استهلال

قال تعالى:

۞ اللَّهُ نُورُ السَّمَاوَاتِ وَالأَرْضِ مَثَلُ نُورِهِ كَمِشْكَاةٍ فِيهَا مِصْبَاحٌ الْمِصْبَاحُ فِي زُجَاجَةٍ الزُّجَاجَةُ كَأَنَّهَا كَوْكَبٌ دُرِّيٌّ يُوقَدُ مِنْ شَجَرَةٍ مُبَارَكَةٍ زَيْتُونَةٍ لا شَرْقِيَّةٍ وَلا غَرْبِيَّةٍ يَكَادُ زَيْتُهَا يُضِيءُ وَلَوْ لَمْ تَمْسَسْهُ نَارٌ نُورٌ عَلَى نُورٍ يَهْدِي اللَّهُ لِنُورِهِ مَنْ يَشَاءُ وَيَضْرِبُ اللَّهُ الأَمْثَالَ لِلنَّاسِ وَاللَّهُ بِكُلِّ شَيْءٍ عَلِيمٌ ۞

سورة النور ﴿35﴾

DEDICATION

To our parents, the candles
That enlighten our
Life path
To our sisters and brothers
The flowers and hope
To our teachers, the guidance
Lamp posts for fact
To everyone who contributes
To humanity
Through science
We dedicate
This humble effort

ACKNOWLEDGMENT

We offer great thanks to Dr. Hisham Ahmed Ali who push this effort forward to reach this stage through his guidance and fruitful comments. We provide great gratitude should be reflected to Mohamed Fadul for his Support and advice to make this work blossom. we are thankful to Dr. Azza for her effort as graduation projects coordinator.

ABSTRACT

In recent years, the use of technology in poultry farms has increased, and this appears in the increase in closed system farms. The main problem facing farmers is the diseases that affect chickens and the rapid spread of the disease among chickens. The project provides a suitable environment for poultry farms and increases chickens' production. The proteus software has been used for the simulation of the circuits; which consist of a Digital Temperature and Humidity sensor -DHT11-, Light Dependent Resistor sensor -LDR-, fan, and lamp all connected to a Raspberry Pi. The DHT11 sensor senses both humidity and temperature and the LDR sensor sense the lighting level, all data from sensors are fed to the Raspberry Pi in order to control the fan and the lamp. The Node-Red software has been used for hardware implementation. Then image processing has been done using -the YOLO version 5 model- to train and detect the chicken then the feature of Swelling of the face and eyes. The farmer can supervise the status of the farm through his smart device that presents the lighting level, temperature, and humidity as well as the detection of Infectious Coryza. Also, it allows controlling ventilation as well as the lighting of the farm. Out of 10 times of operating the system 8 where a fine result on the Raspberry pi as well as operating on the node red dashboard, that for DHT11 sensor, but for LDR sensor the light level wasn't accurate due to the use of capacitor with it.

المستخلص

في الآونة الأخيرة ازداد توجه شركات الدواجن في السودان لاستخدام المزارع المغلقة التي تستخدم فيها التكنلوجيا لزيادة الإنتاجية. من أهم المشاكل التي تواجه قطاع الدواجن هي سرعة انتشار الأمراض بين الدواجن. المشروع يهدف إلى توفير البيئة المناسبة لنمو الدواجن وتقليل انتشار الأمراض في مزارعها. بداية تمت محاكاة دائرة التحكم باستخدام برنامج بروتوس حيث يتحكم الراسبيري باي في التهوية والإضاءة بناء على معطيات حساس الحرارة والرطوبة وحساس الإضاءة (المقاومة الضوئية). لبرمجة الدائرة وتطبيقها عملياً تم استخدام برنامج نود ريد، حيث أتاح لأصحاب المزارع متابعة حالة المزرعة من حيث درجة الحرارة والرطوبة والإضاءة والتحكم فيها عن طريق معدات الانارة والتهوية بضغطة زر عبر تطبيق ذكي في أجهزتهم الذكية. وأيضا يظهر التطبيق الذكي الدجاج المصاب بمرض كوريزا (الزكام المعدي للطيور) وذلك باستخدام معالجة الصور (عن طريق استخدام نموذج خوارزمية يولو الإصدار الخامس) يكشف اعراض انتفاخ الوجه والعينين. تم تجربة النظام 10 تجارب، 8 من التجارب كانت ناجحة.

Table of Contents

استهلال .... **i**

DEDICATION .... **ii**

ACKNOWLEDGMENT .... **iii**

ABSTRACT .... **iv**

المستخلص .... **v**

LIST OF FIGURES .... **viii**

LIST OF ABBREVIATIONS .... **x**

CHAPTER ONE: INTRODUCTION **1**

1.1 Introduction 1

1.2 Problem Statement 3

1.3 Proposed solution 3

1.4 Research Objectives 4

1.5 Methodology 4

CHAPTER TWO: BACKGROUND AND RELATED WORK **5**

2.1 Introduction 5

2.2 Background 5

2.2.1 Hardware components 5

2.2.2 Software Component 13

2.3 Related work 22

2.3.1 Poultry production problems 22

2.3.2 Use of technology in poultry farms 23

CHAPTER THREE: METHODOLOGY **31**

3.1 Introduction 31

3.2 System overview 31

3.3 Simulation 34
3.4 Control part 35
3.4.1 Raspberry pi model B+ software installation 35
3.4.2 Node-Red installation 36
3.4.3 Install nodes in Node-Red 36
3.4.4 Digital Temperature and Humidity sensor (DHT11) flow 37
3.4.5 Light Dependent Resistor (LDR) flow 39
3.5 Image processing part 39
3.5.1 Data-set 39
3.5.2 YOLO V5 and training 44
3.5.3 PyTorch and Visual studio 46
CHAPTER FOUR: RESULTS AND DISCUSSION 49
4.1 Introduction 49
4.2 Control part 49
4.2.1 Digital Temperature and Humidity sensor (DHT11) results 49
4.2.2 Light Dependent Resistor (LDR) results 50
4.3 Image processing part 51
4.4 The Platform 57
CHAPTER FIVE: CONCLUSION AND RECOMMENDATION 56
5.1 Introduction 56
5.2 Conclusion 56
5.3 Recommendation 57
REFERENCE 60
APPENDIX A 66
APPENDIX B 67
APPENDIX C 69

LIST OF FIGURES

2.1	Raspberry Pi V3 B	7
2.2	Digital Temperature and Humidity sensor (DHT11)	8
2.3	Internal Structure of Humidity Sensor	8
2.4	Light Dependent Resistor sensor (LDR)	10
2.5	2-wire fan	10
2.6	Light Emitting Diode (LED)	11
2.7	Liquid Crystal Display (LCD)	11
2.8	Raspberry Pi camera module V2	12
2.9	MCP3208	12
2.10	Linear Monolithic (LM35)	13
2.11	1µF 50V Capacitor	14
2.12	The anatomy of an object	16
2.13	The network architecture of YOLO v5	16
2.14	Intersection over Union (IoU) equation	19
3.1	System block diagram	33
3.2	control part hardware implementation circuit	34
3.3	: Image processing part implementation hardware circuit	35
3.4	Simulation	36
3.5	Simulation flow chart	37
3.6	Digital Temperature and Humidity sensor (DHT11) flow	39

3.7	Light Dependent Resistor (LDR) flow	41
3.8	Choosing images size	42
3.9	Create a list of labels	42
3.10	Creating labels for images	43
3.11	Uploading images to Robo-flow	44
3.12	Preparing images for training	44
3.13	Split images to categories	45
3.14	Training the model	46
3.15	Installing PyTorch and Anaconda	49
3.16	Image processing flow chart	50
4.1	Screen shot of Digital Temperature and Humidity sensor (DHT11) readings on Node-Red dashboard	51
4.2	Screen shot of Light Dependent Resistor sensor (LDR) readings on Node-Red dashboard	52
4.3	overall results for control circuit	53
4.4	Plots of the precision, recall Mean Average Precision (mAP) -0.5- parameters along with class object loss for training epochs	54
4.5	The improvement of model by changed the epochs & cache	54
4.6	Performance of the model YOLOv5s for custom Robo-flow data	55
4.7	Detecting infectious coryza	56
4.8	Platform	57

LIST OF ABBREVIATIONS

ACF	Adaptive Correlation Filters
ADC	Analogue to Digital Converter
AI	Artificial Intelligence
AP	Average Precision
BiFPN	Weighted Bi-directional Feature Pyramid Network
BLE	Bluetooth Low Energy
CNN	Convolution Neural Networks
CSP	Cross Stage Partial Networks
DHT11	Digital Temperature and Humidity sensor
DMA	Direct Memory Access
FFNN	Feed-Forward Neural Network
FPN	Feature Pyramid Network
GPIO	General Purpose input/output
GUI	Graphical User Interface
IDE	Integrated Development Environment
ILO	International Labor Organization
IoU	Intersection over Union
IOT	Internet of Things
KCF	Kernel Correlation Filter

LAN	Local Area Network
LCD	Liquid Crystal Display
LDR	Light Dependent Resistor sensor
LED	Light Emitting Diode
LM35	Linear Monolithic
mAP	Mean Average Precision
MIL	Multitask learning Algorithm
MS	Mean Shift Algorithm
NTC	Negative Temperature Coefficient
PC	Personal Computer
RNN	Recurrent Neural Networks
SOIC	Small Outline Integrated Circuit
TLD	Tracking Learn Detection
VB	Visual Basic
WSN	Wireless Sensor Network
YOLO V5	You Only Look Once Version

CHAPTER ONE

INTRODUCTION

1.1 Introduction:

Across the world, a huge decline of the workforce is observed due to many reasons, like the lack of skilled labor, aging of farmers, and young farmers finding farming an unattractive profession, thus encouraging trends for automated farming operations. According to the International Labor Organization (ILO), agricultural labor in the percentage of the workforce declined from 81% to 48.2% in developing countries. Also, developed countries are not an exception in such a huge decline. In Asia-Pacific, where agriculture occupies a major part of the economy, has been a huge decline in the workforce, which was nearly about 10% from 2014 to 2017. In Japan, the number of people working in farms witnessed a steep fall to 1.7 million in the year 2015, a 15% decline from the previous year. The European agriculture sector also faced such a huge decline in the workforce, which is nearly accounting for 12.8% for the corresponding period. The trend of decline in the agricultural workforce is encouraging government and private organizations to focus on agricultural automation operations by adopting automated poultry farm equipment. [1]

What would make for better poultry production?

From a production standpoint, individual real-time body weights, feed, and water consumption.

From a husbandry and welfare perspective, knowing the stress levels in the bird and bird comfort is assessed through body temperatures and air quality factors, such as carbon dioxide and ammonia.

From a disease management outlook, the ability to spot disease or find morbid birds before the entire flock is affected.

From a food safety perspective, enhanced Salmonella, Campylobacter, and E. coli detection.

From a food processing perspective, increased yield. [**2**]

In general, the poultry industry manages chicken, turkey, duck, goose, and ostrich. The disease can spread rapidly among poultry because they are always kept together in a cage or chicken house. Infectious Bronchitis, Avian Influenza, Infectious Sinusitis, Fowl Pox, and Infectious Coryza are common diseases among the poultry. Coughing, Sneezing, shaking the head, Rales (abnormal breathing sound), Gasping, Discharge from eyes, Nasal discharge, swelling of face, and Paralysis are the common symptoms of those disease. The diseases are classified into bacterial, viral, parasitic, and fungal diseases. Most of these diseases can be analyzed and identified based on the sound, video, and temperature of hens. If we analyze the rale then we can decide whether the hen is sick or not. The proposed model used this idea to capture audio, video, and temperature using various sensors, screen them to decide whether the hen is sick t not, and predict the appropriate disease. The proposed framework can be used to monitor and identify sick hens as soon as they get affected. Combining the Image analysis for motion pattern and thermal sensor analysis for temperature pattern of the sick and normal hens for better prediction and classification. We can extend the image analysis for movement and behavior analysis also. Integration of the sensor results and producing hybrid and accurate results is our current ongoing research. [3]

Machine learning is a method of data analysis that automates analytical model building. It is a branch of artificial intelligence based on the idea that systems can learn from data, identify patterns and make decisions with minimal human intervention.

Machine learning types:

1- supervisor: labels data; data that already contains the solution. The common tasks called (CATEGORICAL AND CONTINUOUS)

2- un supervisor: unlabeled data; don't need to provide the model with any kind of label or solution while the model is being trained. The common task called (CLUSTERING)

3- reinforcements: In reinforcements, the algorithm figures out the solution to learn from new situations by using a trial-and-error method.

1.2 Problem Statement:

One of the major problems in the poultry industry is how to provide a healthy environment for poultry to grow healthy, different weather conditions like temperature, humidity, and light are important to remain in a good condition for the chickens to grow healthy, in general in the poultry industry the ideal temperature is between 21°C – 35°C depends on chickens age and humidity between 50%-70%. The nightmare for poultry projects owners is if a disease enters the poultry farm and is not detected early this causes a large number of chickens lost, so there is a need to detect ill chickens early before the disease spreads among chickens.

1.3 Proposed solution:

The project will help farmers monitoring the environment for poultry farms, by using different sensors to measure temperature, humidity, and lightening to control the temperature, lightening and ventilation of the farm by operating a fan and a lamp to remain the farm in a good condition for the chickens to grow healthy. Also, we will use image processing to detect ill chickens.

1.4 Research Objectives:

The project is aiming to provide a suitable environment for poultry farms, our objectives:

- To monitor the environmental factors of the poultry farm to protect chickens from diseases.
- To increase chickens' production and reduce labor.

1.5 Methodology:

The system contains hardware and software components, to gather the data of the poultry farm environment and analyze it to show it on a cloud platform. To implement the project, we design hardware circuits (sensors circuit, control circuit, and image processing part circuit), design frame-ware(software), then perform simulation to test the software and its integration with hardware, at last**,** integrate the software with our actual hardware circuits and cloud platform then test it until it gives us the desire result.

CHAPTER TWO

BACKGROUND AND RELATED WORK

2.1 Introduction:

This chapter start with background in section 2.2, then literature review in section 2.3, and an overview about poultry production problems in section 2.3.1 Section 2.3.2 shows the use of technology in poultry farms.

2.2 Background:

The system consist of hardware and software components.

2.2.1 Hardware components:

It consist of Raspberry Pi V3 B, DHT11 sensor, LDR sensor, 2-wire fan, LED, MCP3208, LM35, and Raspberry Pi camera module V2.

2.2.1.1 Raspberry Pi V3 B:

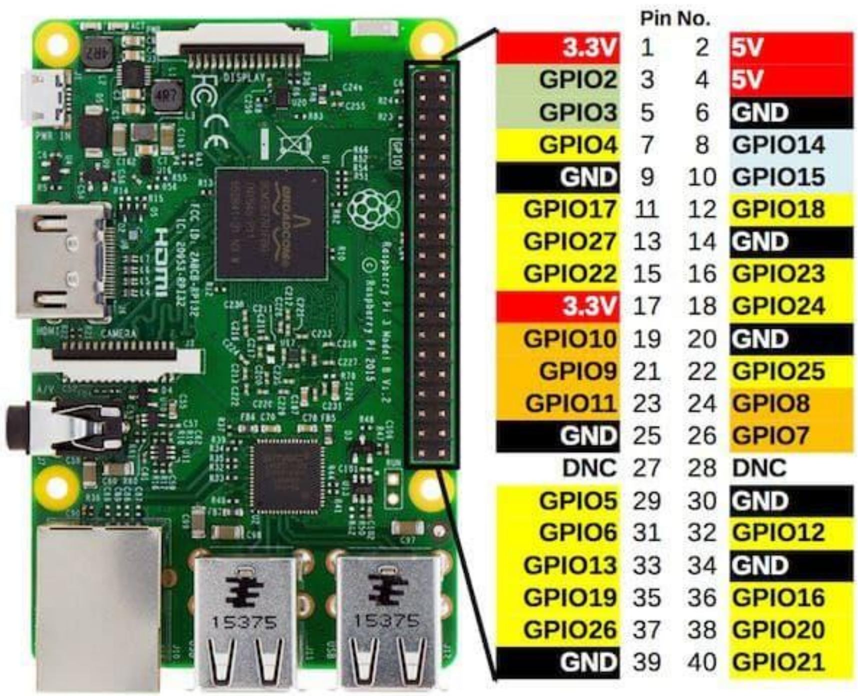

Figure 2-1: Raspberry Pi V3 B[4]

The Raspberry Pi 3 Model B Figure 2-1 is the third generation Raspberry Pi, single board computer can be used for many applications with Broadcom BCM2837 64 bit ARMv8 QUAD Core 64 bit Processor powered Single Board Computer running at 1.2 GHz, 1 GB RAM, BCM43143 WiFi on board, Bluetooth Low Energy (BLE) on board, 40 pin extended GPIO, 4 USB2 ports, 4 pole Stereo output and Composite video port, Full size HDMI, CSI camera port for connecting the Raspberry Pi camera, DSI display port for connecting the Raspberry Pi touch screen display, Micro-SD port for loading your operating system and storing data, Upgraded switched Micro USB power source (now supports up to 2.5 Amps).[5]

2.2.1.2 Digital Temperature and Humidity sensor (DHT11):

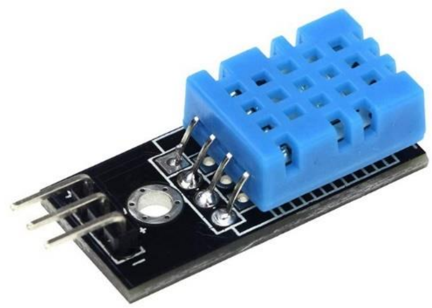

Figure 2-2: Digital Temperature and Humidity sensor (DHT11) [6]

The DHT11 Figure 2-2 is a commonly used Temperature and humidity sensor that comes with a dedicated NTC to measure temperature and an 8-bit microcontroller to output the values of temperature and humidity as serial data. Operating Voltage: 3.5V to 5.5V, operating current: 0.3mA (measuring) 60uA (standby), output: Serial data, temperature range: 0°C to 50°C, humidity range: 20% to 90%, resolution: temperature and humidity both are 16-bit, accuracy: ±1°C and ±1%.[7]

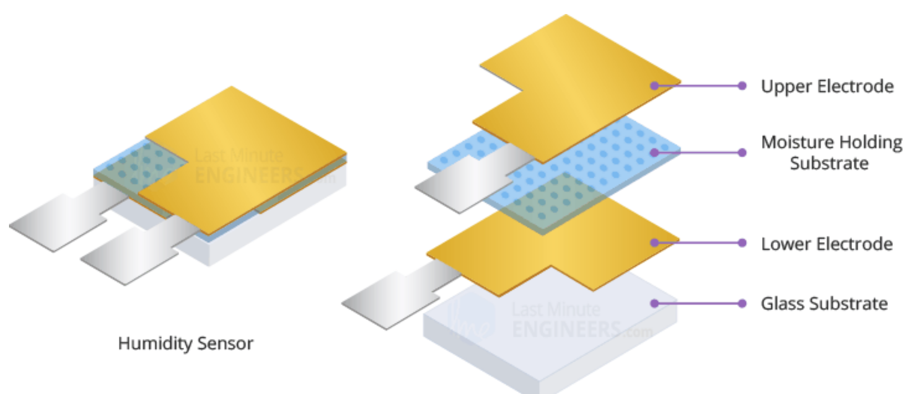

Figure 2-3: Internal Structure of Humidity Sensor[8]

The casing is in two parts so to get inside it is just a matter of getting a sharp knife and splitting the case apart. Inside the case, on the sensing side, there is a humidity sensing component along with a Negative Temperature Coefficient (NTC) temperature sensor (or thermistor)

Humidity sensing component is used, of course to measure humidity, which has two electrodes with moisture holding substrate (usually a salt or conductive plastic polymer) sandwiched between them. The ions are released by the substrate as water vapor is absorbed by it, which in turn increases the conductivity between the electrodes. The change in resistance between the two electrodes is proportional to the relative humidity. Higher relative humidity decreases the resistance between the electrodes, while lower relative humidity increases the resistance between the electrodes.

Besides, they consist of a NTC temperature sensor/Thermistor to measure temperature. A thermistor is a thermal resistor – a resistor that changes its resistance with temperature. Technically, all resistors are thermistors – their resistance changes slightly with temperature – but the change is usually very very small and difficult to measure.

Thermistors are made so that the resistance changes drastically with temperature so that it can be 100 ohms or more of change per degree! The term “NTC” means “Negative Temperature Coefficient”, which means that the resistance decreases with increase of the temperature.

On the other side, there is a small printed circuit board (PCB) with an 8-bit small outline integrated circuit (SOIC) -14 packaged Integrated circuit (IC). This IC measures and processes the analog signal with stored calibration coefficients, does analog to digital conversion and spits out a digital signal with the temperature and humidity. [8]

The equation for calculating relative humidity is as follows:

$$RH = \left(\frac{\rho_w}{\rho_s}\right) \cdot 100\%$$

where \rho_w is the density of water vapor, while \ rho_s is the density of saturated water vapor.

The relative humidity is expressed in percentages, so that condensation occurs at 100% humidity and completely dry air at 0%.[9]

2.2.1.3 Light Dependent Resistor sensor (LDR):

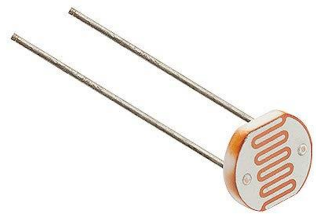

Figure 2-4: Dependent Resistor sensor (LDR) [10]

Light dependent resistors (LDR) Figure 2-4, are light sensitive devices most often used to indicate the presence or absence of light, or to measure the light intensity. LDRs have a sensitivity that varies with the wavelength of the light applied and are nonlinear devices. [11]

2.2.1.4 2-wire fan:

Figure 2-5: 2-wire fan [12]

A 2-wire fan Figure 2-5 has power, ground, and a tachometric (“tach”) output, which provides a signal with frequency proportional to speed.[13]

2.2.1.5 Light Emitting Diode (LED):

Figure 2-6: Light Emitting Diode (LED) [14]

LEDs Figure 2-6 are electronic components (semiconductor diodes) that are capable of emitting light when an electrical current goes through them**.** LED is the acronym for light emitting diode, have a much longer lifetime (30,000 to 100,000 hours), which reduces luminaire replacement and maintenance costs, and they are excellent to be used in systems with micro controllers with TTL voltage as they work within a low range of voltage (2-3 V).[15]

2.2.1.6 7 Liquid Crystal Display (LCD):

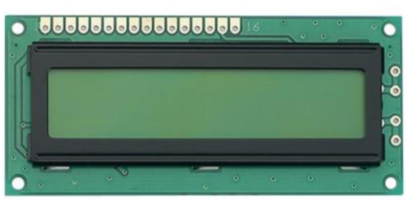

Figure 2-7: 7 Liquid Crystal Display (LCD) [16]

Liquid Crystal Display (LCD) Figure 2-7 is widely used in various electronics' applications, LCD16x2 has 2 lines with 16 characters in each line. Each character is made up of 5x8 (column x row) pixel matrix.[16]

2.2.1.7 Raspberry Pi camera module V2:

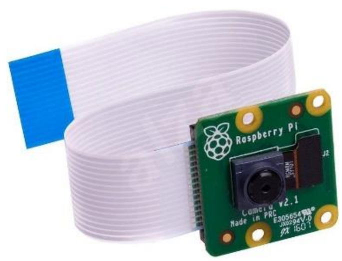

Figure 2-8: Raspberry Pi camera module V2[17]

The Raspberry Pi Camera Module v2 Figure 2-8 is a high quality 8 megapixel camera based around the Sony IMX219 image sensor - allowing you to create HD video and still photographs. ... It's capable of 3280 x 2464 pixel static images, and also supports 1080p30, 720p60 and 640x480p90 video.[18]

2.2.1.8 MCP3208:

Figure 2-9: MCP3208[19]

The MCP3208-CI/P Figure 2-9 is an 8 channel, 12bit Analogue to Digital Converter (ADC) with SPI interface in 16 pin DIP package. This ADC combines high performance and low power consumption in a small package by making it as an ideal for embedded control applications.[20]

2.2.1.10 Linear Monolithic (LM35):

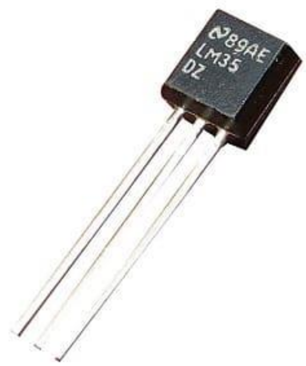

Figure 2-10: Linear Monolithic (LM35) [21]

LM35 Figure 2-10 is a temperature sensor that outputs an analog signal which is proportional to the instantaneous temperature. The output voltage can easily be interpreted to obtain a temperature reading in Celsius. The advantage of lm35 over thermistor is it does not require any external calibration. [22]

2.2.1.11 1μF 50V Capacitor:

Figure 2-11: 1μF 50V Capacitor [23]

High quality miniaturized capacitors Figure 2-11 utilize case sizes smaller than conventional general purpose caps while maintaining very high performance, low impedance characteristics, 10% tolerance, and an operating temperature of -40° to +85° C. [23]

2.2.2 Software Component:

It consists of Proteus, Pytorch, Node-Red, Yolo v5, Google colab, Make sense, Robo-flow, and Visual studio.

2.2.2.1 Proteus:

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

It was developed in Yorkshire, England by Labcenter Electronics Ltd and is available in English, French, Spanish and Chinese languages. The first version of what is now the Proteus Design Suite was called PC-B and was written by the company chairman, John Jameson, for DOS in 1988. [24]

2.2.2.2 Pytorch:

PyTorch is an open source machine learning library based on the Torch library, used for applications such as computer vision and natural language processing, primarily developed by Facebook's Artificial Intelligence (AI) Research lab. It is free and open-source software released under the Modified BSD license. [25]

2.2.2.3 Node-RED:

Node-RED provides a browser-based flow editor that makes it easy to wire together flows using the wide range of nodes in the palette. Flows can be then deployed to the runtime in a single-click.

JavaScript functions can be created within the editor using a rich text editor.

A built-in library allows you to save useful functions, templates or flows for re-use.

The light-weight runtime is built on Node.js, taking full advantage of its event-driven, non-blocking model. This makes it ideal to run at the edge of the network on low-cost hardware such as the Raspberry Pi as well as in the cloud.

With over 225,000 modules in Node's package repository, it is easy to extend the range of palette nodes to add new capabilities. [26]

2.2.2.4 Yolo v5:

An object detector is designed to create features from input images and then to feed these features through a prediction system to draw boxes around objects and predict their classes.

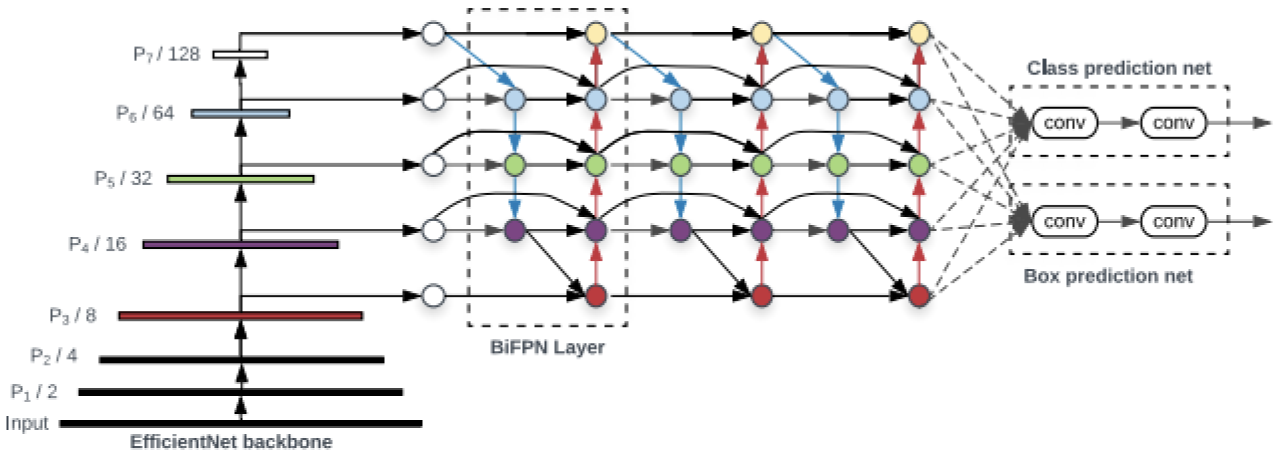

Figure 2-12: The anatomy of an object detector [27]

The YOLO model was the first object detector to connect the procedure of predicting bounding boxes with class labels in an end to end differentiable network.

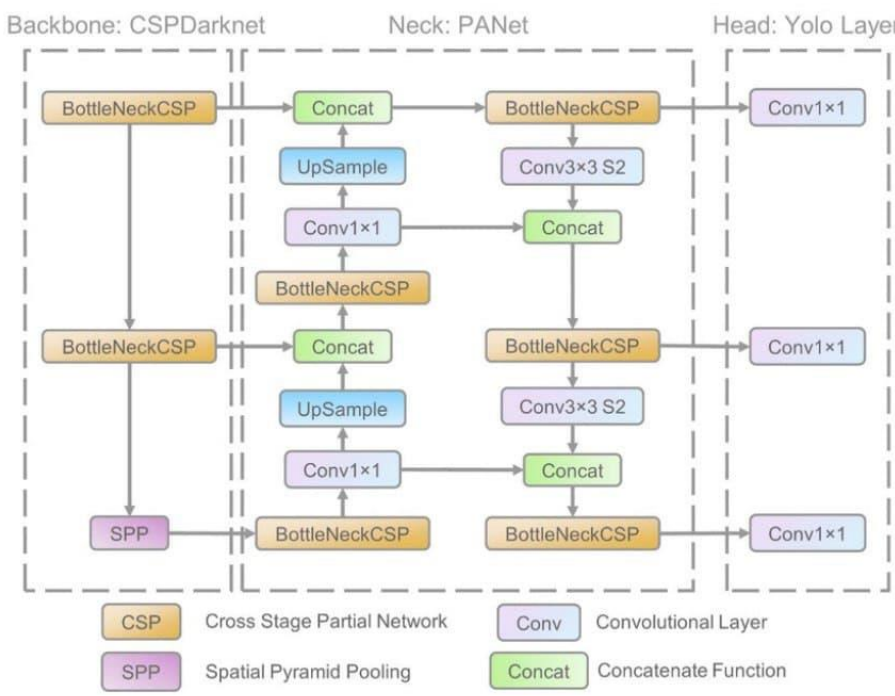

Figure 2-13: The network architecture of YOLO v5 [28]

The YOLO network consists of three main pieces.

1. Backbone: Model Backbone is mostly used to extract key features from an input image. CSP (Cross Stage Partial Networks) are used as a backbone in YOLO v5 to extract rich in useful characteristics from an input image.

2. Neck: The Model Neck is mostly used to create feature pyramids. Feature pyramids aid models in generalizing successfully when it comes to object scaling. It aids in the identification of the same object in various sizes and scales.

Feature pyramids are quite beneficial in assisting models to perform effectively on previously unseen data. Other models, such as FPN, BiFPN, and PANet, use various sorts of feature pyramid approaches. PANet is used as a neck in YOLO v5 to get feature pyramids.

3. Head: The model Head is mostly responsible for the final detection step. It uses anchor boxes to construct final output vectors with class probabilities, abjectness scores, and bounding boxes. [28]

Of course, there are many approaches one can take to combining different architectures at each major component. The contributions of YOLOv4 and YOLOv5 are foremost to integrate breakthroughs in other areas of computer vision and prove that as a collection, they improve YOLO object detection.

Training procedures are equally important to the end performance of an object detection system, though they are often less discussed.

Data Augmentation - Data augmentation makes transformations to the base training data to expose the model to a wider range of semantic variation than the training set in isolation.

Loss Calculations - YOLO calculates a total loss function from constituent loss functions - GIoU, obj, and class losses. These can be carefully constructed to maximize the objective of mean average precision.

The largest contribution of YOLOv5 is to translate the Darknet research framework to the PyTorch framework. The Darknet framework is written primarily in C and offers fine grained control over the operations encoded into the network. In many ways the control of the lower level language is a boon to research, but it can make it slower to port in new research insights, as one writes custom gradient calculations with each new addition.

The process of translating (and exceeding) the training procedures in Darknet to PyTorch in YOLOv3 is no small feat. [29]

2.2.2.5 The confusion matrix:

The confusion matrix is a very popular measure used while solving classification problems. It can be applied to binary classification as well as for multiclass classification problems.

2.2.2.5.1 The Average Precision (AP):

The average precision (AP) is a way to summarize the precision-recall curve into a single value representing the average of all precisions. The AP is calculated according to the next equation. Using a loop that goes through all precisions/recalls, the difference between the current and next recalls is calculated and then multiplied by the current precision. In other words, the AP is the weighted sum of precisions at each threshold where the weight is the increase in recall.

A better alternative is to use a quantitative measure to score how the ground-truth and predicted boxes match. This measure is the intersection over union (IoU). The IoU helps to know if a region has an object or not.

$$\mathrm{AP} = \int_0^1 p(r)dr \qquad \text{eq(1)}$$

Mean Average Precision (mAP)

2.2.2.5.2 Intersection over Union (IoU):

Intersection over Union (IoU) measures the overlap between 2 boundaries. By using that to measure how much-predicted boundary overlaps with the ground truth (the real object boundary). In some datasets, has been predefined an IoU threshold (say 0.5) in classifying whether the prediction is a true positive or a false positive. [30]

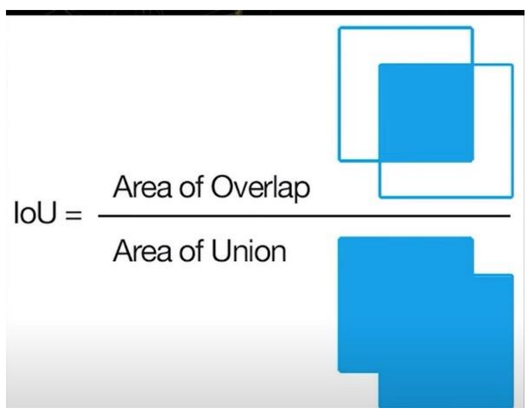

Figure 2-14: IoU equation [30]

2.2.2.5.3 Precision:

The precision is calculated as the ratio between the number of Positive samples correctly classified to the total number of samples classified as Positive (either correctly or incorrectly). The precision measures the model's accuracy in classifying a sample as positive.

$$Precision = \frac{True_{positive}}{True_{positive} + False_{positive}} \quad \text{eq(2)}$$

2.2.2.5.4 Recall:

The recall is calculated as the ratio between the number of Positive samples correctly classified as Positive to the total number of Positive samples. The recall measures the model's ability to detect positive samples. The higher the recall, the more positive samples detected.

$$Recall = \frac{True_{positive}}{True_{positive} + False_{negative}} \quad \text{eq(3)}$$

2.2.2.5.5 Accuracy:

Accuracy is a metric that generally describes how the model performs across all classes. It is useful when all classes are of equal importance. It is calculated as the ratio between the number of correct predictions to the total number of predictions**.** [30]

$$Accuracy = \frac{True_{positive} + True_{negative}}{True_{positive} + True_{negative} + False_{positive} + False_{negative}} \quad \text{eq(4)}$$

2.2.2.5 Google colab:

Colab is an online Jupyter Notebooks environment from Google. It does everything that a local notebook would do (and more) but it is in the cloud, so no software installation is necessary and it is available from any internet connected computer.

It’s very familiar-looking being similar to the Jupyter environment that you might have installed on you own PC — we’ll look at some of the differences later.

It only runs Python 3 at present (although R and Scala are mentioned as possible future developments) and it comes with many of the most popular Python libraries already installed.

And if you need to install any additional packages, you can do so using *pip* in a notebook cell. [31]

2.2.2.6 Make sense:

Make sense is a free-to-use online tool for labeling photos. Thanks to the use of a browser it does not require any complicated installation - just visit the website and you are ready to go. It also doesn't matter which operating system you're running on - we do our best to be truly cross-platform. It is perfect for small computer vision deep learning projects, making the process of preparing a dataset much easier and faster. Prepared labels can be downloaded in one of the multiple supported formats. The application was written in TypeScript and is based on React/Redux duo. [32]

2.2.2.7 Robo-flow:

Roboflow is a computer vision platform that allows users to build computer vision models faster and more accurately through the provision of better data collection, preprocessing, and model training techniques. Roboflow allows users to upload custom datasets, draw annotations, modify image orientations, resize images, modify image contrast and perform data augmentation. It can also be used to train models.

Just like I mentioned, Roboflow also has a universal annotation conversion tool that allows users to upload and convert annotations from one format to another without having to write conversion scripts for custom object detection datasets.[33]

2.2.2.8 Visual studio:

Visual Studio is an Integrated Development Environment (IDE) developed by Microsoft to develop GUI (Graphical User Interface), console, Web applications, web apps, mobile apps, cloud, and web services, etc. With the help of this IDE, you can create managed code as well as native code. It uses the various platforms of Microsoft software development software like Windows store, Microsoft Silverlight, and Windows API, etc. It is not a language-specific IDE as you can use this to write code in C#, C++, VB (Visual Basic), Python, JavaScript, and many more languages. It provides support for 36 different programming languages. It is available for Windows as well as for macOS. [34]

2.2.2.9 WordPress:

WordPress is an open source content management system, built in PHP and MySQL databases, distributed by Automatic under the GNU General Public License version 2 or higher; It is developed by a group of volunteer developers. [35]

2.3 Related work:

2.3.1 Poultry production problems:

Public concern regarding the conditions in which producing animals are maintained has led to the need for developing methods to verify minimum animal welfare standards. As defined by the World Organization for Animal Health [36], “An animal is in a good state of welfare if (as indicated by scientific evidence) it is healthy, comfortable, well nourished, safe, able to express innate behaviour, and if it is not suffering from unpleasant states such as pain, fear and distress”. However, to prove and verify animal welfare requirements in practice is not simple. In intensive poultry production a large number of factors, such as stocking density, environmental deterioration, unsuitable social environments, thermal stress, or difficulties in accessing essential resources can be major sources of stress that can lead to welfare deterioration and reduced performance. Many of these factors can be controlled through well-established management practices to provide birds with an optimal environment. However, the sharp control of the temperature and relative humidity required to minimize the occurrence of welfare problems in poultry might not be easy to achieve under high density or if the available farm equipment is inadequate. In addition, unforeseen situations or potential interactions among factors may be difficult to predict and control, thus potentially impacting on welfare. Welfare assessment serves to verify that the conditions to satisfy welfare standards during production are indeed met. [37]

Avian influenza is an infection caused by avian (bird) influenza (flu). This influenza virus occurs naturally among birds. Wild birds worldwide get flu A infections in their intestines, but usually do not get sick from flu infections.

However, avian influenza is very contagious among birds and some of these viruses can make certain domesticated bird species, including chickens, ducks, and turkeys, very sick and kill them. Infected birds can shed influenza virus in their saliva, nasal secretions. Susceptible birds become infected when they have contact with contaminated secretions or excretions. Domesticated birds may become infected with avian influenza virus through direct contact with infected waterfowl or other infected poultry. Infection with avian influenza viruses in domestic poultry causes two main forms of disease that are distinguished by low and high extremes of virulence. The "low pathogenic" form may go undetected and usually causes only mild symptoms (such as ruffled feathers and a drop-in egg production). However, the highly pathogenic form spreads more rapidly through flocks of poultry. This form may cause disease that affects multiple internal organs and has a mortality rate that can reach 90100% often within 48 hours. [38]

2.3.2 Use of technology in poultry farms:

Ayahiko et al. are doing research in a broiler-house environment system using sensor network and mail delivery system. In their work, they have combined WSN with the mail delivery system to observe environmental change in the broilerhouse. The environmental system needs to be deployed in some broiler-house to measure climatic changes. A user shall be able to inspect the summary data from the cellular phone and the data can be transmitted through a warning mail in case of a rapid temperature change. The sensor modules are deployed in each broiler-house and the network is constructed using wireless LAN communication, as the system needs to monitor two or more broiler-houses. The always-connected high-speed Internet is preferable to accumulate, to process data, and to offer it to the user in a comprehensible form. But it is difficult to provide always-connected high speed Internet at the chicken farm which is used by experiment. The server is set up in the remote place, and they propose the system that delivers data from the chicken fa rm with mail. [39]

The author proposed system that can be applicable in Poultry Farm and agriculture sector. In poultry farm, it is use to feed the food in container, maintain the temperature using water sprinkler, remove the gas using soil mixture and in Agriculture it is use to Preparation of soil, Spraying to plants, Fertilizer to plants. Through this proposed system it will be helpful to the user.

The system involves mainly two sections first to feed the food into particular contained and the second one is to control the temperature sensor to the freshness of chicken’s food. It improves poultry’s climate and reduce Labor cost and save food and chicken feeding on time and avoid contaminated food from insects. The Poultry farm uses a computer network technology. In the author study, a wireless sensor network technology is designed which monitor and control the climate of poultry farm and also humidity. The poultry management system uses hardware and open-source software. It also includes temperature, humidity, light intensity and also quality of air. System focusses to provide the setup like IOT, low-cost hardware and open-source software. System detects many problems faced by poultry industry. It saves time, dependency of labor’s and improve healthy environment, also increases poultry production. monitoring and controlling the poultry environment using a wireless Sensors GPRS network and also to take a correct action. Using Arduino, Temperature Humidity Sensor Module, and Ultrasonic sensor. In addition, the system could work on the android mobile application helping the owner to monitor the poultry farm such as food feeding function, object detection, water sprinkling, and unwanted gas reduction. The proposed system can reduce manpower and feed the food to chickens, reduce the unwanted gas, maintain temperature in farm this is fully automatic. Hence this system will be reduced cost, time, manpower, decreasing environment pollution. [40]

Project presents a flexible answer in a trial of up the accuracy in observation the environmental conditions like temperature, water level, food feeding and reducing work force for industrial household's poultry farm. A wireless sensor network (WSN) was accustomed monitor the essential environmental conditions and every one the management processes square measure finished the assistance of a Arduino ATmega2560 microcontroller. this method is capable of collection, analyzing and presenting knowledge on a Graphical interface (GUI). It conjointly permits the user to urge the updated detector data at any time through the SMS entryway service and sends alert message promptly sanctioning user interventions once required. Thereby the system minimizes the consequences of environmental fluctuations caused by unforeseen changes and reduces the gone Labor power of farm.

In accordance with the overall system architecture analysis, the implementation of the system consisted of protocol, hardware and software development, routing protocol design, for hardware design there is water level indicator, LM35 may be an exactitude IC temperature sensing element and for software design its LabVIEW is Laboratory Virtual Instrument Engineering Workbench. The system is flexible, it can be integrated into small and medium sized poultry farms with minimal modifications. Currently this system also provides many options which are user friendly enabling the farmer to manage all the necessary farming factors resulting in increased number of population and food. This system has a distance coverage of 30 miles. [41]

The author proposed a system that uses IoT and sensors to analyze and identify the infected hen. This reduces the cost of labor and increase the accuracy of the identification process. In this paper we discuss about the overall system, audio and video analysis methods and comparing the results using MATLAB. The process of sick identification has been optimized using the MATLAB results. First is using the image analysis including the RGB images (SEN-11745 res 728X488 a RGB camera, with the lens angle 70°) and thermal images (Far Infrared Thermal Sensor Array –AMG8833 Grid-EYE 8X8 RES from Panasonic). AMG8833 can measure temperatures ranging from 0 °C to 80 °C (32 °F to 176 °F) with an accuracy of ±2.5 °C (4.5 °F) and the maximum distance is 7 m. The second (SEN-14262 a Microphone controller) is using the sound of the hen. In this paper we are describing the analysis methods we have done for identifying the infected hen and the performance of the various audio feature extraction and classification methods used for analysis the sick hen. The author fund that MFCC audio feature extraction along with the KNN medium has better prediction for identifying the sick hen. Image and audio analysis together produce accurate results. The signature thump of sound for various diseases is important point of research could be carried out.[42]

Temperature and Humidity Monitoring System architecture, it contains the hardware and software that are used during development. DHT11 sensor is used to detect temperature and humidity [43]. DHT11 is a very low price and low power consumption sensor that has basic capability data logging, converting from analog to digital. It is good at 0-50ºC temperature reading with ±2ºC accuracy and 20-80% humidity readings with 5% accuracy. Such measurement is suitable enough for any chicken cage in Indonesia.

Arduino DUE is used to capture the sensor data from DHT11. Arduino DUE is a powerful 84 MHz and 32-bit processor microcontroller. It has Direct Memory Access (DMA) feature that is able to support multi-tasking within the microcontroller. USB OTG capability is also a benefit of this Arduino to enable Wi-Fi module connection. Low-cost Wi-Fi module ESP8266 is added to the prototype to support internet connection. Using such connection, the data that has been processed in Arduino will be transmitted to the IoT platform. Blynk app is used as a platform to control and monitor the deployed device. This platform is also very useful to connect the device to the Blynk cloud server and analyses the telemetry data over Wi-Fi or any internet connection. It has several libraries to support data analysis. Another pros is Blynk cloud server is open-source and deployable in minutes. See Figure 2. The sensor is then installed in the chicken cage with some broilers inside. The cage area is 80 x 11 m. Sensor is placed every 15 m in one side of the cage. The monitoring prototype is deployed and obtaining data from the sensor that is connected using Wi-Fi.

The temperature and humidity that are reported in Blynk app. This app makes user able to acknowledge the status of the chicken cage from time to time and from anywhere. Depicts the web-based monitoring system that has been deployed. Such website is able to show the status within the cage and will be used later during controlling the humidity and temperature. [44]

Poultry tracking is primarily used for evaluating abnormal behavior and predicting disease in poultry. Offline video is often used to track and record poultry behavior. However, poultry are group-housed animals. The difficulty of accurately monitoring large-scale poultry farms lies in the automatic tracking of individual poultry. To this end, this paper demonstrates the use of a deep regression network to track single poultry based on computer vision technology. By referring to the Alexnet network, the broiler chicken area of the previous frame and the search area of the next frame were input into the convolutional layer respectively, and the coordinates of the prediction area were obtained by full-connection layer regression. The method was compared with some existing tracking algorithms. Preliminary tests revealed that when compared with MeanShift Algorithm (MS), Multitask learning Algorithm (MIL), Kernel Correlation Filter (KCF), Adaptive Correlation Filters (ACF) and tracking-learn-detection (TLD), the poultry tracking algorithm named TBroiler tracker proposed in this paper has better performance on the overlap ratio, pixel error and the failure rate. TBroiler achieved a mixed tracking performance evaluation (MTPE) of 0.730. The evaluation scores of other methods were 0.362 (MS), 0.355 (MIL), 0.434 (KCF), 0.051 (ACF), and 0.248 (TLD). In addition, the method can be further optimised to improve the overall success rate of verification. [45]

Chicken removal system was developed for removing dead chickens from poultry houses. Two modes are designed for the small chicken removal system. One is remote control; the designed system is connected to the user equipment through WiFi to enable breeders to control the system via the human–machine interface from anywhere. Thus, breeders need not enter poultry houses to remove dead chickens manually. The other is the automation mode, the designed system can operate automatically without human intervention and perform tasks, such as navigation and deep learning for object detection. [46]

Murad et al. developed a monitoring system based on a wireless sensor network (WSN) for poultry farms. Their system comprises Crossbow's TelosB motes integrated with commercial sensors that can measure temperature and humidity. The data collected by the sensors are uploaded to an online database to enable managers to obtain information from the online monitoring solution provided by the system. [47]

Mirzaee-Ghaleh et al. Monitored and maintained four indoor climate parameters, namely the temperature, humidity, CO2 concentration, andNH3 concentration, by using three self-developed fuzzy logic controllers. [48]

Zhang et al. Proposed a system based on a WSN for controlling the environmental parameters in buildings housing livestock. This system allows a breeder to monitor the temperature, humidity, lighting level, CO_2 concentration, NH_3 concentration, and H_2S concentration in buildings housing livestock in real time. In addition, the aforementioned method reduces labor costs and energy consumption. [49]

Astill et al. Investigated areas of the poultry industry affected by smart sensor technologies. They also described how sensor technology is related to big data analytics and Internet of Things systems. This technology can increase the output of the poultry industry. Thus, the introduction of technology to the poultry industry can increase the output and convenience of this industry. [50]

Proposed the fault diagnosis of electric impact drills using thermal imaging, the feature extraction of thermal images using Binarized Common Areas of Image Difference (BCAoID). The recognition results were 97.91–100%. Fault diagnosis based on thermal images can protect rotating machinery and engines. In our study, distances between the chicken removal system and the measurement equipment are a large issue. However, thermal cameras have distance constraints when measuring the target. To obtain an accurate temperature, the distance between the thermal camera and target should not be too far. Therefore, the visible light camera was adopted for the chicken removal system in this study. [51]

Zhu et al. Adopted a Support Vector Machine (SVM) algorithm for automatic dead chicken detection in modern chicken farms. The experiments showed that the detection accuracy was over90%. [52]

Jake et al. Reviewed the research on detecting and predicting emerging diseases in poultry with the implementation of new technologies and big data. Avian influenza virus was the focus. [53]

CHAPTER THREE

METHODOLOGY

3.1 Introduction:

This chapter shows the methodology of the project. Section 3.2 show an overview of the system. Section 3.3 Simulation. Section 3.4 Control part. Section 3.5 Image processing part.

3.2 System overview:

The system consists of two major parts, first part for monitoring the environment, started first by collecting the data of the suitable environment for the chickens to grow in a healthy way. To monitor the farm environment, DHT11 sensor is used to measure the temperature and humidity, second LDR to measure light intensity. All the data collected by the sensors will be processed by Raspberry Pi controller, the system has three output devices, first an LCD and it shows when the temperature or humidity and the farmer can know if it isn't suitable for chickens to grow healthy, second a LED that opens according to the light intensity third fan that opens according to temperature. The Node-Red software has been used for hardware implementation, The farmer can supervise the status of the farm through his smart device that presents the lighting level, temperature, and humidity. Also, it allows controlling ventilation as well as the lightening of the farm.

Second part for image processing to detect the ill chickens, Raspberry Pi camera module V2 used for recognition and detecting of the chickens' features. Here used machine learning (supervised learning) to train the system by YOLO V5.

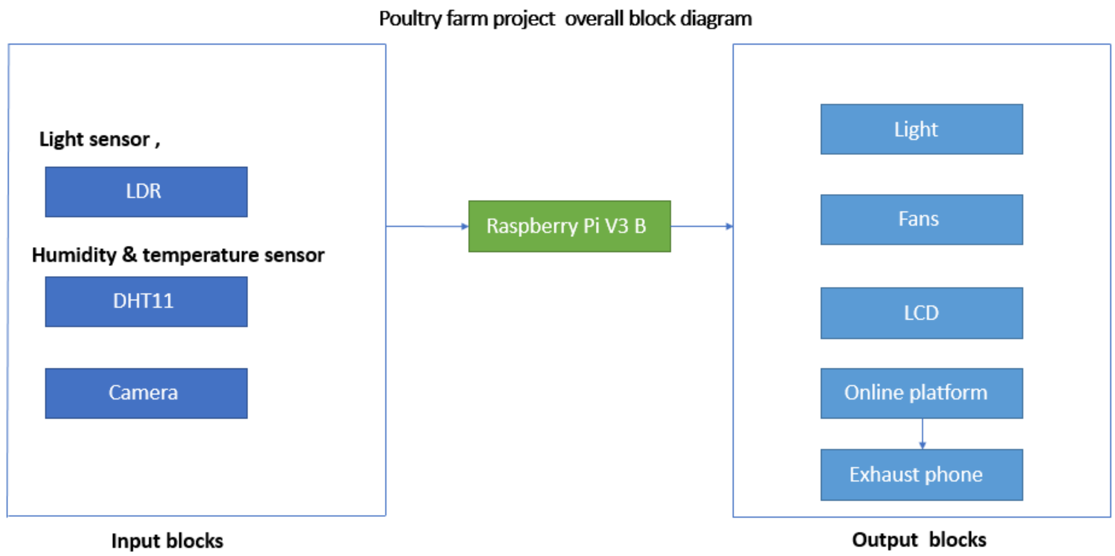

Figure 3-1: System block diagram

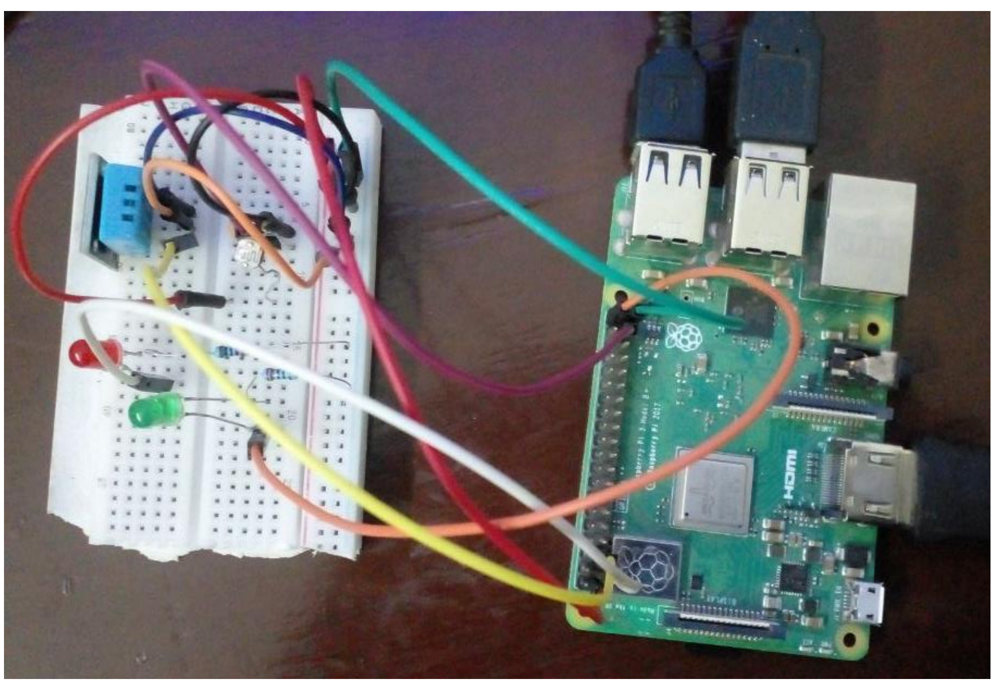

Figure 3-2: control part hardware implementation circuit

The figure shows hardware implementation circuit, the sensor DHT11 is connected to General-Purpose Input/output (GPIO)3 -pin No.5-, the sensor LDR is connected to GPIO26 -pin No.37-, and a one micro farad capacitor is connected to LDR sensor, the red led represent the fan output and is connected to GPIO2 -pin No.3-, the green led represent the lamp output and is connected to GPIO21 -pin No.40-.

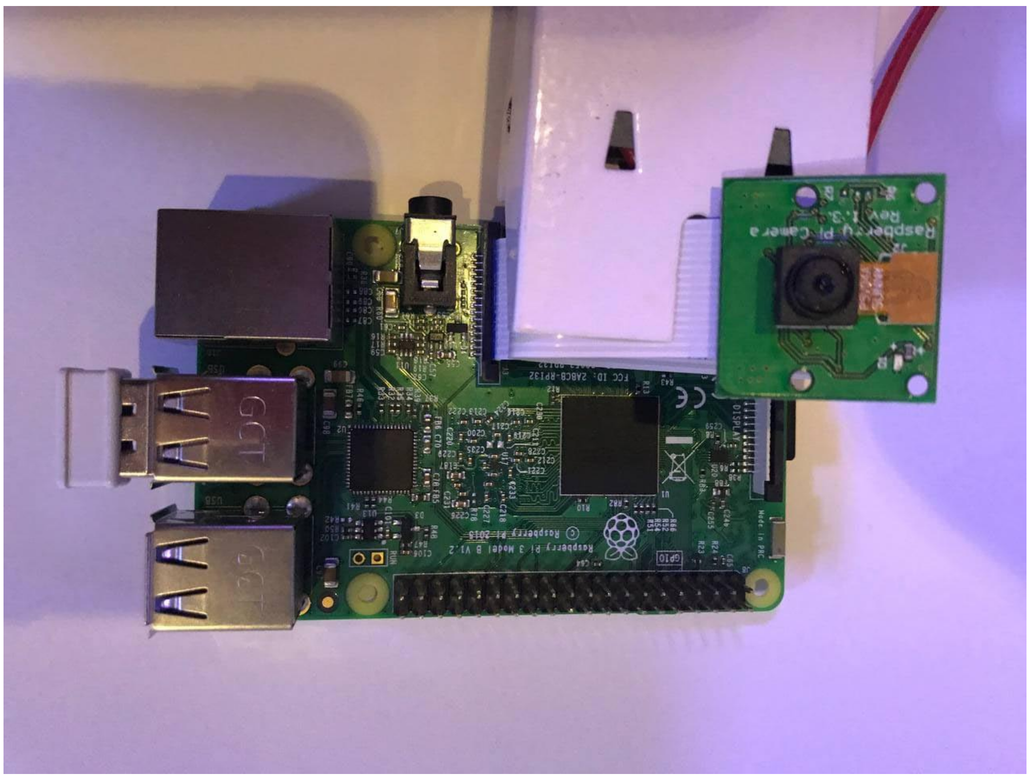

Figure 3-3: Image processing part implementation hardware circuit

The figure shows hardware implementation circuit, the raspberry pi camera module v2 is connected to raspberry pi version3 B+.

A website (poultry961196958.wordpress.com) with a friendly interface, has been used make using the system comfortable for farmers. They can just choose the part they need to monitor. The control part showed the sensors level as well as the switches to control the lighting and ventilation. In the diseases monitoring part they can see the results of detection of ill chickens. By running code locally by detect.py code downloaded with YOLO V5 package, and change the source to video stream can get real-time detect. Next, pass the ID address to the web-side to have online live stream detect features.

3.3 Simulation:

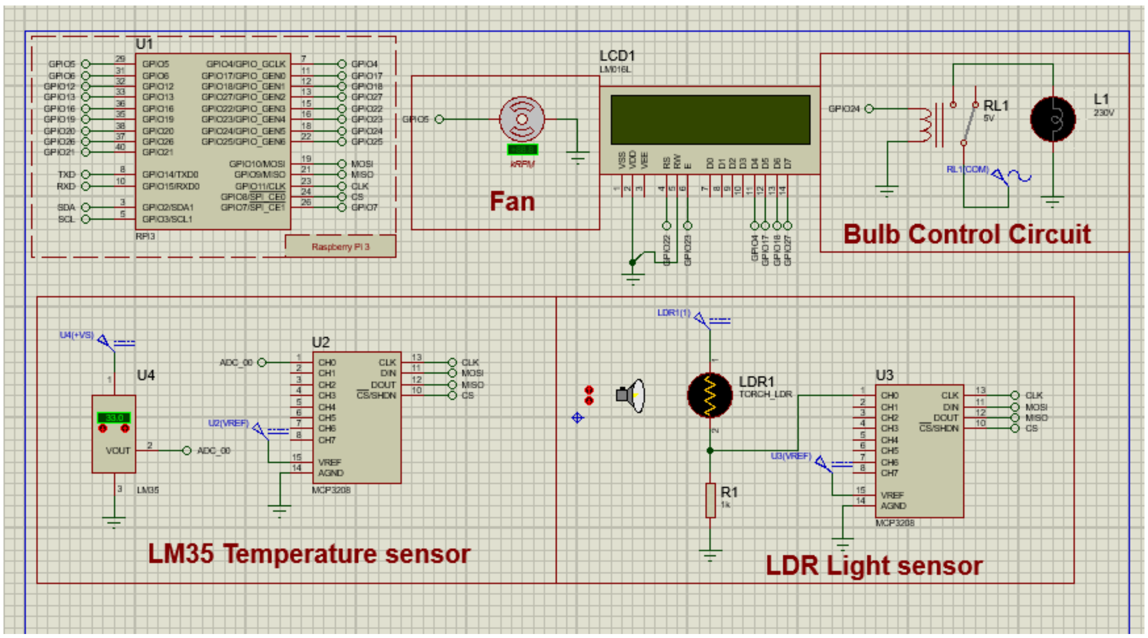

Figure 3-4: Simulation

By using a Raspberry Pi V3 B, LM35, LDR, Bulb, LCD, Relay, two MCP3208 drivers and a resistor. First LM35 used to measure temperature and is connected with the driver, the collected data (temperature) from LM35 has been shown on LCD and the fan turned on when the temperature was more than 35°C and off when less. Second LDR used to measure light intensity and is connected with a driver and resistor, if the light intensity is low the Bulb will be on.

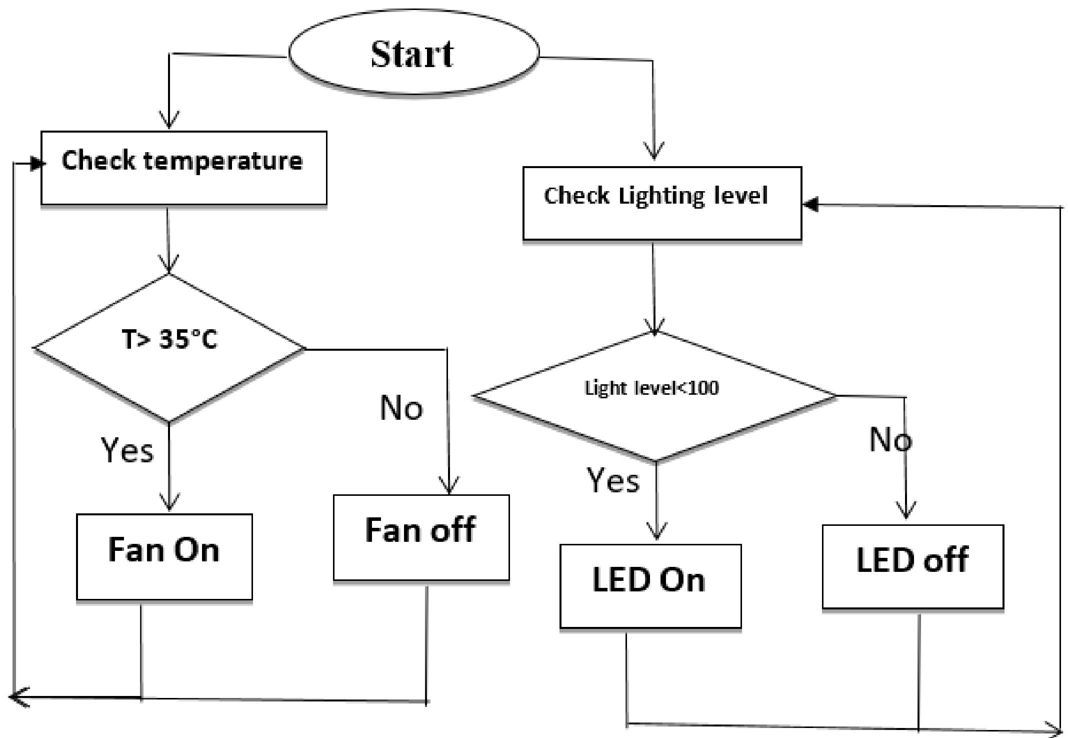

Figure 3-5: Simulation flow chart

The flowchart shown presents the process of the simulation code. First the temperature part, first step is to check if it's more or less than 35 C , if more the fans will work otherwise it will stop. Second the light level, if it's more than 100 the lights will be Off otherwise it will be On . That's all happened according to the program written using Python language on a Raspberry Pi controller.

3.4 Control part:

3.4.1 Raspberry pi model B+ software installation:

The Raspberry Pi operating system installation steps:

1. Go to (raspberrypi.com/software/operating-systems) and download the OS with desktop and recommended software.
2. Using Raspberry pi Imager to write the OS to Raspberry pi card (SanDisk class 10, 32 GB has been used).
3. Write the command "sudo raspi-config" on the terminal, then will open raspberry pi software configration tool.
4. Enable interface options (SPI, I2C, Serial, 1-wire, and Remote GPIO).

3.4.2 Node-Red installation:

Node-Red installation steps:

1. Go to (https://nodered.org/docs/getting-started/raspberrypi) and the copy the command
   "bash < (curl -sL https://raw.githubusercontent.com/node-red/linux-installers/master/deb/update-nodejs-and-nodered)" and then paste it on the terminal to download Node-Red.
2. To let the Node-Red to auto start and boot write the command
   "sudo systemctl enable nodered.service" on the terminal.
3. Open Node-Red on the browser by typing the URL "http//localhost:1880".

3.4.3 Install nodes in Node-Red:

Steps for downloading nodes in Node-Red:

1. Download bcm2835 library from http://airspayce.com/mikem/bcm2835.
2. Open the downloaded library folder in the terminal after extracting it.
3. Enter the following commands in the terminal one by one:
   ". /configure"

"Make"

"sudo make check"

"sudo make install"

"sudo npm install --unsafe-perm -g node-dht-sensor"

4. In Node-Red manage palette search for (DHT11, HX711, and Gauge node) and install them.

3.4.4 Digital Temperature and Humidity sensor (DHT11) flow:

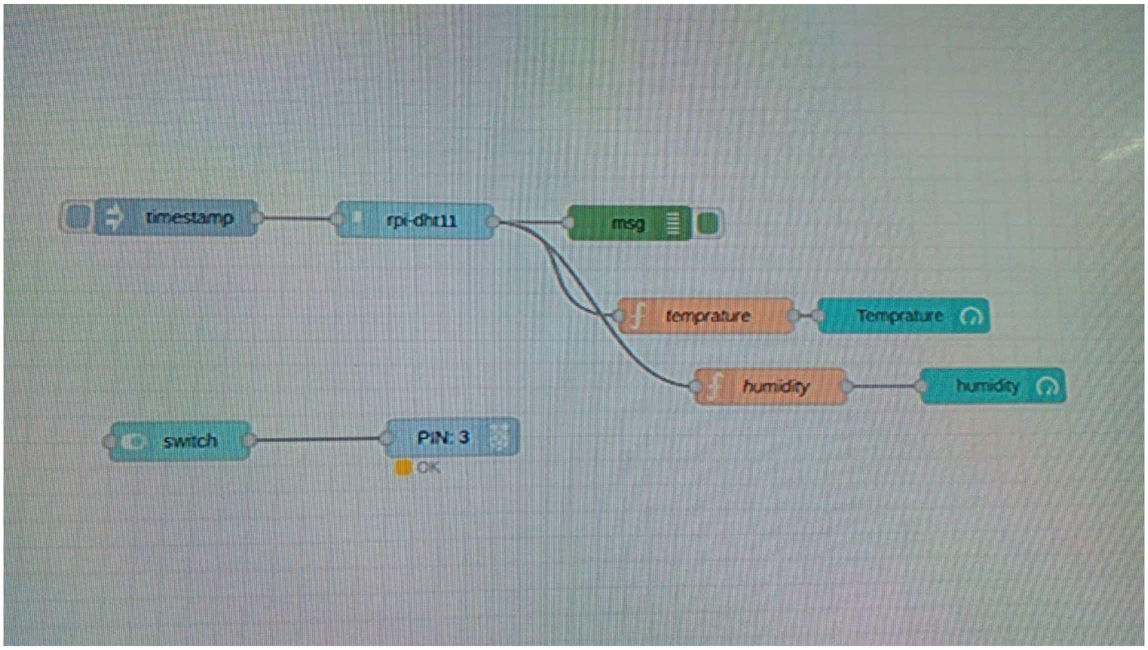

Figure 3-6: Digital Temperature and Humidity sensor (DHT11) flow

The flow starts from Inject(timestamp): The Inject node can be used to manual trigger a flow by clicking the node's button within the editor. It can also be used to automatically trigger flows at regular intervals. The message sent by the Inject node can have its payload and topic properties set. [54]

Then the message goes to the next node DHT11(rpi-dht11): This is a Node Red node to manage connection to a DHT11 sensor on a Raspberry Pi. It allows you to specify the variables that define the connections to the sensor. [55] By using GPIO3 in pin no.5 for DHT11 signal, 3.3v pin no.1 to for +pin of DHT11, and GND pin no.9 for -pin of DHT11.

The next node Debug(msg): The Debug node can be used to display messages (the option of display complete message) in the Debug sidebar within the editor, the sidebar provides a structured view of the messages it is sent, making it easier to explore the message, alongside each message, the debug sidebar includes information about the time the message was received and which Debug node sent it. Clicking on the source node ID will reveal that node within the workspace, the button on the node can be used to enable or disable its output. It is recommended to disable or remove any debug nodes that are not being used, the node can also be configured to send all messages to the runtime log, or to send short (32 characters) to the status text under the debug node. [54]

Function node: The Function node allows JavaScript code to be run against the messages that are passed through it, the first is for temperature and its function code:

```
msg.payload = msg.payload;
return msg;
```

The second function is for humidity and its function code:

```
msg.payload = msg.humidity;
return msg;
```

Two-gauge nodes is for displaying the temperature and humidity values on a web page.

Switch node: the switch node allows messages to be routed to different branches of a flow by evaluating a set of rules against each message, the switch node allows messages to be routed to different branches of a flow by evaluating a set of rules against each message. [54] It has been used to operate the fan, the switch that has been use is dashboard switch.

The last node is pin no.3, it's (rpi-gpio out) node that represents the fan, it controlled by the dashboard switch using the webpage.

3.4.5 Light Dependent Resistor sensor (LDR) flow:

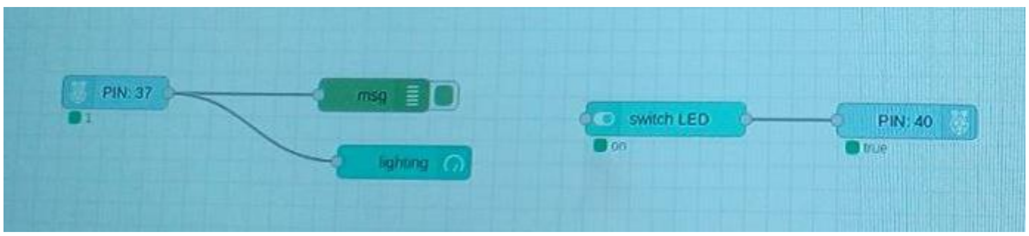

Figure 3-7: Light Dependent Resistor sensor (LDR) flow

By using GPIO26 in pin no.37 for LDR signal, the (rpi-gpio in) node has been used as an input node of the light sensor. The debug node displays the message of the current light intensity, and the gauge node display it on webpage. The dashboard switch node used to allow operating of LED in GPIO21 in the pin no.40 from webpage.

3.5 Image processing part:

3.5.1 Data-set:

The data are composed of 1 class with a total of 708 images size 128x128, which were downloaded from images.cv website. the data folder contains 3 sub-folders: train, test, and val. The format of all the images is .jpg. For better model performance, the website provides to increased using data augmentation techniques: Rotate, Noise, Flip, and Brightness, so that the total number of images is increased by 30%, our steps:

1- Downloaded dataset from images.cv
https://images.cv/dataset/chicken-image-classification-dataset

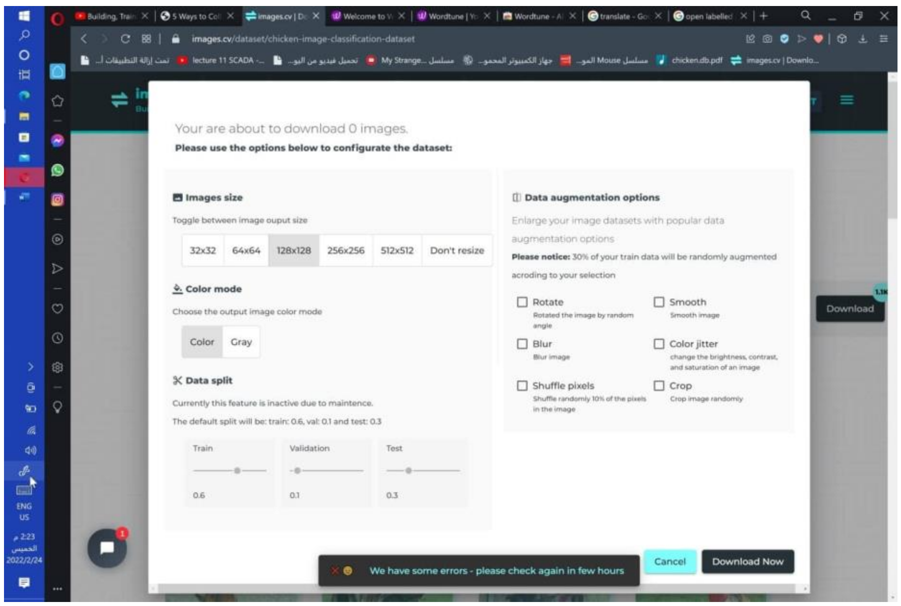

Figure 3-8: Choosing images size

2- Labelled data by used make sens.ai
https://www.makesense.ai

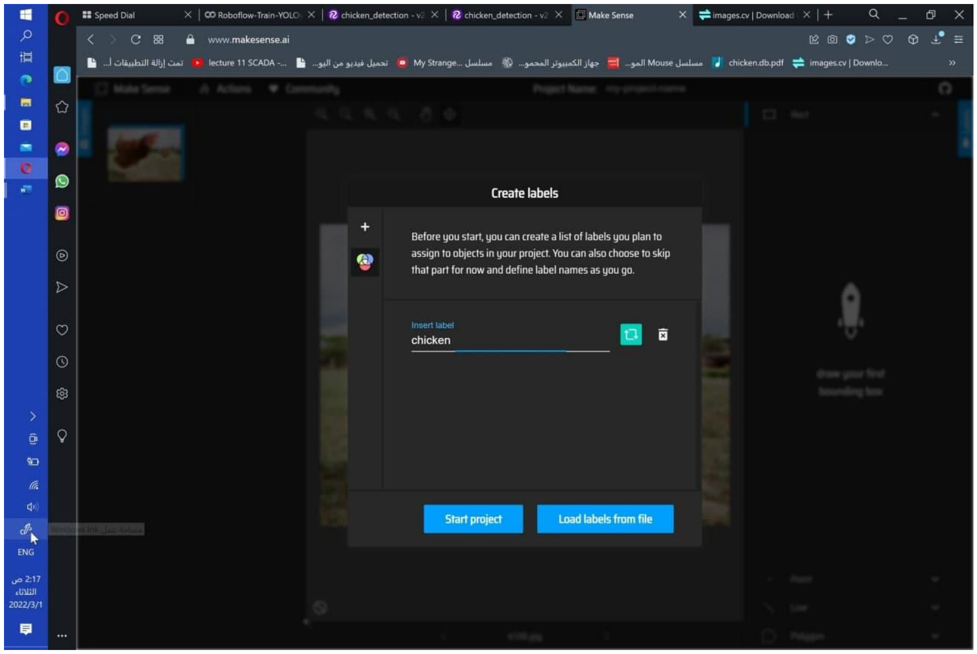

Figure 3-9: Create a list of labels

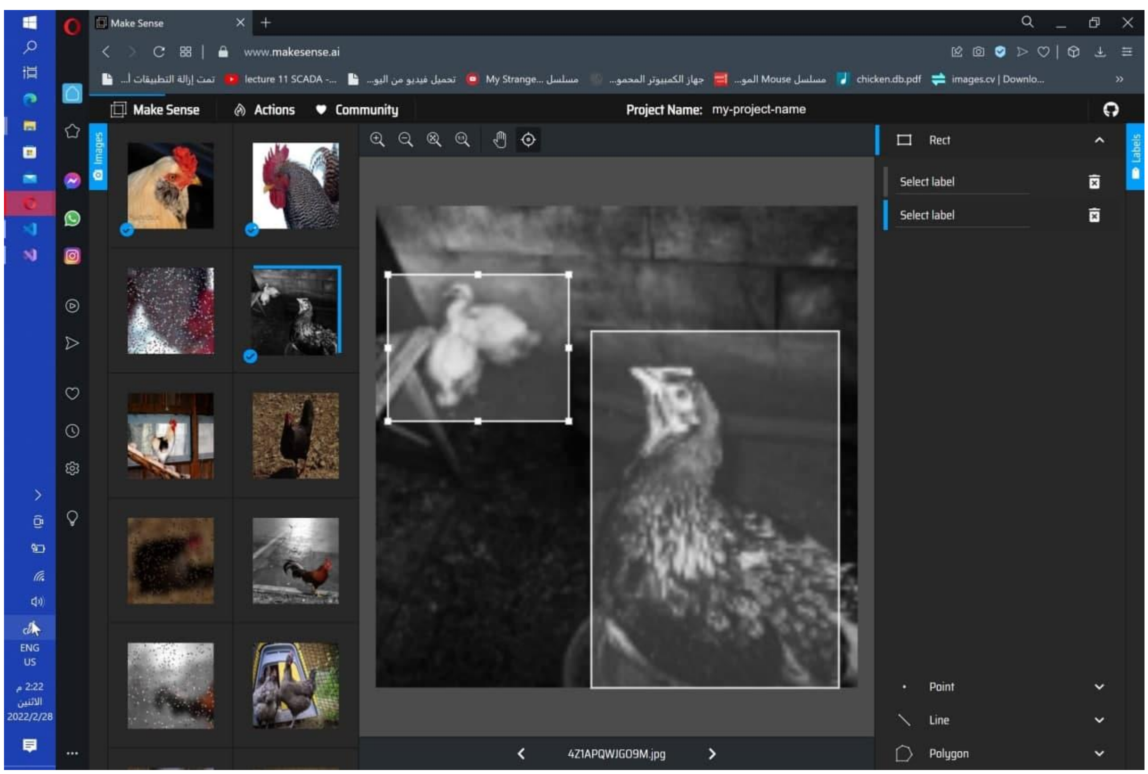

Figure 3-10: Creating labels for images

3- Export annotations to a zip package containing files in yolo format

4- load dataset after labelled in robo-flow, the robo-flow provide training for machine learning data, preparing data, split data to three categories train, test and evaluation, it also provides pre-processing and augmentation

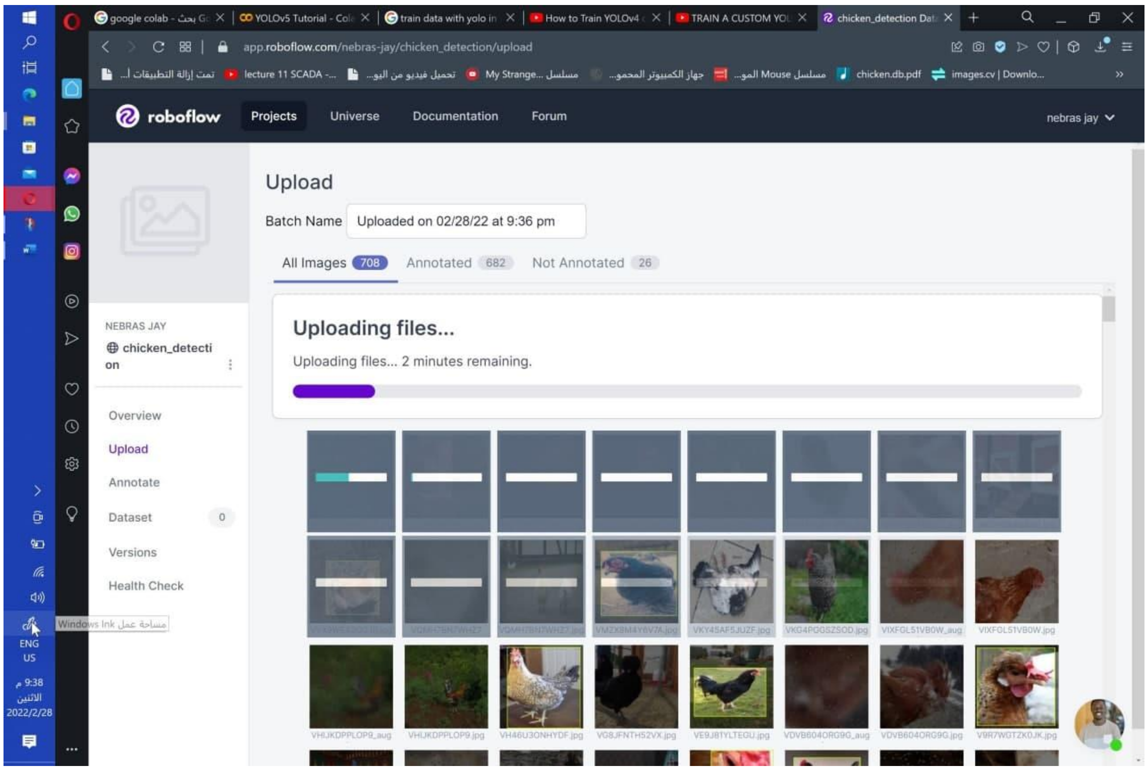

Figure 3-11: Uploading images to Robo-flow

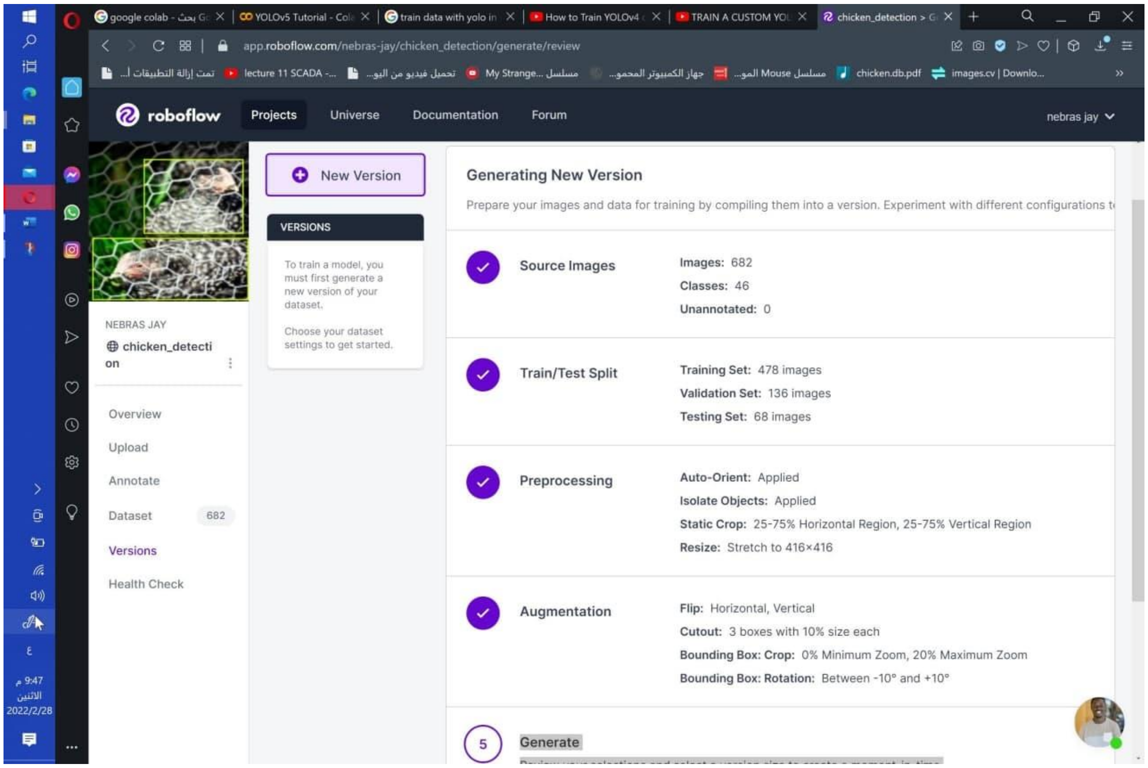

Figure 3-12: Preparing images for training

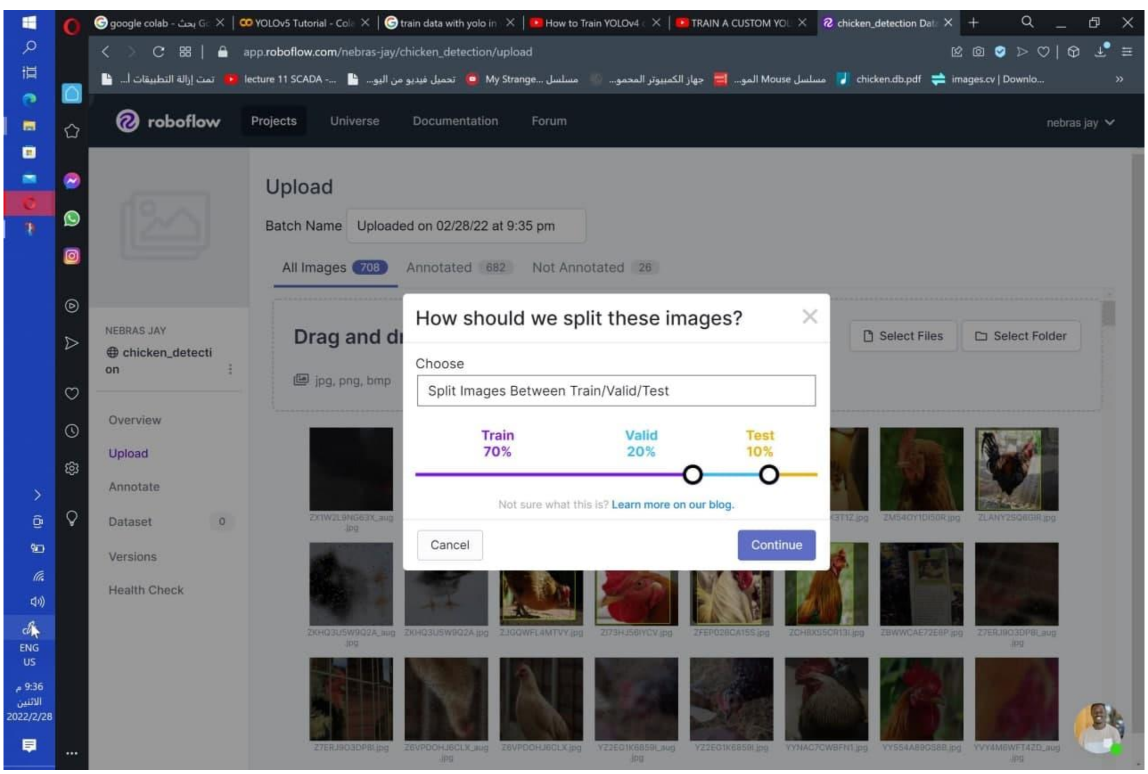

Figure 3-13: Split images to categories

5- for training module, yolo v5 has been used in google colap, google colap provide Jupyter Notebook feature, free GPU and inter-active code editor

https://colab.research.google.com/github/roboflow-ai/yolov5-custom-training-tutorial/blob/main/yolov5-custom-training.ipynb

3.5.2 YOLO V5 and training:

Next steps show the preparing for the suitable environment for YOLO V5 to work, how to train the YOLO V5 model to recognize objects:

1- Install Dependencies:

```
# 1-1 prepare environment by install requirements

#Base----------------------------------------
matplotlib>=3.2.2
numpy>=1.18.5
opencv-python>=4.1.2
Pillow>=7.1.2
PyYAML>=5.3.1
requests>=2.23.0
scipy>=1.4.1
torch>=1.7.0
torchvision>=0.8.1
tqdm>=4.41.0
```

2- Assemble Our Dataset:

In order to train the model, assembling the dataset of representative images with bounding box annotations around the objects that wanted to be detected was needed also dataset to be in YOLO V5 format was needed. Which provided by Robo-flow.

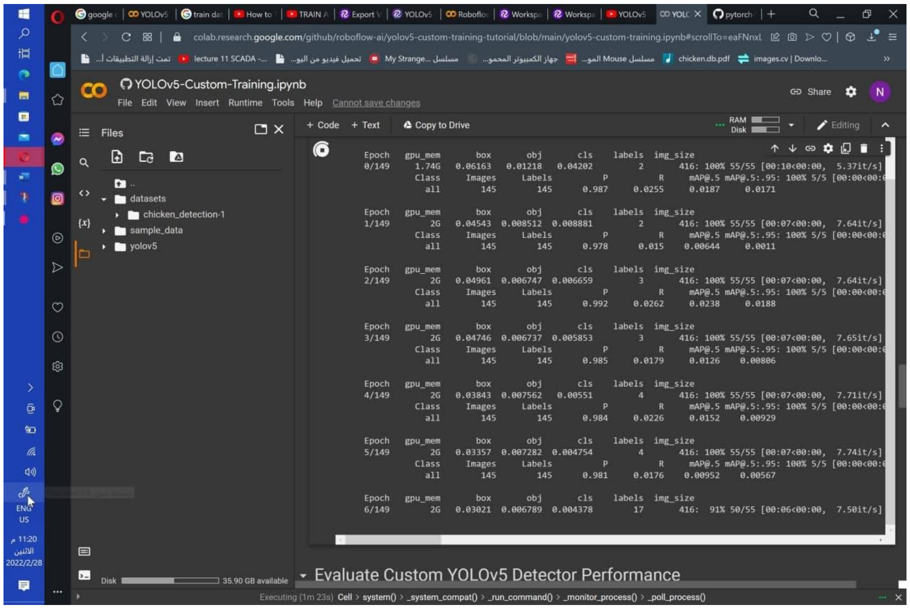

Figure 3-14: Training the model

3- Train YOLOv5 model:
Here, it was able to pass a number of arguments:
- img: define input image size
- batch: determine batch size
- epochs: define the number of training epochs. (Note: often, 3000+ are common here!)
- data: Our dataset location is saved in the dataset.location
- weights: specify a path to weights to start transfer learning from.
- cache: cache images for faster training

4- Evaluate Custom YOLOv5 Detector Performance:
Training losses and performance metrics are saved to the Tensor board and also to a logfile.

5- Run Inference with Trained Weights
Run inference with a pretrained checkpoint on contents of test/images folder downloaded from Robo-flow.

6- Conclusion and Next Steps:

YOLOv5 model has been trained to recognize the objects.

3.5.3 PyTorch and Visual studio:

Steps for preparing the environment to implement detection:

1- installed and clone the Yolov5:
git clone https://github.com/ultralytics/yolov5.git

2- install a python virtual environment
sudo apt-get install python3-venv
To create a virtual environment now use the command:
python3 -m venv <GRP_environment>
Where "<GRP_environment>" is the name of the environment that we change to our liking.

3- activate our environment:
source your_environment/bin/activate

4- Visual Studio Code Editor has been used to create python environment to figure out how to install a Virtual environment in Visual Studio code, you can use the following tutorial:
https://youtu.be/Wuuiga0wKdQ

The reason for activating a python virtual environment is so to prevent dependencies from clashing. You might need different versions of the same library like Numpy for example for different projects, so you create virtual environments for each project and install all the required libraries for a project in that virtual environment.

5- After getting into the cloned YOLO v5 repository, because of using the windows operating system, edit the requirements.txt file in the YOLO v5 folder and replace the line "pycocotools" with the line "pycocotools -windows"

6- install the dependencies needed for yolov5
pip install -U -r requirements.txt

7- install PyTorch and made sure that it had GPU and it has CUDA support.

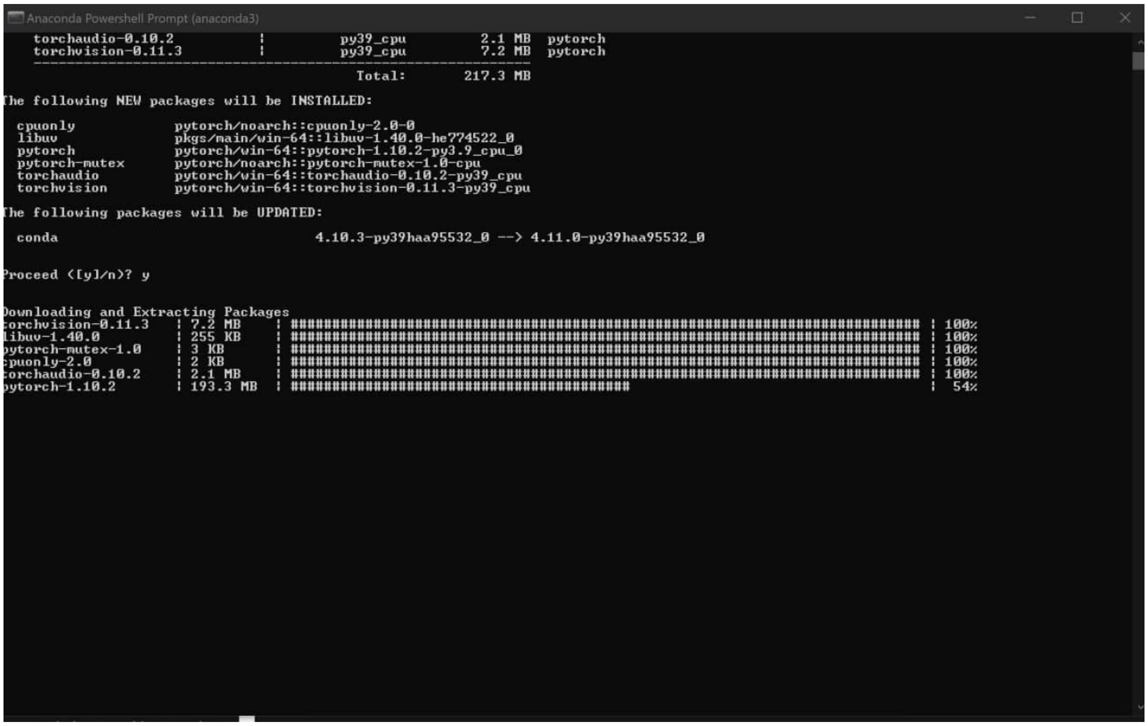

Figure 3-15: Installing PyTorch and Anaconda

First, install Pytorch and set up a virtual Python environment inside Windows, using anaconda as a package manager.

Detection using YOLO v5:

After getting our training model file from chicken detection YOLO v5 on google colap, run this command in terminal:

python detect.py — source./inference/images/ — weights yolov5s.pt — conf 0.4

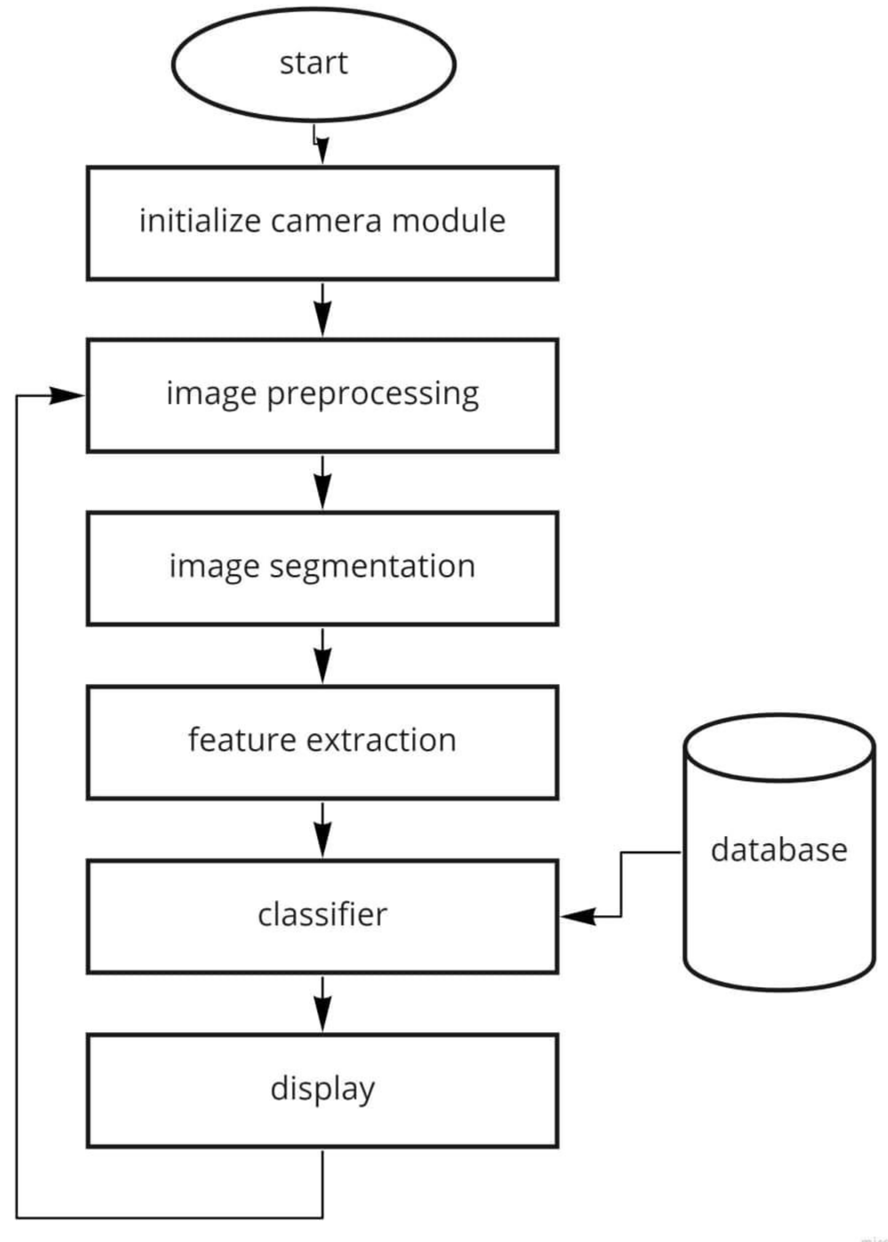

Figure 3-16: Image processing flow chart

CHAPTER FOUR

RESULTS AND DISCUSSION

4.1 Introduction:

This chapter shows the results of the project, section 4.2 shows the results of control part, section 4.3 shows the results of image processing part.

4.2 Control part:

4.2.1 Digital Temperature and Humidity sensor (DHT11) results:

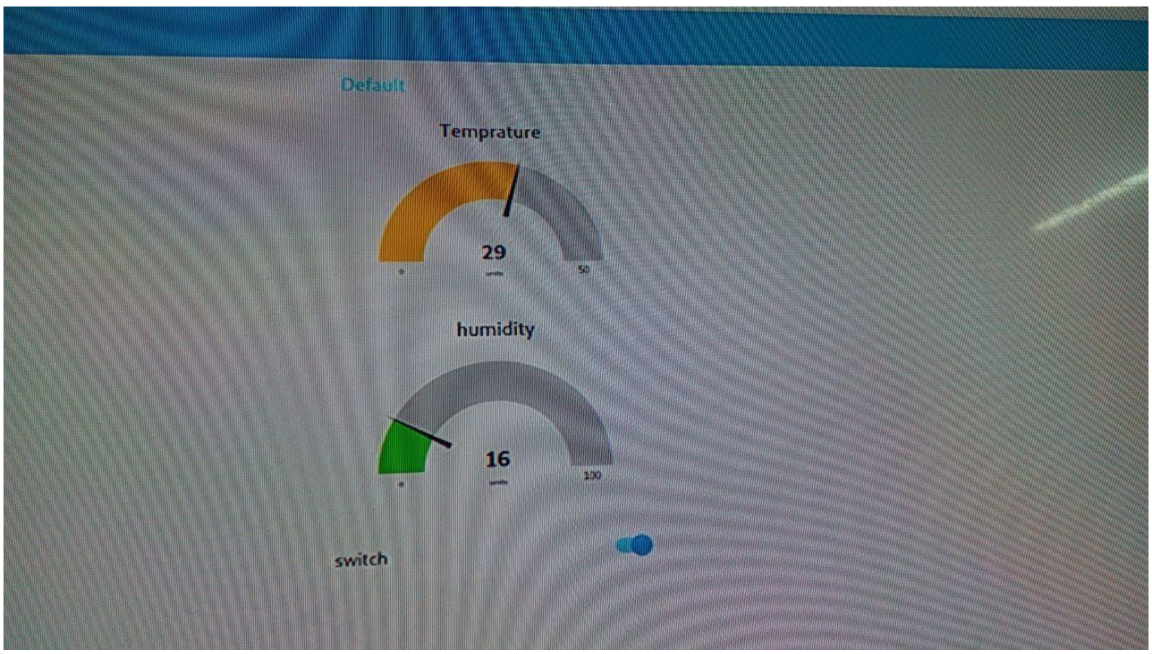

Figure 4-1: Screen shot of Digital Temperature and Humidity sensor (DHT11) readings on Node-Red dashboard

After designing the DHT11 circuit and implementation on Node-Red as a flow, then deployed the system and it gave temperature and humidity readings. The farmer can see the temperature and humidity readings from any place using dashboard webpage link, and can use computer or mobile to see the dashboard webpage, then the farmer can open or close the fan using dashboard switch.

4.2.2 Light Dependent Resistor (LDR) results:

Figure 4-2: Screen shot of Light Dependent Resistor (LDR) readings on Node-Red dashboard

After designing the LDR circuit and implementation on Node-Red as a flow, then deployed the system and it gave light intensity readings. The farmer can see the light intensity readings from any place using dashboard link, and can use computer or mobile to see the dashboard, then the farmer can open or close the lamp using dashboard switch.

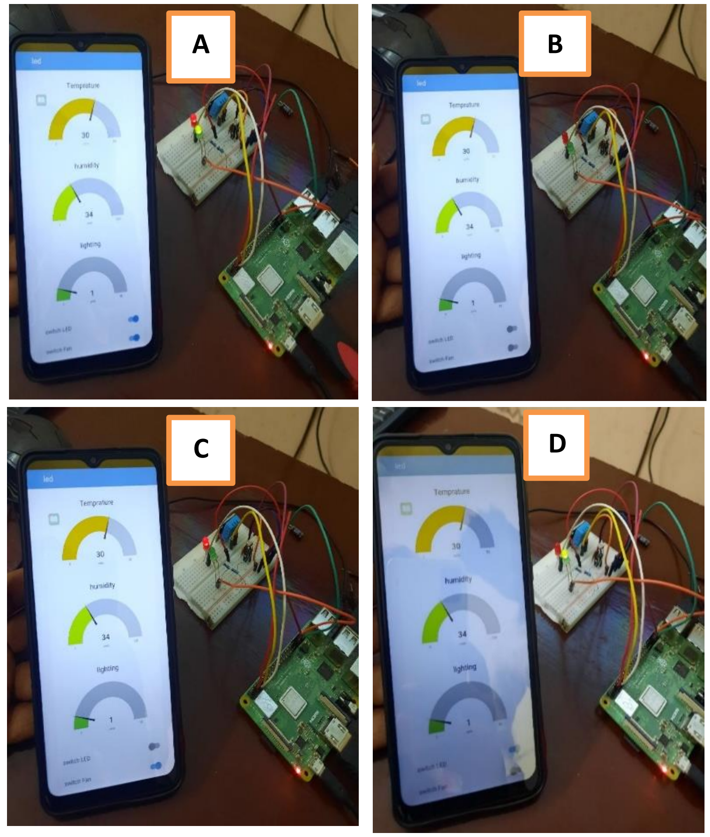

Figure 4-3: overall results for control circuit

The figure shows the display of dashboard of gauge nodes on webpage of Node-Red on smartphone, with all of the readings of temperature, humidity and lighting level as well as fan and lamp dashboard switches. To show the results a red LED instead of a fan has been used, and green LED as a lamp. in Figure 4-3 (A) both fan and lamp are ON, Figure 4-3 (B) both fan and lamp are OFF, Figure 4-3 (C) fan is ON and lamp is OFF, Figure 4-3 (D) fan is OFF and lamp is ON.

4.3 Image processing part:

It is clear from Figure 4- that the model can almost detect the class with a medium prediction value. The accuracy curves for precision, recall, and mAP_0.5 with confidence value and F1 score are plotted in Figure 4-4.

Figure 4-4: Plots of the precision, recall, Mean Average Precision (mAP) -0.5- parameters along with class object loss for training epochs

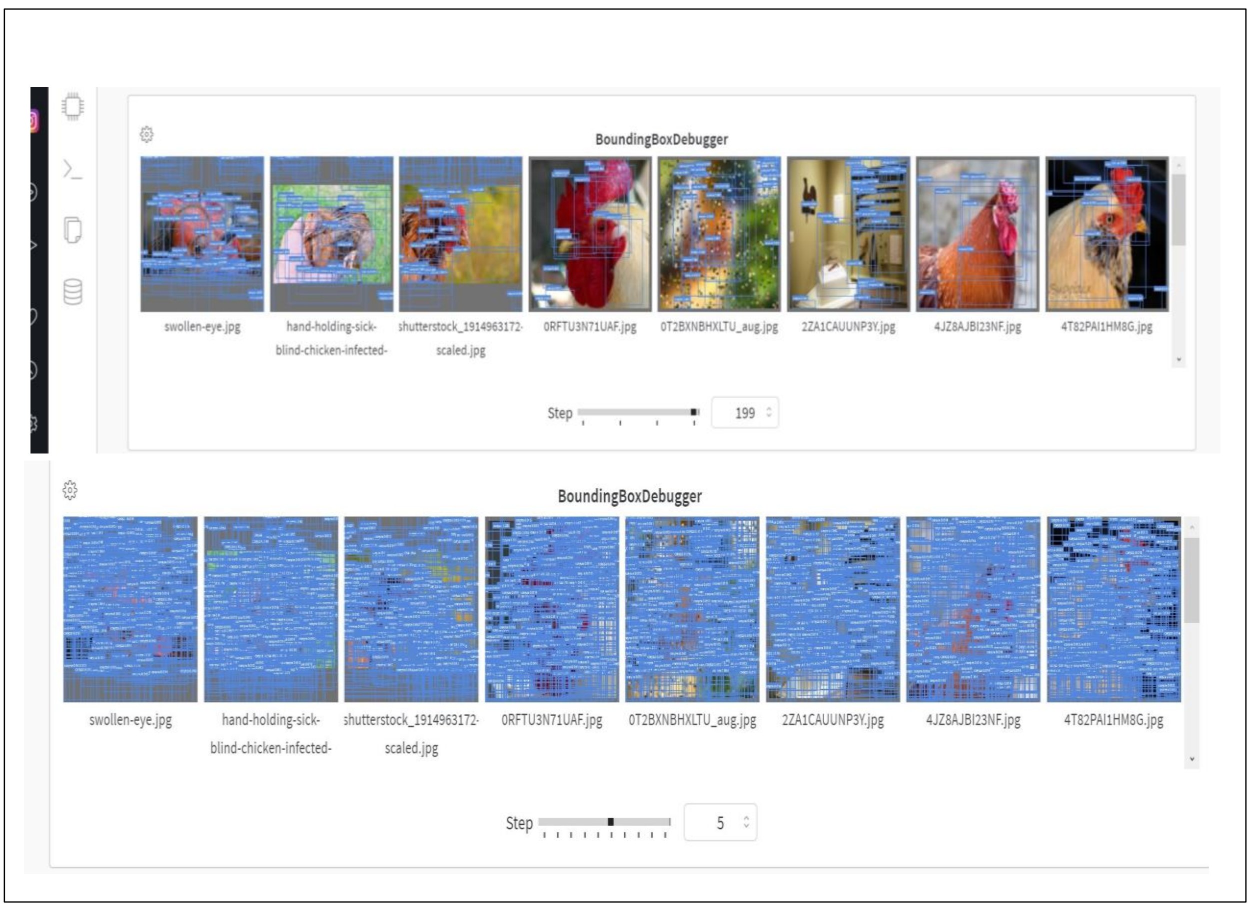

Figure 4-5: The improvement of model by changed the epochs & cache

The graphs in Figure 4-5 show the improvement in the model by displaying different performance metrics for both the training and validation sets. In this model, has been used early stopping to select the best weights. The presented model shows improved precision, recall, and mAP until reaching a peak at 100, 10, 500 and 1000 epochs and 16,3,100 and 300 caches, respectively. The validation data's classification loss also showed a rapid decline after epoch 4. The loss function demonstrates how well a particular predictor performs in identifying the input data elements in a dataset. The lower the loss, the better the classifier models the relationship between input data and output targets. It displays how effectively the algorithm predicts the proper class of a given item in the situation of classification loss. Figure 4- shows the results of those metrics for all classes, obtained on the first dataset with mode YOLO v5s.

Name (12 visua	batch_size	cache	epochs	best/mAP_(	best/recall	best/precisi	metrics/pre	metrics/rec	metrics/mA	best/mAP_(	metrics/mA
T12	300	ram	1000	0.3823	0.3333	0.99	0.9987	0.3333	0.3421	0.2058	0.1351
T11	250	ram	800	0.3356	0.3333	0.9942	0.9934	0.3333	0.3356	0.1677	0.1677
T10	300	ram	500	0.5555	0.6667	0.6647	0.6663	0.666	0.5555	0.2902	0.2902
T9	100	ram	500	0.3351	0.3333	0.9972	0.9974	0.3333	0.3351	0.3016	0.3016
T8	20	ram	500	0.3764	0.3333	0.9986	0.9986	0.3333	0.3764	0.2846	0.2846
T7	3	ram	10	0.0247	0.3333	0.03883	0.03894	0.3333	0.02489	0.01258	0.01266
T6	16	ram	90	0.3397	0.3333	0.6772	0.7362	0.3333	0.3398	0.1019	0.1019
T5	16	ram	150	0.3352	0.3333	0.9902	0.9903	0.3333	0.3352	0.2682	0.2682
T4	16	ram	200	0.995	1	0.9989	0.9989	1	0.995	0.5459	0.5459
T3	16	ram	100	0.995	0.9893	1	1	0.9893	0.995	0.547	0.547
T2	16	ram	50	0.995	1	0.8147	0.8422	1	0.995	0.3971	0.3971
T1	16	ram	3	0.005476	0.5	0.007668	0.007646	0.5	0.00546	0.001095	0.001092
fiery-flow	16	ram	3	-	-	-	-	-	-	-	
lunar-mu	16	ram	3	-	-	-	-	-	-	-	-

1-15 of 15

Figure 4-6: Performance of the model YOLOv5s for custom Robo-flow data

Figure4-6 shows the results for each of the 11 training sets and the entire validation set. The number of known targets to be detected is shown in the third column. The detector's accuracy and recall are shown in the fourth and fifth columns. Finally, the sixth column displays the mean average accuracy for the given intersection over the union. Finally, our results showed that the presented approach can be used to investigate or identify Infectious Coryza disease.

The best result of detecting infectious coryza is shown in Figure 4-7, had parameters (100 batch and 800 epochs trained on the yolov5s model).

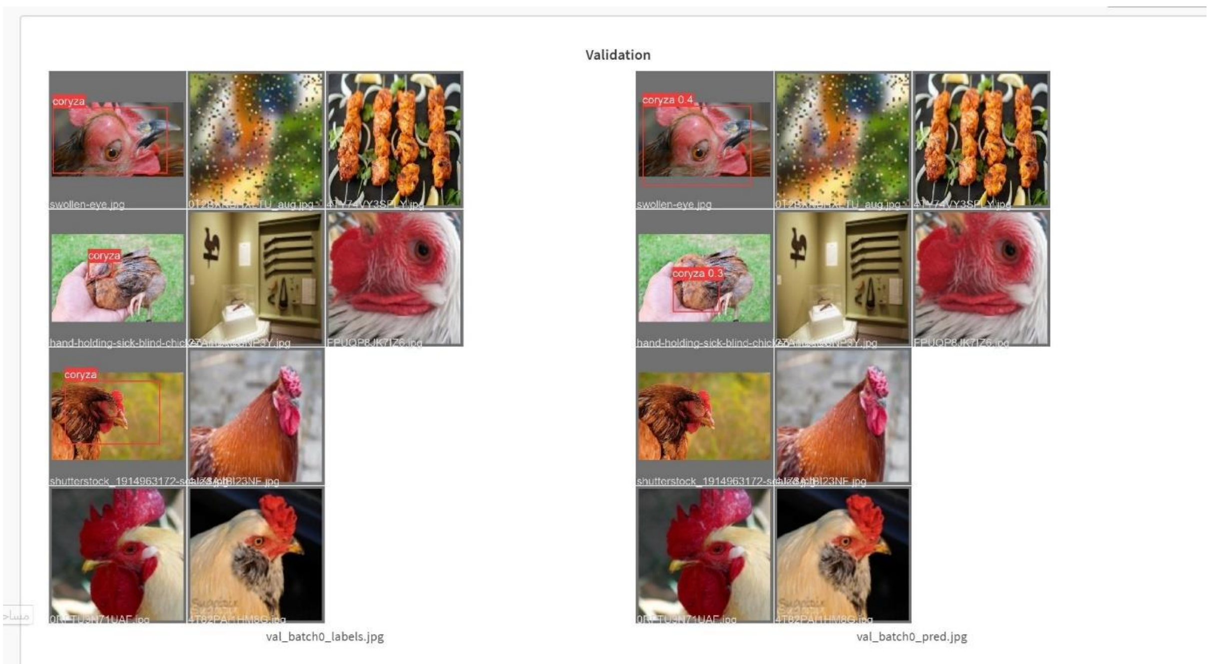

Figure 4-7: Detecting infectious coryza

4.4 The platform:

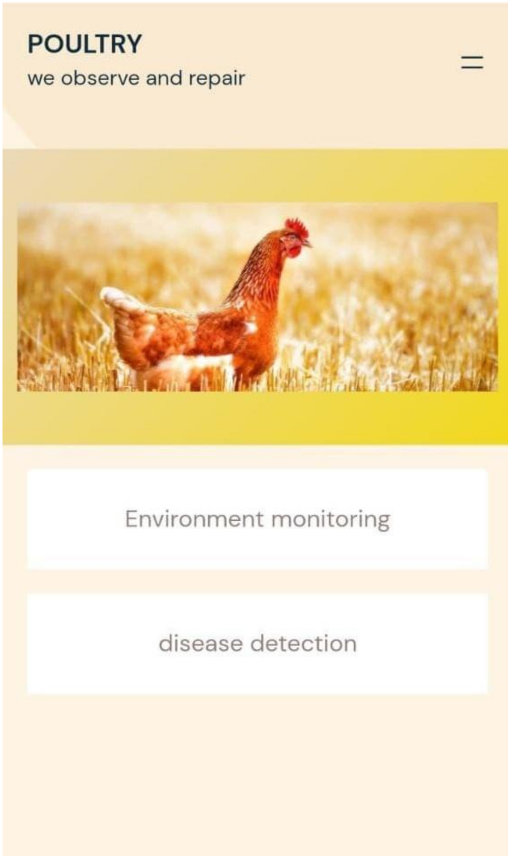

Figure 4-8: Platform

The platform worked well with the normal webpages but sometimes may not with node-red dashboard. using remote access node in the node red can be useful to solve this part.

CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

5.1 Introduction:

This chapter contains the conclusion and recommendation of the project, section 5.2 shows the conclusion, and section 5.3 shows our recommendation.

5.2 Conclusion:

The project allowed farmers to remotely supervise the status of the environment inside the farm from anywhere in the world, as well as early detection of Infectious Coryza disease by using image processing, that helped to reduce the labor, decrease the effort, increase the farm production, reduce the cost, and prevention from diseases caused by environmental issues.

This study provides a real-time disease detection management system based on an YOLOv5 model. The model has been trained and implemented the proposed strategy to enhance 'coryza' recognition in poultry farms by utilizing an open dataset accessible at Robo-flows. The experimental findings showed that the YOLO v5 algorithm has an overall accuracy of 72.3% for disease identification for mAP (0.5). In the near future, this study can be applied in autonomous poultry farms environments for increase the productivity of the poultry farm and avoiding the loss affecting the investment.

5.3 Recommendation:

The main recommendation is:

- To detect the ill chicken in an efficient way, so using a thermal camera can be useful, because high temperature is one of the common symptoms of various diseases that affect chickens. by detecting this symptom, it can cover a range of diseases, use FLIR Lepton 3.5 Thermal Camera is recommend, because it gives high accuracy results, and it can detect the dead chickens.
- Making the control part work automatic as an option along with the supervisor mode that we made, so that it does not need a farmer to run the cooling system or lights.
- Adding more sensors such as load cell to measure the weight of the chicken’s food to know remotely how much it ate in a day is recommended.
- Is recommended to use an analog to digital converter such as MCP3208 with LDR to give more accurate results.
- Recommended to use node red remote access node may perform better and present better results also the researcher may use different platform that works for both different sides of the project.

Reference:

1. INTELLIGENCE, M. 2021. *AUTOMATED POULTRY FARM MARKET - GROWTH, TRENDS, COVID-19 IMPACT, AND FORECASTS (2022 - 2027)* [Online]. Available: https://www.mordorintelligence.com/industry-reports/automated-poultry-farm-market [Accessed 18th of February 2021].

2. SITE, T. P. 2018. *Flocking to digital: the future of poultry technologies* [Online]. Available: https://www.thepoultrysite.com/articles/flocking-to-digital-the-future-of-poultry-technologies [Accessed 4th April 2021].

3. RAJ, A. A. G. & JAYANTHI, J. G. IoT-based real-time poultry monitoring and health status identification. 2018 11th International Symposium on Mechatronics and its Applications (ISMA), 2018. IEEE, 1-7.

4. STEVE. 2018. *Raspberry Pi GPIO programming in c* [Online]. Available: https://www.bigmessowires.com/2018/05/26/raspberry-pi-gpio-programming-in-c [Accessed 1st of September 2021].

5. PI, R. 2021. *Raspberry Pi 3 Model B* [Online]. Available: https://www.canakit.com/raspberry-pi-3-model-b.html [Accessed 1st of September 2021].

6. ARDUBITRONICS. 2021. *Sensor DHT11* [Online]. Available: https://www.ardubitronics.com/home/137-sensor-dht11.html [Accessed 1st of September 2021].

7. 101, C. 2021. *DHT11–Temperature and Humidity Sensor* [Online]. Available: https://components101.com/sensors/dht11-temperature-sensor [Accessed 1st of September 2021].

8. ENGINEERS, L. M. 2022. *How DHT11 DHT22 Sensors Work & Interface With Arduino* [Online]. Available: https://lastminuteengineers.com/dht11-dht22-arduino-tutorial/ [Accessed 28th of February 2022].

9. YUSTRES, Á., LÓPEZ-VIZCAÍNO, R., CABRERA, V. & NAVARRO, V. 2021. Physicochemical and Hydrodynamic Aspects of Soil. Electrochemically Assisted Remediation of Contaminated Soils. Springer.

10. BRENTSCHEERLINCK. 2021. *LDR* [Online]. Available: https://www.instructables.com/LDR/ [Accessed 1st of September 2021].

11. EEPOWER. 2021. *Photo-resistor* [Online]. Available: https://eepower.com/resistor-guide/resistor-types/photo-resistor [Accessed 1st of September 2021].

12. ELECOK. 2022. *MARTECH P1248025HS1N 24V 0.14A 3.36W 2 Wires Cooling Fan* [Online]. Available: https://www.elecok.com/martech-p1248025hs1n-24v-0-14a-3-36w-2-wires-cooling-fan.html [Accessed 27th of February 2022].

13. ANALOG. 2021. *Why and How to Control Fan Speed for Cooling Electronic* [Online]. Available: https://www.analog.com/en/analog-dialogue/articles/how-to-control-fan-speed [Accessed 2nd of September 2021].

14. SMARTBITBN. 2021. *Red LED 5mm (20 pcs)* [Online]. Available: https://smartbitbn.com/product/red-led-5mm-20-pcs/ [Accessed 2nd of September 2021].

15. INTERNATIONAL, L. 2020. *What are LEDs?* [Online]. Available: https://www.ledsinternational.com/what-are-leds/ [Accessed 2nd of September 2021].

16. WINGS, E. 2020. *LCD 16x2 Display Module* [Online]. Available: https://www.electronicwings.com/sensors-modules/lcd-16x2-display-module, [Accessed 2nd of September 2021].

17. COMPUTER.IS. 2021. *Raspberry Pi myndavél V2 8MP/1080P* [Online]. Available: https://www.computer.is/is/product/raspberry-pi-myndavel-v2-8mp-1080p [Accessed 3rd of September 2021].

18. SUPPLY, P. 2021. *Raspberry Pi Camera Board v2.1 (8MP, 1080p)* [Online]. Available: https://uk.pi-supply.com/products/raspberry-pi-camera-board-v2-1-8mp-1080p [Accessed 3rd of September 2021].

19. MICROCHIP. 2020. *Microchip 2000kbit OTP EPROM Memory 32-Pin PDIP, AT27C020-90PU* [Online]. Available: https://ie.rs-online.com/web/p/eprom/1274062 [Accessed 19th of September 2021].

20. 14, E. 2021. *mcp3208 – CI/P* [Online]. Available: https://in.element14.com/microchip/mcp3208-ci-p/ic-adc-12-bit-100ksps-spi-dip/dp/1084269 [Accessed 19th of September 2021].

21. PLANET, I. W. 2019. *LM35 Linear Temperature Sensor Module* [Online]. Available: https://www.iotwebplanet.com/product/lm35-linear-temperature-sensor-module/ [Accessed 19th of September 2021].

22. PROJECT, E. 2021. *LM35 Temperature Sensor Pin out, Interfacing guide, Circuit Construction and Working Principals* [Online]. Available: https://www.engineersgarage.com/lm35-description-and-working-principal [Accessed 19th of September 2021].

23. CHIP, T. 2022. *1UF 50V CAPACITOR* [Online]. Available: https://www.twinschip.com/1uF_50V_Capacitor [Accessed 7th of March 2022].

24. WIKIPEDIA. 2022. *Proteus Design Suite* [Online]. Available: https://en.wikipedia.org/wiki/Proteus_Design_Suite [Accessed 6th of February 2022].

25. WIKIPEDIA. 2022. *PyTorch* [Online]. Available: https://en.wikipedia.org/wiki/PyTorch [Accessed 6th of February 2022].

26. NODE-RED. 2022. *Node-RED* [Online]. Available: https://nodered.org/#features [Accessed 24th of February 2022].

27. TAN, M., PANG, R. & LE, Q. V. Efficientdet: Scalable and efficient object detection. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020. 10781-10790.

28. GARG, A. 2021. *How to Use Yolo v5 Object Detection Algorithm for Custom Object Detection* [Online]. Available: https://www.analyticsvidhya.com/blog/2021/12/how-to-use-yolo-v5-object-

detection-algorithm-for-custom-object-detection-an-example-use-case/ [Accessed 7th of March 2022].

29. SOLAWETZ, J. 2020. *YOLOv5 New Version - Improvements And Evaluation* [Online]. Available: https://blog.roboflow.com/yolov5-improvements-and-evaluation/ [Accessed 2nd of March 2022].

30. GAD, A. F. 2021. *Evaluating Deep Learning Models: The Confusion Matrix, Accuracy, Precision, and Recall* [Online]. Available: https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/ [Accessed 7th of March 2022].

31. JONES, A. 2022. *Google Colab: An Online Jupyter Notebook That You Should Definitely Try* [Online]. Available: https://towardsdatascience.com/google-colab-an-online-jupyter-notebook-that-you-should-definitely-try-2572a3d4afb6 [Accessed 2nd March 2022].

32. SKALSKI, P. 2022. *makesense.ai* [Online]. Available: https://github.com/SkalskiP/make-sense [Accessed 2nd March 2022].

33. THEOPHILUS, S. 2021. *Roboflow: Converting Annotations for Object Detection* [Online]. Available: https://medium.com/analytics-vidhya/converting-annotations-for-object-detection-using-roboflow-5d0760bd5871 [Accessed 2nd March 2022].

34. AGGARWAL, A. 2019. *Introduction to Visual Studio* [Online]. Available: https://www.geeksforgeeks.org/introduction-to-visual-studio/ [Accessed 2nd of March 2022].

35. FOUNDATION, W. 2022. *WordPress* [Online]. Available: https://en.wikipedia.org/wiki/WordPress [Accessed 9th of March 2022].

36. PASTORET, P.-P. & CHAISEMARTIN, D. J. R. S. E. T. 2011. The importance of governance and reliable veterinary certification. 30**,** 347-52, 339.

37. BEN SASSI, N., AVERÓS, X. & ESTEVEZ, I. J. A. 2016. Technology and poultry welfare. 6**,** 62.

38. SANTOSHKUMAR, K. C., CHAVHAN, S. J. M. C. & ENGINEERING, P. 2012. Development of wireless sensor node to monitor poultry farm. 296**,** 27-32.

39. NIIMI, A., WADA, M., ITO, K., TODA, M., HATANAKA, K., KONISHI, O. J. A. L. & ROBOTICS 2008. Broiler-house environment monitoring system using sensor network and mail delivery system. 13**,** 264-270.

40. MITKARI, S., PINGLE, A., SONAWANE, Y., WALUNJ, S. & SHIRSATH, A. J. S. 2019. IOT based smart poultry farm. 6.

41. SINDUJA, K., JENIFER, S. S., ABISHEK, M. S., SIVASANKARI, B. J. I. R. J. O. E. & TECHNOLOGY 2016. Automated control system for poultry farm based on embedded system. 3**,** 620-624.

42. ALEX, A. G. R. & JOSEPH, G. J. Real-Time Poultry Health Identification Using IoT Test Setup, Optimization and Results. International Symposium on Signal Processing and Intelligent Recognition Systems, 2018. Springer, 30-40.

43. ADA, L. 2019. *DHT11, DHT22 and AM2302 Sensors* [Online]. Available: https://learn.adafruit.com/dht/overview [Accessed 9th of July 2021].

44. PAPUTUNGAN, I. V., AL FARUQ, A., PUSPASARI, F., AL HAKIM, F., FAHRURROZI, I., OKTIAWATI, U. Y. & MUTAKHIROH, I. Temperature and Humidity Monitoring System in Broiler Poultry Farm. IOP Conference Series: Materials Science and Engineering, 2020. IOP Publishing, 012010.

45. FANG, C., HUANG, J., CUAN, K., ZHUANG, X. & ZHANG, T. J. B. E. 2020. Comparative study on poultry target tracking algorithms based on a deep regression network. 190**,** 176-183.

46. LIU, H.-W., CHEN, C.-H., TSAI, Y.-C., HSIEH, K.-W. & LIN, H.-T. J. S. 2021. Identifying images of dead chickens with a chicken removal system integrated with a deep learning algorithm. 21**,** 3579.

47. MURAD, M., YAHYA, K. M. & HASSAN, G. M. Web based poultry farm monitoring system using wireless sensor network. Proceedings of the 7th International Conference on Frontiers of Information Technology, 2009. 1-5.

48. MIRZAEE-GHALEH, E., OMID, M., KEYHANI, A., DALVAND, M. J. C. & AGRICULTURE, E. I. 2015. Comparison of fuzzy and on/off controllers for winter season indoor climate management in a model poultry house. 110**,** 187-195.

49. ZHANG, Y., CHEN, Q., LIU, G., SHEN, W. & WANG, G. J. I. J. O. D. S. N. 2016. Environment parameters control based on wireless sensor network in livestock buildings. 12**,** 9079748.

50. ASTILL, J., DARA, R. A., FRASER, E. D., ROBERTS, B., SHARIF, S. J. C. & AGRICULTURE, E. I. 2020. Smart poultry management: Smart sensors, big data, and the internet of things. 170**,** 105291.

51. GLOWACZ, A. J. M. 2021. Fault diagnosis of electric impact drills using thermal imaging. 171**,** 108815.

52. ZHU, W., PENG, Y. & JI, B. An automatic dead chicken detection algorithm based on SVM in modern chicken farm. 2009 Second International Symposium on Information Science and Engineering, 2009. IEEE, 323-326.

53. ASTILL, J., DARA, R. A., FRASER, E. D. & SHARIF, S. J. F. I. V. S. 2018. Detecting and predicting emerging disease in poultry with the implementation of new technologies and big data: A focus on avian influenza virus. 5**,** 263.

54. NODE-RED. 2022. *The Core Nodes* [Online]. Available: https://nodered.org/docs/user-guide/nodes [Accessed 25th of February 2022].

55. NODE-RED. 2022. *node-red-contrib-dht-sensor* [Online]. Available: https://flows.nodered.org/node/node-red-contrib-dht-sensor [Accessed 25th of February 2022].