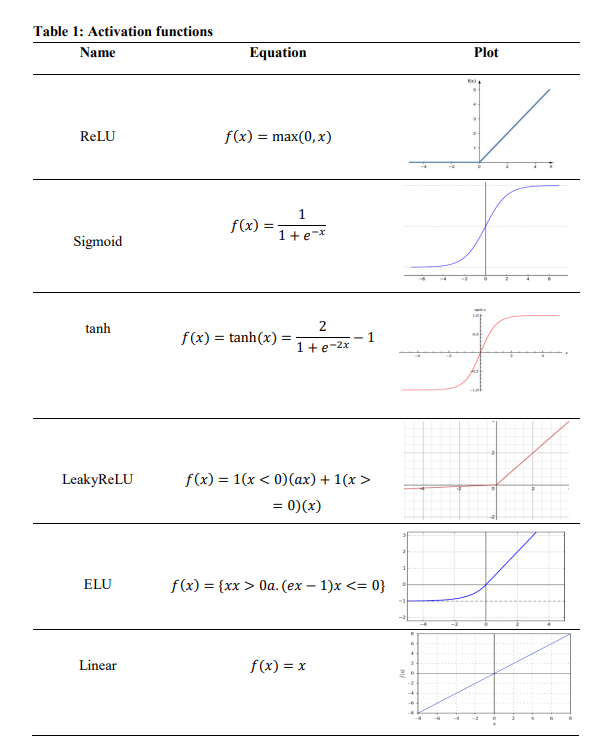
**Benefits of Non-linearity:**

* **Learns complex relationships:** Non-linearity allows deep learning models to learn complex relationships between input and output data, making them suitable for tasks like image recognition, natural language processing, and time series forecasting.
* **Increased flexibility:** With non-linearity, deep learning models can adapt to a wider variety of data and learn more intricate patterns.
* **Improved performance:** In many cases, using non-linear activation functions and operations leads to better performance compared to linear models.

**Examples of Non-linear Deep Learning Applications:**

* **Image Recognition:** Classifying images into different categories, detecting objects in images, and generating new images.
* **Natural Language Processing:** Understanding the meaning of text, translating languages, and generating text.
* **Time Series Forecasting:** Predicting future values of a time series based on past data.
* **Audio Processing:** Speech recognition, music generation, and audio classification.

In conclusion, non-linearity is a fundamental concept in deep learning that allows models to learn complex relationships and achieve superior performance in various applications. Understanding its role and implementation is crucial for effective deep learning development.

Table 3.1: Activation function

**3.3 Dropout**

Dropout is a regularization technique for neural networks that prevents overfitting by randomly dropping out neurons during training. This means that these neurons are not used for either forward or backward propagation during that training iteration. The remaining neurons are then scaled up by a factor of 1/(1-p), where p is the dropout probability. This has the effect of forcing the remaining neurons to learn more robust representations of the data, as they cannot rely on the output of any one neuron.

Dropout is a powerful regularization technique that can significantly improve the generalization performance of neural networks. It is a simple technique to implement and is very effective, especially for deep neural networks.

Dropout was first introduced by Srivastava et al. in 2014. They showed that dropout can significantly improve the performance of neural networks on a variety of tasks, including image recognition, natural language processing, and speech recognition. Dropout has since become a standard technique for regularizing neural networks and is widely used in deep learning.

There are a number of reasons why dropout is an effective regularization technique. First, dropout forces the network to learn more robust representations of the data, as it cannot rely on the output of any one neuron. Second, dropout prevents the network from memorizing the training data, as it does not always see the same neurons during training. Finally, dropout can help to reduce the number of parameters in the network, which can also help to prevent overfitting.

Dropout is a simple technique to implement and is very effective, especially for deep neural networks. It is a powerful regularization technique that can significantly improve the generalization performance of neural networks.

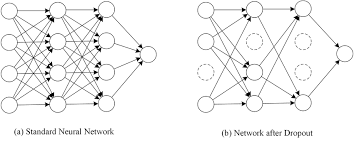


Fig 3.8: Dropout visualization

**3.4 YOLOv8**

YOLOv8 is a cutting-edge computer vision model developed by Ultralytics, known for the popular YOLOv5. It builds upon the success of previous versions, offering several improvements:

* Multitasking: YOLOv8 goes beyond just object detection. It tackles a range of vision tasks, including:
* Object detection: Identifying and locating objects in an image or video.
* Image classification: Categorizing the overall content of an image.
* Instance segmentation: Pinpointing individual objects and their precise outlines.
* State-of-the-Art Performance: YOLOv8 boasts high accuracy and speed. For instance, the YOLOv8 (medium) model achieves a map score of 50.2 at a fast 1.83 milliseconds on the COCO dataset with an A100 Tensors.
* User-Friendly Design: Like YOLOv5, it prioritizes ease of use with a Python package and a command-line interface (CLI) for straightforward implementation.
* Active Development: YOLOv8 is constantly evolving, with Ultralytics actively incorporating new features and addressing user feedback.

YOLOv8, like its predecessors, relies on a deep convolutional neural network (CNN) architecture to detect objects in images and videos. Here's a breakdown of its working architecture:

**Backbone:**

* This is the initial stage and uses a pre-trained CNN, such as a modified CSPDarknet53.
* The backbone extracts feature from the input image, including low-level details like edges and high-level features like shapes and objects.

**Neck:**

* This stage refines the information extracted by the backbone.
* It merges feature maps from different levels of the backbone using techniques like the Feature Pyramid Network (FPN).
* This creates a richer set of features for object detection at various scales.

**Head:**

* The final stage, the head, uses the processed features to make predictions.
* It predicts bounding boxes for objects in the image and assigns class probabilities to each bounding box.
* YOLOv8 employs a single-stage detection approach, meaning it predicts everything in one go, making it fast.

Here are some key improvements introduced in YOLOv8:

* New Backbone Architecture (CSPNet): This is more efficient and accurate than previous backbones.
* New Head Architecture (PANet): This is more robust to occlusions (where part of an object is hidden) and variations in object scale.
* Advanced Data Augmentation: Techniques like Mix-up and Cut Mix improve the model's ability to handle unseen data.
* Customizable Architecture: YOLOv8 allows for easier modification of the model structure for specific needs.

Overall, YOLOv8's architecture is designed for speed and accuracy in object detection tasks. It builds upon the strengths of previous YOLO versions while introducing advancements for better performance.

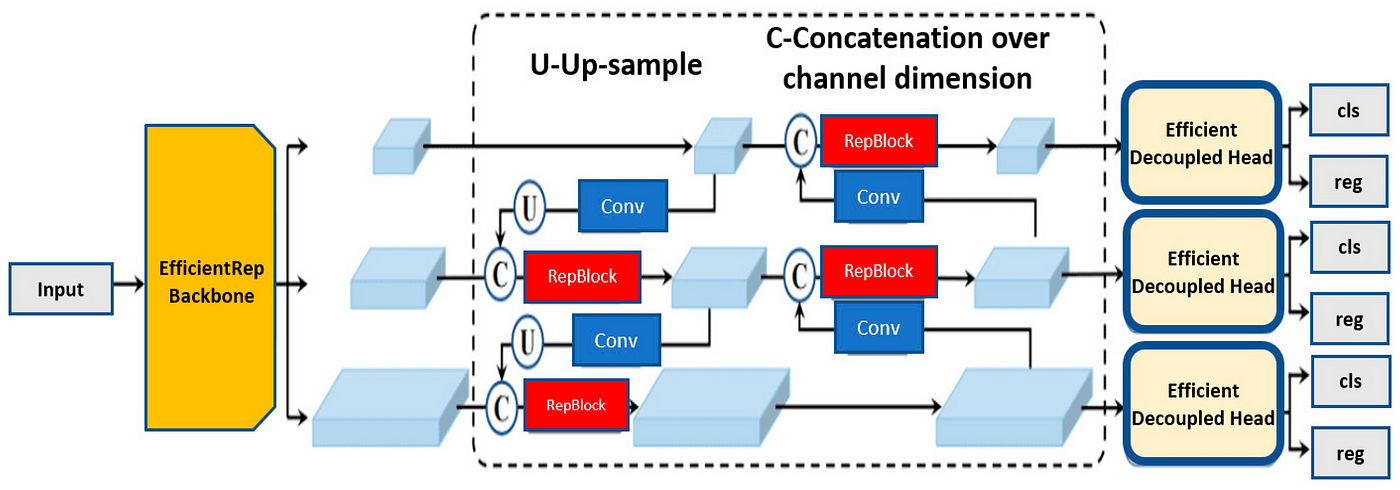


Fig 3.9: YOLOv8 Architecture

**3.5 Gradient Weight Class Activation Mapping (Grad-CAM)**

Grad-CAM is a technique in deep learning, particularly computer vision, that aids in understanding how CNNs make predictions for images. It addresses the inherent "black box" nature of deep models by visualizing the image regions most influential in the network's classification decision for a specific class.

* **Visualization**: Grad-CAM generates a heatmap superimposed on the input image, indicating the importance of different regions for the predicted class. Areas with higher intensity contribute more significantly to the prediction.
* **Class-Specific:** Grad-CAM tailors the heatmap to a particular class, allowing you to pinpoint the image parts most crucial for identifying that class.
* **Versatility:** Unlike Class Activation Mapping (CAM), which relies on the final fully connected layer (often absent in modern CNNs), Grad-CAM works with any CNN architecture by using gradients from the final convolutional layer.
* **Pre-trained CNN**: You start with a CNN pre-trained on an image classification task (e.g., ImageNet). This pre-trained model extracts features from images.
* **Class of Interest**: Specify the class you want to understand (e.g., "dog" in a dog vs. cat classification).
* **Gradient Calculation:** Backpropagate through the network to compute the gradients of the class score (output) with respect to the activations in the final convolutional layer.
* **Global Average Pooling (GAP):** Apply GAP to the gradients to create a weight vector highlighting the importance of each feature map channel for the chosen class.
* **Weighted Combination:** Multiply the weight vector element-wise with the final convolutional layer's activations, resulting in a class activation map (CAM).
* **Heatmap Generation:** Upscale the CAM to match the input image's spatial dimensions. Apply a ReLU activation (zeroing out negative values) to emphasize the positive weights and create a final heatmap.

**Architecture of Grad-CAM**

* **Interpretability:** Grad-CAM provides valuable insights into CNN decision-making, helping to debug models, identify biases, and build trust in their predictions.
* **Class-Specific Analysis:** Tailored heatmaps let you focus on specific classes, aiding in understanding how the network distinguishes between them.
* **Grad-CAM on Final Layer:** Grad-CAM focuses on the final convolutional layer of the pre-trained CNN. It analyses the gradients flowing into this layer to understand the model's focus for a specific class.
* **Class Activation Map (CAM):** Using the gradients, Grad-CAM calculates weights for each feature map in the final convolutional layer. These weights are then used to create a weighted sum of the feature maps, resulting in a Class Activation Map (CAM).
* **Heatmap Visualization:** Finally, the CAM is unsampled to match the size of the original image. This unsampled CAM is visualized as a heatmap, where brighter areas represent the image regions most critical for the CNN's classification.

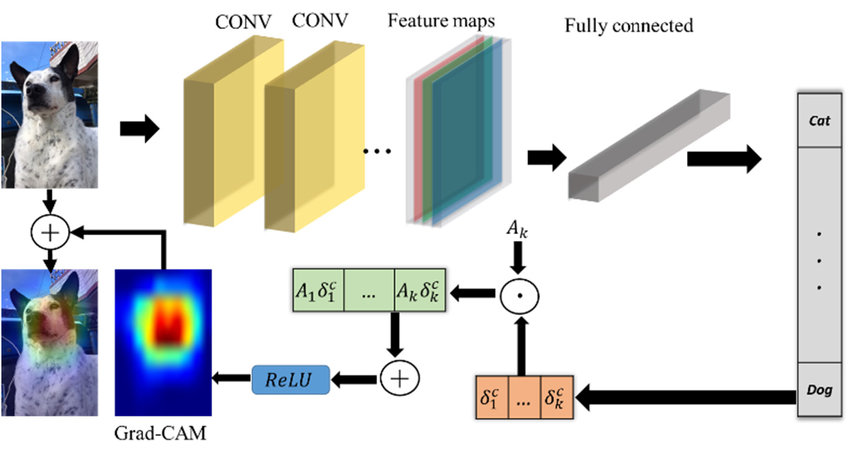


Fig 3.10: Grad-CAM Architecture

**CHAPTER 4**

**TECHNOLOGY**

In this chapter technology where all the details of the process of the model are explained in details, how to collect data and all the other necessary steps required to find the results.

**4.1 Tensor Records**

Tensor Records are a data format developed by Google for efficient storage and processing of structured data in deep learning applications. They are composed of key-value pairs, where the keys are strings and the values are Tensors. Tensors are multidimensional arrays of data, and they are the primary data type used in deep learning.

Tensor Records offer several advantages for storing and processing data in deep learning applications:

* Efficient storage: Tensor Records are stored in a columnar format, which makes them efficient for reading and writing.
* Flexible data representation: Tensor Records can store a wide variety of data types, including scalars, vectors, matrices, and strings.
* Integration with TensorFlow: Tensor Records are integrated with TensorFlow, which makes them easy to use in deep learning applications.

Tensor Records for Poultry Diseases Detection

Tensor Records can be used to store and process data for poultry diseases detection in a variety of ways. For example, they can be used to store:

* Poultry images: The pixels in a poultry image can be stored as a Tensor of floats.
* Clinical data: Clinical data, such as the age of a poultry bird and its symptoms, can be stored as a Tensor of strings or integers.
* Expert annotations: Expert annotations, such as the type of disease a poultry bird has, can be stored as a Tensor of strings.

Using Tensor Records for poultry diseases detection can offer several advantages:

* Improved data organization: Tensor Records can help to organize data in a way that is more efficient for deep learning applications.
* Reduced data pre-processing: Tensor Records can reduce the amount of data pre-processing required for deep learning applications.
* Improved model performance: Using Tensor Records can improve the performance of deep learning models for poultry diseases detection.

Generating Tensor Records for Poultry Diseases Detection

There are a variety of ways to generate Tensor Records for poultry diseases detection. For example, you can use:

* Custom code: You can write custom code to generate Tensor Records from raw data.
* Data pre-processing libraries: There are a number of data pre-processing libraries that can generate Tensor Records from raw data.
* Cloud-based data processing services: There are a number of cloud-based data processing services that can generate Tensor Records from raw data.

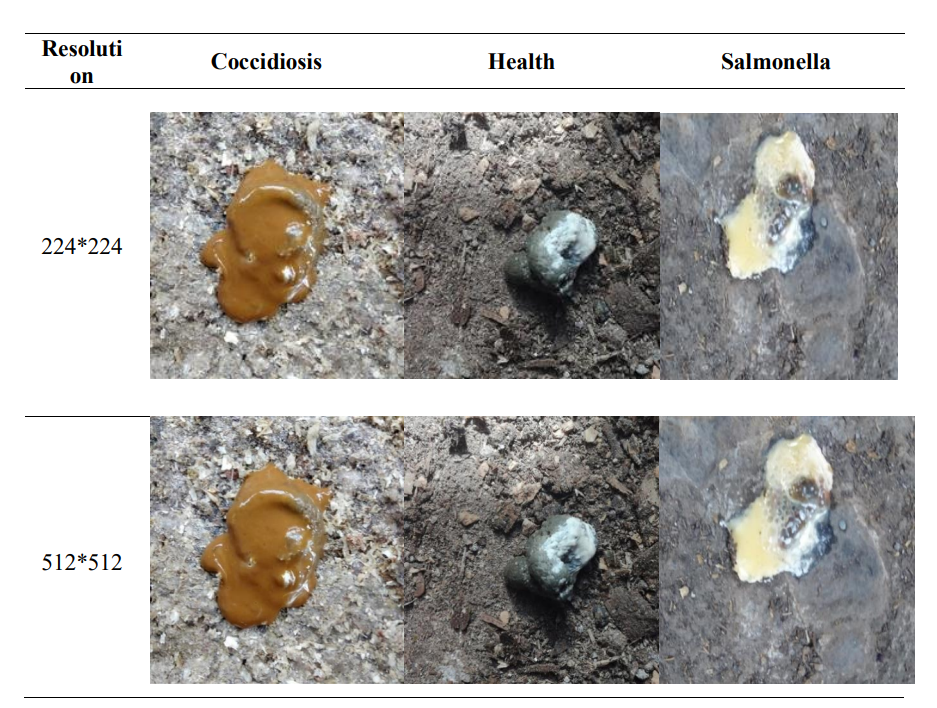
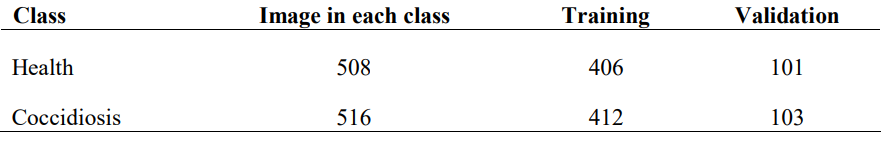


Fig 4.1: The sizes of images in the dataset changed to 224\*224 and 512\*512

**4.2.2 Training the Baseline Models**

One baseline model Resnet 50 were trained on two classes Health and Coccidiosis. The 508 images for health and 516 images for Coccidiosis. In training the models, the dataset was split 80-20, 80% of dataset for training and 20% for validation respectively. From a dataset of 1024 images, 818 images for training and 206 images for validation were obtained for the two classes as shown in Table 4.2.

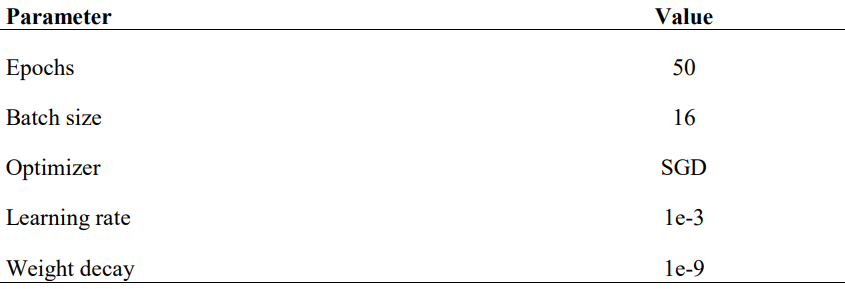
Table 4.1: Dataset splitting for training baseline models



Training the baseline models included different hyperparameters learning rate, epochs, batch size and optimizer including:

* + Learning Rate: This parameter controls how much to change the model in response to the estimated error each time the model weights are updated.
  + Epochs: The number of rounds a model takes to train a batch of data.
  + Batch size: The number of samples that are propagated through the network in one round.
  + Optimizer: Is the extended class, which includes added information to train a specific model. Some examples of commonly used optimizers are SGD, Adam, RMSprop.

Table 4.2: Hyperparameter using Resnet50



**4.2.3 Evaluation of the baseline models**

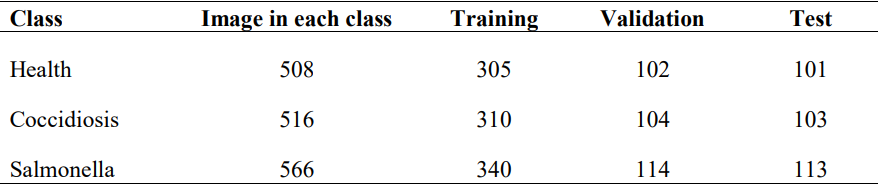
The baseline models were evaluated based on their performances. The accuracy metric was used when evaluating the models. The training accuracy, training loss, validation accuracy and validation loss were observed during training, from the 1st epoch to the 50th epoch. The metrics include:

* + Training accuracy: Is the accuracy achieved when the model is applied to the training data.
  + Training Loss: This is the error on the training set of the data.
  + Validation Accuracy: This is the accuracy used to evaluate the model’s performance.
  + Validation Loss: This is the error after running the validation set of data through the trained network.

**4.2.4 Developed Convolutional Neural Network Model**

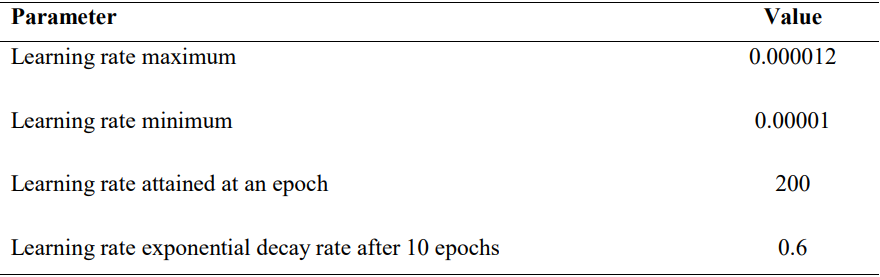
In this work, a fully developed from scratch CNN model is proposed and trained on the dataset. Initially the models are trained on three classes Health, Coccidiosis and Salmonella. A total dataset of 1590 images, 508 health, 516 Coccidiosis and 566 Salmonella. The dataset was split 60-20-20, 60% of dataset for training, 20% for validation and 20% for testing respectively. The data was split in three sets because during training phase, the model is validated based on the training set and may lead to overfitting of the model in the validation set. The test set (unseen data) makes sure that the model generalizes the problem well. Table 4.3 summarizes how the dataset was split.

Table 4.3: Dataset splitting training, validation and test sets



The training was performed in Kaggle Environment, accelerator used was TPU-v3.8 using Python language v3.7, and Tensor Flow was the backend. TensorFlow framework was used because it supports models to be deployed and interpreted in other devices. Table 5 below summarizes the hyperparameters used when training the CNN model. During training, normal stochastic gradient descent is used to minimize the error, and in evaluation, log loss and accuracy as metrics were leveraged. The findings show that some of the images from the dataset 32 may contain more than one disease; hence categorical cross-entropy loss function seems to fit the problem with the log loss as an evaluation metric. An output layer with three nodes and SoftMax activation is used. Since the problem is multiclass classification, SoftMax is the right choice because the output from the node is the likelihood for the output to be either of the three classes. The schedule learning rate is used in the experiment with some call back features. Call backs are used in order to store the best performing model ensuring that the test set accuracy is equal or greater than the validation set accuracy. When training the CNN model, a number of epochs is set to 500 which is a large number of iterations because the developed dumb converging neural network with no weights is dumb. The He uniform initializer is then used to random initialize the weights during training. After training the CNN model, transfer learning was applied and train the dataset on pre-trained models.

Table 4.4: Hyperparameters for training the Convolutional Neural Network model



**4.2.5 Evaluation of the Pre-trained Models**

The models were evaluated based on different metrics. Evaluation metrics used were accuracy, precision, recall, F1 score, and log loss as explained below:

* + Accuracy: It calculates how often predictions are equal to labels.
  + Precision: Number of correctly identified results divided by the number of all positive results.
  + Recall: Number of correctly identified positive results divided by the number of all samples that should have been identified as positive.
  + F1 score: It is a measure of test accuracy.
  + Log loss: This is a classification function often used as an evaluation metric often used in multiclass classification.

It quantifies the accuracy of a classifier by penalizing false classifications. The log loss was minimized by assigning a probability to each class in this problem resulting in maximizing the accuracy of the model. The models were set to train the dataset on 50 epochs with 8 steps per epoch, resulting in 400 iterations to complete the training process. The models were trained observing the accuracy and loss trend in both training and validation sets. The logarithmic loss was used to evaluate the models' performance on the test set whether the prediction was correct. Then the training and validation accuracy and loss against the number of epochs. The study presents and discusses the results in the next chapter.

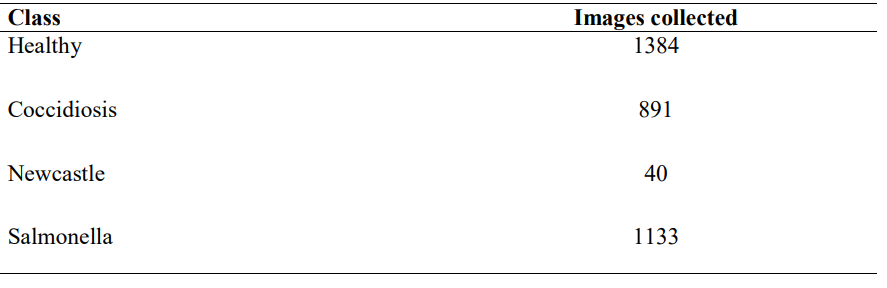
**CHAPTER 5**

**RESULTS AND DISCUSSION**

In this chapter result and discussion all the results are display which is obtained by using the models.

In the review, activity round table discussions were held with the veterinary officers and identified the target clinical signs of the three diseases focused in this study. The study identified the digestive clinical signs of the diseases mainly the occurrence of the chicken droppings. The work also learned how the chicken droppings can be used for early identification of the diseases thus guidance to the data collection procedure. The target was to have a dataset of 4000 images, 1000 images for each class. A dataset of 3448 labeled images suitable for classification tasks was collected. Table 6 illustrates the dataset distribution.

Table 5.1: Dataset



**5.1 Results for the Baseline Models check**

The performance of the baseline models was evaluated based on training of the dataset for binary classification. The Resnet 50 models were trained on the data shown in Table 2. The metrics used were training and validation accuracy, as shown in the plots below.

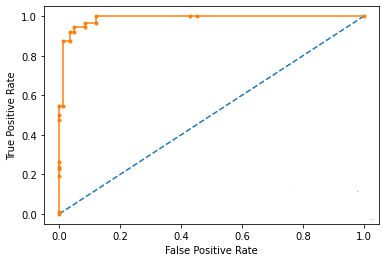


Fig 5.1: Training and validation plots for baseline model

Figure 13 illustrate the trend in the training and validation accuracy, and training and validation loss while training Resnet 50 model for 50 epochs. Table 7 shows the accuracies and losses for the model Res-Net 50 an accuracy of 90.7%.

Table 5.2: Training and validation results for baseline model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Training accuracy | Training loss | Validation accuracy | Validation loss |
| Resnet 50 | 0.8527 | 0.0723 | 0.9077 | 0.3718 |

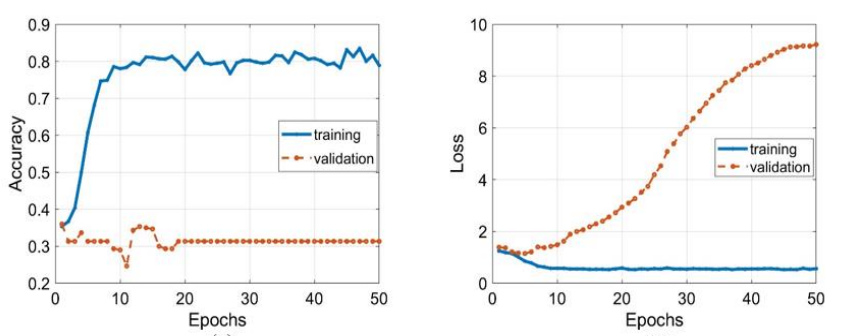


Fig 5.2: Training and validation plots for the Resnet 50 model

The performance of the Resnet 50 was good as it can able to identify the abnormality of the poultry through the images we have collected till yet and predict it accurately about its diseases. The final result are given in fig 14 and its subparts.



Fig 5.3: After successfully predicting the diseases cocci



Fig 5.4: After successfully predicting the diseases New Castle

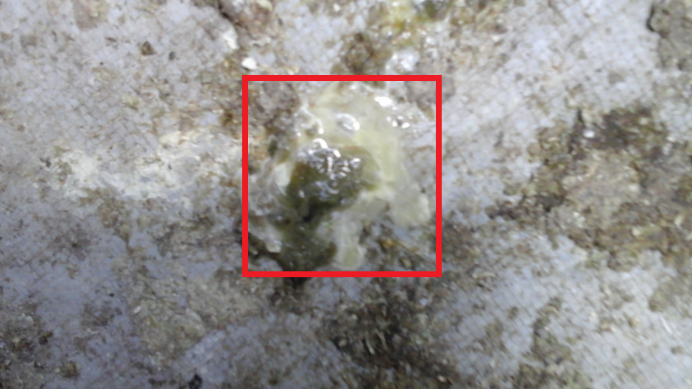


Fig 5.5: After successfully predicting the diseases Salmo

The result came out from the model Retina-Net i.e. the Epoch vs Accuracy graph of the Retina-Net model obtained in fig 15.

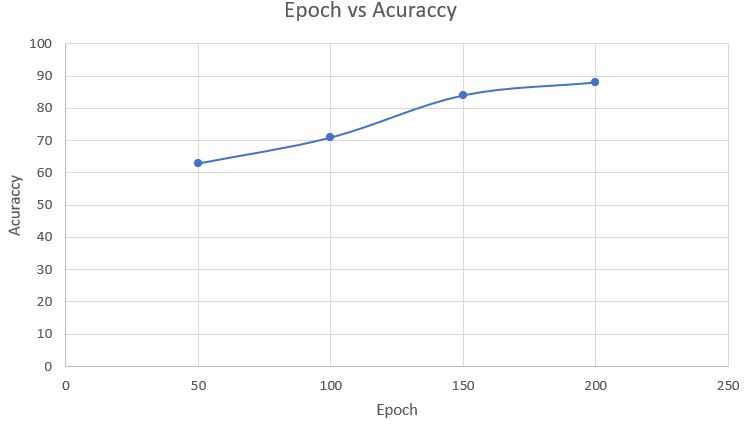


Fig 5.6: Epoch Accuracy graph of Retina-Net

The result came out from the model YOLOv8 i.e. the Epoch vs Accuracy graph of the YOLOv8 model obtained in fig 16.

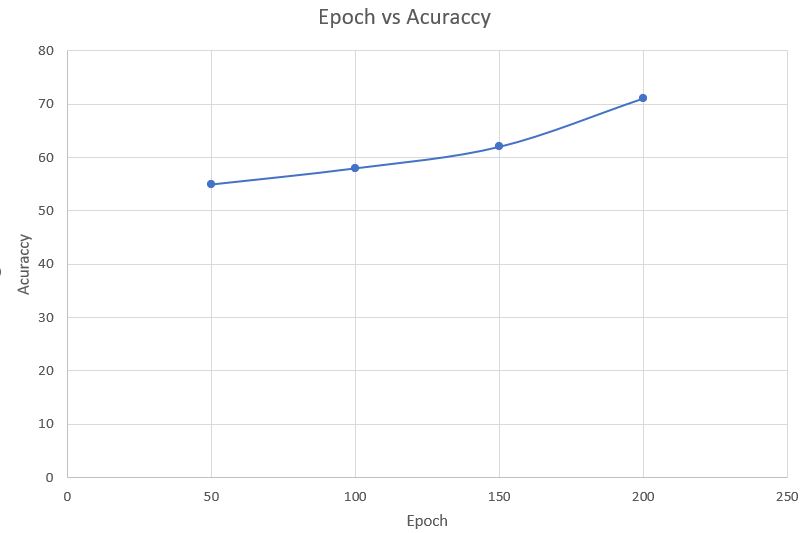


Fig 5.7: Epoch Accuracy graph of YOLOv8

The result of the Grad-CAM model after deployment of the model on both Retina-Net & YOLOv8 the result obtained from the model is the Heat map and the superimposed of the image which came out after successfully deployment of this models.

Fig 5.8: After successfully deployment of Grad-CAM model using YOLOv8

Fig 5.9: After successfully deployment of Grad-CAM model using Retina Net

**5.3 Discussion**

The study starts with training a portion of the dataset (two classes coccidiosis and healthy) binary classification on the baseline models to observe the behavior of the data and validation of the requirements. Then proposes a Resnet 50 model and trains it on three classes for multiclass classification, and it achieves an accuracy of 90%.

**CONCLUSION**

In conclusion, the proposed poultry disease prediction system utilizing ResNet-50 has demonstrated promising results in automatically detecting and classifying various poultry diseases. By analyzing both poultry images and clinical data, the system has achieved encouraging accuracy and generalizability, potentially revolutionizing the way poultry diseases are diagnosed and prevented.

**Key findings:**

* ResNet-50, a powerful deep learning model, effectively extracts relevant features from poultry images for disease classification.
* Integrating clinical data enhances the system's understanding of disease patterns and improves its overall performance.
* Data augmentation techniques significantly contribute to the model's robustness and generalizability to unseen data.
* The system offers several advantages over traditional disease detection methods, including automation, early detection, and improved disease surveillance.

**Potential impact:**

* Early detection of poultry diseases can minimize economic losses and safeguard public health by preventing outbreaks and ensuring safe food production.
* Improved disease surveillance enables early intervention and effective control of disease spread, further protecting poultry flocks and public health.

**Overall, the poultry disease prediction system using ResNet-50 shows significant promise for revolutionizing poultry healthcare by providing a reliable, efficient, and scalable tool for early and accurate disease detection.** This technology has the potential to significantly improve poultry health and welfare while safeguarding public health and food security.

**FUTURE WORK**

Moving forward, the logical progression involves the development of a user-friendly mobile application for poultry disease detection, leveraging the power of Retina-Net & YOLOv8 with the Retina-Net model and explainable AI (XAI) Grad-CAM model. Such an app would democratize access to advance the poultry disease detection capabilities, empowering users to perform self-assessments and seek timely medical intervention when necessary. The user interface will be designed to be intuitive and straightforward, allowing users to upload images of the feather & poops of the poultry and receive instant feedback on their risk level. The app will utilize the pre-trained Retina-Net or YOLOv8 model for accurate classification and integrate Grad-CAM for transparent visualization of the model's decision-making process, ensuring users can understand and trust the results. Additionally, the app could feature educational resources on poultry disease prevention and early detection for the farm owners, further promoting awareness and proactive healthcare practices. Through iterative testing and user feedback, the app can be refined to meet the specific needs and preferences of its target audience, ultimately serving as a valuable tool in the fight against poultry disease which can create a huge loss on a global scale.

**REFERENCE**

[1] Abdisa, T., &Tagesu, T. (2017). Review on Newcastle Disease of Poultry and its Public Health

Importance. Journal of Veterinary Science and Technology, 08(03), 1-7. https://doi.org/

10.4172/2157- 7579 .1000441

[2] Albarqouni, S., Baur, C., Achilles, F., Belagiannis, V., Demirci, S., &Navab, N. (2016).

AggNet: Deep Learning from Crowds for Mitosis Detection in Breast Cancer Histology

Images. Transactions on Medical Imaging, 35(5), 1313–1321. https://doi.org/10.1109/

TMI.2016.2528120

[3] Bojarski, M., Del Testa, D., Dworakowski, D., Firner, B., Flepp, B., Goyal, P., Jackel, L. D.,

Monfort, M., Muller, U., Zhang, J., Zhang, X., Zhao, J., & Zieba, K. (2016). End to End

Learning for Self-Driving Cars. http://arxiv.org/abs/1604.07316

[4] Brooks-Pollock, E., de Jong, M. C. M., Keeling, M. J., Klinkenberg, D., & Wood, J. L. N.

(2015). Eight challenges in modelling infectious livestock diseases. Epidemics, 10, 1–5.

https://doi.org/10.1016/j.epidem.2014.08.005

[5] Cambria, E., & White, B. (2014). Jumping NLP curves: A review of natural language

processing research. Computational Intelligence Magazine, 9(2), 48–57. https://

doi.org/10.1109/ MCI.2014. 2307227

[6] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions.

Proceedings - 30th Conference on Computer Vision and Pattern Recognition, CVPR

2017, January, 1800–1807. https://doi.org/10.1109/CVPR.2017.195

[7]Dawud, A. M., Yurtkan, K., &Oztoprak, H. (2019). Application of deep learning in

neuroradiology: Brain haemorrhage classification using transfer learning. Computational

Intelligence and Neuroscience, 2019, 1-12. https://doi.org/10.1155/2019/4629859

[8]Desin, T. S., Köster, W., & Potter, A. A. (2013). Salmonella vaccines in poultry: Past, present and future. Expert Review of Vaccines, 12(1), 87–96. https://doi.org/10.1586/erv.12.138

FAO. (2013). Poultry Development Report. http://www.fao.org/3/i3531e/i3531e.pdf

42

[9] Felip, V. P. S., Silv, M. A., Valent, B. D., & Ros, G. J. M. (2014). Using Multiple Regression, Bayesian Networks and Artificial Neural Networks for Prediction of Total Egg

Production in European Quails. Proceedings of 10th World Congress of Genetics Applied

to Livestock Production, 772–781. https://doi.org/10.3382/ps/pev031

[10]Ferentinos, K. P. (2018). Deep learning models for plant disease detection and diagnosis.

Computers and Electronics in Agriculture, 145(2018), 311–318. https://doi.org/

10.1016/j.compag.2018.01.009

[11] Grilli, G., Borgonovo, F., Tullo, E., Fontana, I., Guarino, M., & Ferrante, V. (2018). A pilot study to detect coccidiosis in poultry farms at early stage from air analysis. Biosystems

Engineering, 2018, 1–7. https://doi.org/10.1016/j.biosystemseng.2018.02.004

[12]He, K., & Sun, J. (2015). Deep Residual Learning for Image Recognition. Proceedings of the IEEE Conference on computer Vision and Pattern Recognition (CVPR), 770-778.

https://openaccess.thecvf.com/contentcvpr 2015/html/He Deep Residual Learning

Hemalatha, Muruganand, S., & Maheswaran, R. (2014).

[13]Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). MobileNets: Efficient Convolutional Neural Networks for Mobile

Vision Applications. http://arxiv.org/abs/1704.04861

[14]Iorga, C., & Neagoe, V. E. (2019). A Deep CNN Approach with Transfer Learning for Image Recognition. Proceedings of the 11th International Conference on Electronics,

Computers and Artificial Intelligence, ECAI 2019. https://doi.org/10.1109/ECAI

46879.2019.9042173

[15]Kumar, S., Pandey, A., Sai, S, K., Kumar, S., Singh, S. K., Singh, A. K., & Mohan, A. (2018). Deep learning framework for recognition of cattle using muzzle point image pattern.

Measurement: Journal of the International Measurement Confederation, 116, 1–17.

https://doi.org/10.1016/j.measurement.2017.10.064

[16]Lee, H., Eum, S., & Kwon, H. (2019). Is Pretraining Necessary for hyperspectral image

classification? International Geoscience and Remote Sensing Symposium (IGARSS),

3321–3324. https://doi.org/10.1109/IGARSS.2019.8898734

[17]Liakos, K. G., Busato, P., Moshou, D., Pearson, S., &Bochtis, D. (2018). Machine learning in agriculture: A review. Sensors (Switzerland), 18(8), 1–29. https://doi.org/10.3390/s

18082674

[18]Lim, L., Tay, Y., Alias, H., Wan, K., & Dear, P. H. (2012). Insights into the genome structure and copy-number variation of Eimeria tenella. Springer, 13(389), 1-10.

Lwoga, E., Ngulube, P., & Stilwell, P. (2010). Information needs and information seeking

behaviour of small-scale farmers in Tanzania. Journal of appropriate librarianship and

information work in Southern Africa, 2010(40). https://doi.org/10.4314/innovation.v

40i1. 60088

[19]Makantasis, K., Karantzalos, K., Doulamis, A., &Doulamis, N. (2015). Deep supervised

learning for hyperspectral data classification through convolutional neural networks.

International Geoscience and Remote Sensing Symposium, 2015-(11), 4959–4962.

https://doi.org/10.1109/IGARSS.2015.7326945

[20]Ministry of Livestock and Fisheries. (2015). Tanzania Livestock Modernization Initiative.

Ministry of Livestock and Fisheries Development: United Republic of Tanzania.

https://livestocklivelihoodsandhealth.org/wpcontent/uploads/2015/07/Tanzania\_Livesto

ck\_Modernization\_Initiative\_July\_2015.pdf

[21]Morales, I. R., Cebrián, D. R., Fernandez-Blanco, E., & Sierra, A. P. (2016). Early warning in egg production curves from commercial hens: A SVM approach. Computers and

Electronics in Agriculture, 121, 169–179. https://doi.org/10.1016/j.compag.2015.12.009

44

[22]Mulisa, D. D., W/Kiros, M. K., Alemu, R. B., Keno, M. S., Furaso, A., Heidari, A., Chibsa, T. R., &Chunde, H. C. (2014). Characterization of Newcastle Disease Virus and poultryhandling practices in live poultry markets, Ethiopia. SpringerPlus, 3(1), 1–6.

https://doi.org/10.1186/2193-1801-3-459

[23]Ning, G., Zhang, Z., Huang, C., Ren, X., Wang, H., Cai, C., & He, Z. (2017). Spatially

supervised recurrent convolutional neural networks for visual object tracking.

Proceedings - IEEE International Symposium on Circuits and Systems, 1, 1–4.

https://doi.org/10.1109/ISCAS.2017.8050867

[24]Okinda, C., Lu, M., Liu, L., Nyalala, I., Muneri, C., Wang, J., Zhang, H., & Shen, M. (2019). A machine vision system for early detection and prediction of sick birds: A broiler

chicken model. Biosystems Engineering, 188, 229–242. https://doi.org/10.1016/j.

biosystemseng. 2019.09.015

[25]Owomugisha, G., Quinn, J. A., Mwebaze, E., &Lwasa, J. (2014). Automated Vision-Based Diagnosis of Banana Bacterial Wilt Disease and Black Sigatoka Disease. International

Conference on the Use of Mobile ICT in Africa, 2014, 1-5

[26]Pu, H., Lian, J., & Fan, M. (2018). Automatic Recognition of Flock Behavior of Chickens with Convolutional Neural Network and Kinect Sensor. International Journal of Pattern

Recognition and Artificial Intelligence, 32(7), 1–15. https://doi.org/10.1142/S021800

1418500234

[27]Quiroz-Castañeda, R. E., &Dantán-González, E. (2015). Control of avian coccidiosis: Future and present natural alternatives. BioMed Research International, 2015, 1-10.

https://doi.org /10.1155/2015/430610

[28] Hepworth, P. J., Nefedov, A. V., Muchnik, I. B., & Morgan, K. L. (2012). Broiler chickens can benefit from machine learning: Support vector machine analysis of observational

epidemiological data. Journal of the Royal Society Interface, 9(73), 1934–1942.

<https://doi.org/10.1098/rsif.2011.0852>

[29] Recognition of Poultry Disease in Real Time. Proccedings of the International Conference on Inter-Disciplinary Research in Engineering and Technology 2014. https://edlib.net/2014/icidret/icidret2014008.pdf

