GREENHOUSE MONITORING AND CONTROLLING SYSTEMS USING IMAGE PROCESSING AND SENSORY TECHNIQUES

2021
MASTER THESIS
ELECTRICAL AND ELECTRONICS ENGINEERING

ABDALRAZG FATHI BENNASER

Thesis Advisor

Prof. Dr. Necmi Serkan TEZEL

GREENHOUSE MONITORING AND CONTROLLING SYSTEMS USING IMAGE PROCESSING AND SENSORY TECHNIQUES

Abdalrazg Fathi BENNASER

T.C.
Karabuk University
Institute of Graduate Programs
Department of Electrical and Electronics Engineering
Prepared as
Master Thesis

Prof. Dr. Necmi Serkan TEZEL

KARABUK
December 2021

I certify that in my opinion the thesis submitted by Abdalrazg Fathi BENNASER titled "GREENHOUSE MONITORING AND CONTROLLING SYSTEMS USING IMAGE PROCESSING AND SENSORY TECHNIQUES" is fully adequate in scope and in quality as a thesis for the degree of Master's in Electrical and Electronics Engineering.

Assoc. Prof. Dr. Necmi Serkan TEZEL ....
Thesis Advisor, Department of Electrical and Electronics Engineering

This thesis is accepted by the examining committee with a unanimous vote in the Department of Electrical and Electronic Engineering as a Master's thesis. 23.12.2021

Examining Committee Members (Institutions)		Signature
Chairman	: Assoc. Prof. Dr. Necmi Serkan TEZEL (KBU)	....
Member	: Assoc. Prof. Dr. Ziyodulla YUSUPOV (KBU)	....
Member	: Assoc. Prof. Dr. Turgut ÖZTÜRK (BTÜ)	....

The degree of Masters by the thesis submitted is approved by the Administrative Board of the Institute of Graduate Programs, Karabuk University.

Prof. Dr. Hasan SOLMAZ ....
Director of the Institute of Graduate Programs

"I declare that all the information within this thesis has been gathered and presented in accordance with academic regulations and ethical principles and I have according to the requirements of these regulations and principles cited all those which do not originate in this work as well."

Abdalrazg Fathi BENNASER

ABSTRACT

M. Sc. Thesis

GREENHOUSE MONITORING AND CONTROLLING SYSTEMS USING IMAGE PROCESSING AND SENSORY TECHNIQUES

Abdalrazg Fathi BENNASER

Karabuk University
Institute of Graduate Programs
The Department of Electrical and Electronics Engineering

Thesis Advisor:
Assoc. Prof. Dr. Necmi Serkan TEZEL
December 2021, 76 pages

Designs and Simulations for greenhouse system is significant research subject due to critical need of systems that can bridge the gaps in manpower and scalability. Therefore, monitoring and control system are necessary to achieve agricultural production goals and cost control. Image processing is a rapidly developing technique that allows the identification of non-conforming samples and assigning a specific value to each condition based on machine learning methods. Moreover, Control system depending on microcontrollers, sensors, and intervention tools were found effective in the greenhouse context. The aim of the current research is to design a monitoring system for the conditions of plantations in greenhouses using image processing techniques and allow for controlled conditions based on automation, as well as human judgement, through a controlling system with sensory techniques adjusted using an android phone application. A monitoring system was designed through using image processing and CNN machine learning techniques.

The system was developed through the use of Raspberry Pi 4 Model B as a microprocessor, a Raspberry Camera rev 1.0, and a machine learning code. The outcomes of image processing were judged on a binary system of 1 = wanted and 0 = unwanted for image classification. The accuracy of the system was tested through three arrangements: 60% training/ 40% testing with an accuracy of 96.32%, 70% training/ 30% testing with an accuracy of 98.43%, and 80% training/ 20% testing with an accuracy of 99.14%. The greenhouse setting was put to test and the system was able to correctly identify all samples according to their true classification. A comparison with the literature showed that the designed system is superior based on the complexity of criteria through an RGB classification. The control system was designed to monitor vital greenhouse parameters: temperature and humidity, rain, and soil moisture. An Arduino UNO microcontroller was used to control the system. Control components included ventilation fans, infrared lamp, water pump, side and top louvers, and spectrum LED light. A mobile application was designed as a user interface to monitor changes in the environment, in addition to functions of system components in response to changes in the vital greenhouse parameters. A simulation of the setup was performed through forced testing modes: increase in temperature (mode 1), decrease in temperature (mode 2), decrease in soil moisture level (mode 3), decrease in humidity level (mode 4). The efficiency of the system is proven through the testing modes and the comparison to similar systems in the literature showed the extensively and dependability of the designed control system.

Key Words **:** Greenhouse automation, image processing, control system.

Science Code : 90541

ÖZET

Yüksek Lisans Tezi

GÖRÜNTÜ İŞLEME VE DUYUSAL TEKNİKLER KULLANILAN SERA İZLEME VE KONTROL SİSTEMLERİ

Abdalrazg BENNASER

Karabük Üniversitesi
Lisansüstü Eğitim Enstitüsü
Eliktrik ve Elektronik Mühendisliği Anabilim Dalı

Tez Danışmanı:
Doç. Dr. Necmi Serkan TEZEL
Aralık 2021, 76 sayfa

Sera sistemi için Tasarımlar ve Simülasyonlar, insan gücü ve ölçeklenebilirlikteki boşlukları kapatabilecek sistemlere olan kritik ihtiyaç nedeniyle önemli bir araştırma konusudur. Bu nedenle, tarımsal üretim hedeflerine ve maliyet kontrolüne ulaşmak için izleme ve kontrol sistemi gereklidir. Görüntü işleme, makine öğrenmesi yöntemlerine dayalı olarak uygun olmayan örneklerin tanımlanmasına ve her koşula belirli bir değerin atanmasına olanak tanıyan hızla gelişen bir tekniktir. Ayrıca sera bağlamında mikrodenetleyicilere, sensörlere ve müdahale araçlarına dayalı kontrol sistemi etkili bulunmuştur. Mevcut araştırmanın amacı, görüntü işleme teknikleri kullanılarak seralardaki fidanlık koşulları için bir izleme sistemi tasarlamak ve bir android telefon kullanılarak ayarlanan duyusal tekniklerle bir kontrol sistemi aracılığıyla, otomasyona dayalı kontrollü koşullara ve ayrıca insan yargısına izin vermektir. uygulama. Görüntü işleme ve CNN makine öğrenmesi teknikleri kullanılarak bir izleme sistemi tasarlanmıştır. Sistem, mikroişlemci olarak Raspberry

Pi 4 Model B, Raspberry Camera rev 1.0 ve makine öğrenme kodu kullanılarak geliştirildi. Görüntü işlemenin sonuçları, görüntü sınıflandırması için 1 = istenen ve 0 = istenmeyen ikili bir sistemde değerlendirildi. Sistemin doğruluğu üç düzenleme ile test edilmiştir: %60 eğitim/ %96,32 doğrulukla %40 test, %98,43 doğrulukla %70 eğitim/ %30 test ve %80 eğitim/ %20 doğrulukla test %99.14. Sera ayarı teste tabi tutuldu ve sistem tüm numuneleri gerçek sınıflandırmalarına göre doğru bir şekilde tanımlayabildi. Literatürle bir karşılaştırma, tasarlanan sistemin bir RGB sınıflandırması yoluyla kriterlerin karmaşıklığına dayalı olarak üstün olduğunu göstermiştir. Kontrol sistemi, hayati sera parametrelerini izlemek için tasarlandı: sıcaklık ve nem, yağmur ve toprak nemi. Sistemi kontrol etmek için Arduino UNO mikrodenetleyici kullanılmıştır. Kontrol bileşenleri havalandırma fanları, kızılötesi lamba, su pompası, yan ve üst panjurlar ve spektrum LED ışığını içeriyordu. Hayati sera parametrelerindeki değişikliklere yanıt olarak sistem bileşenlerinin işlevlerine ek olarak, ortamdaki değişiklikleri izlemek için bir kullanıcı arayüzü olarak bir mobil uygulama tasarlanmıştır. Kurulumun bir simülasyonu, zorunlu test modları aracılığıyla gerçekleştirilmiştir: sıcaklıkta artış (mod 1), sıcaklıkta azalma (mod 2), toprak nem seviyesinde azalma (mod 3), nem seviyesinde azalma (mod 4). Sistemin verimliliği test modları aracılığıyla kanıtlanmıştır ve literatürdeki benzer sistemlerle karşılaştırma, tasarlanan kontrol sisteminin yaygınlığını ve güvenilirliğini göstermiştir.

Anahtar Kelimeler : Sera otomasyonu, görüntü işleme, kontrol sistemi.

Bilim Kodu : 90541

ACKNOWLEDGMENT

I would like to express my deepest gratitude and thanks to my advisor, Prof. Dr. Necmi Serkan TEZEL, for his advice and guidance in the development and completion of this research.

CONTENTS

Page

Page

LIST OF FIGURES

Page

Page

LIST OF TABLES

Page

SYMBOLS AND ABBREVIATIONS INDEX

SYMBOLS

P(c)	: Class prior probability
P(c|x)	: Posterior probability
P(x)	: Predictor prior probability
P(x|c)	: Likelihood
X	: Input variable matrix
y	: Output variable matrix
β	: Regression coefficient matrix
ε	: Error value matrix

ABBREVIATIONS

ANN	: Artificial Neural Network
BPNN	: Back Propagation Neural Network
CFD	: Computational Fluid Dynamics
CPU	: Central Processing Unit
DBN	: Deep Belief Network
DRC	: Electronic Design Automation
FSSEM	: Feature Subset Selection Expectation Maximization
GPU	: Graphics Processing Unit
HD	: High Definition
HDR	: High Dynamic Range
HMP	: Hierarchical Matching Pursuit
HPS	: High Performance System
ICSP	: In-Circuit Serial Programming
IoT	: Internet of Things
IP	: Internet Protocol

IPSO	: Improved Particle Swarm Optimization
KNN	: k-nearest neighbor
LED	: Light Emitting Diode
LLE	: Locally Linear Embedding
LPDDR	: Low-Power Double Data Rate
MC	: Microcontroller
ML	: Machine Learning
MOGA	: Multi-Objective Generic Algorithm
PCA	: Principal Component Analysis
POD	: Proper Orthogonal Decomposition
PWM	: Pulse-Width Modulation
RAM	: Random Access Memory
RGB	: Red Green Blue
SOM	: System on Module
SRAM	: Statis Random-Access Memory
SVM	: Support Vector Machine
TPU	: Thermal Processing Unit
VDD	: Drain to Drain Voltage
VGA	: Video Graphics Array

PART 1

INTRODUCTION

1.1. PROBLEM STATEMENT

The design of smart greenhouse system has been a subject of interest for several years, which is attributed to the scalability drive of agriculture to meet demands [1]. Current greenhouse techniques require high workmanship in monitoring and control that consume a lot of manhours and require competitive skills. These factors add several challenges to scalable agriculture such as costs and lack of enough human skills that can identify and control issues in crops [2]. Due to the development in electronics technology, it became possible to replace these skills with atomized systems to identify issues and allow the control of affecting conditions [3]. Furthermore, the current technologies in agriculture allow farmers to automate several recurring activities, including pesticide spraying, supporting soil with fertilizer, weeding unwanted growth, and plant irrigation [4]. These systems are mainly based on a pre-scheduled program and are more action oriented rather than comprehensive systems that can provide reliable data feedback. Additionally, the current research trend is developing and implementing a full diagnosis and control system that can identify issues within greenhouses and allow their owners to take decisions accordingly, despite the view of it being a very unrealistic based on the current achievements in research and practice [5].

Image processing is a rapidly developing technique that allows the identification of non-conforming samples and assigning a specific value to each condition based on machine learning methods. Agriculture has high potential to benefit from this technology as crops are highly identifiable by their appearance, which makes image sensory and processing a key solution that can replace or support skills in the greenhouse [6]. Moreover, greenhouses are used to provide controlled conditions to the crops based on needs and for growing them in severe outdoor conditions. The currently used techniques are conventional systems that require manpower presence in the facility. The availability of suitable sensory technology and action tools allow the control of these conditions remotely through a mobile application. Therefore, such a solution allows to sensor and control the conditions within the greenhouse, or multiple greenhouses, instantly without the need for physical human presence [7].

1.2. RESEARCH AIM AND OBJECTIVES

The main aim of the research is to design a monitoring system for the conditions of plantations in greenhouses using image processing techniques and allow for controlled conditions based on automation, as well as human judgement, through a controlling system with sensory techniques adjusted using an android phone application. The achievement of the targeted systems requires the fulfilment of the following objectives:

- Understand the needs of automated greenhouse systems and the critical criteria that need to be addressed in functional electronics design.
- Design a diagnostic monitoring system that provides an educated judgment of plant conditions based on database information collected from the field and image processing techniques.
- Design a diagnostic and interceptive controlling system that provides greenhouse conditions to the user, including heating and humidity, which are critical for plant growth and development. Information is fed to the system through sensory techniques into an interactive android application, where the user can adjust these conditions accordingly.
- Demonstrate the functions of the systems through an actual greenhouse set up, where the two systems function to achieve their design intents.
- Investigate the potential benefits of the systems and compare them to similar systems suggested in the literature to comprehend opportunities and limitations.

1.3. SCOPE, METHODS AND RESEARCH SIGNIFICANCE

The designed systems use different techniques according to the capabilities and needs of the intended functions. The monitoring system use image processing techniques with a reasonable database for daisy flowers. The system uses Raspberry Pi as a microcontroller for the system and provides feedback on plant conditions according to implemented algorithms. The controlling system uses sensory techniques for temperature and humidity conditions that is fed to an android mobile application, where the user can interact with these conditions by increasing them or decreasing them using an infrared light and a water tank. The system is operated using an Arduino microcontroller.

The designs targeted by the current research allows for alarming farmers of their plant conditions, monitoring greenhouse environment, and adjusting this environment to enhance plant conditions. Furthermore, these systems allow the remote interaction with these systems through a mobile application, which facilitates the scalability factor that forms a challenge in agriculture. This means that instead of designating a technician to each greenhouse, a single technician can monitor those systems and assign the necessary actions, which reduces costs, efficiency, and productivity in greenhouses. The image processing technique used in the monitoring system has been identified as a modern and promising research field by several journals [8,9], while research shows that such systems can be highly efficient in terms of performance and quality. The targeted monitoring system in the current research aims to exceed the 96% accuracy threshold that had been demonstrated in most previous research of agricultural grading systems [10]. Additionally, the controlling system added in this research allows for remote interaction that solves another issue in agriculture in regards with costs and scalability. The designs that are presented in this research form a key stepping point into technology implementation for wider range of plants and crops that are placed greenhouses.

1.4. RESEARCH STRUCTURE

The thesis report is structured into five main chapters to cover the theoretical and practical parts of the study. The theory of the research is mainly addressed in the first and second chapters, while the practical implementation is addressed in the remaining chapters. Accordingly, the research chapters are summarized as follows:

- Introduction: the problem of the research is stated, along with the proposed solution that is addressed through the course of the research. The main aim of the research is specified, as well as the specific objectives that are targeted. Moreover, the scope, method, and significance of the study are discussed to highlight the contribution to literature.
- Theoretical framework and literature: a review of the basics of the technologies utilized in the study: image processing and sensors. Moreover, the advances accomplished in the field of agriculture automation are presented with the different techniques that were proposed in the literature to solve the several problems around them. The results that have been achieved by the implementation of similar techniques are investigated for research planning and comparison purposes.
- Materials and methods: the elements necessary for the two identified techniques of the research are presented within this chapter. The first type of materials and methods are the electronic components that are used in the designed systems, along with their specifications. Furthermore, the algorithms used to handle the systems are structured and presented.
- System design and results: the proposed systems are designed on schematics and through physical implementation. The designs include the monitoring and controlling systems that are used as user interfaces. The systems are tested for their precision and accuracy.
- Discussion and conclusion: the systems and their outcomes are compared with similar systems presented in the literature. The final research findings are summarized and presented, in addition to recommendations regarding the designed systems and future research opportunities.

PART 2

THEORETICAL FRAMEWORK AND LITERATURE

2.1. GREENHOUSE AUTOMATION TECHNOLOGIES

Based on the agricultural, economic, and logistical needs in greenhouses, several opportunities were explored to automize systems and replace physical workmanship with different technological components. The main criterion that is monitored in greenhouses is temperature, which is vital for crop growth under controlled conditions. Therefore, the main focus of studies is on systems that can monitor, record, and control heating, as well as lighting, throughout the growth cycle of the crop. Nonetheless, technological developments for greenhouses did not stop at that criterion, but also expanded the possibilities towards other systems that can intervene with other environmental factors [11]. The current advancements in greenhouse system were classified into seven main categories, as illustrated in Figure 2.1. The classification is based on several factors, including the extent of human interference and the purpose of the system.

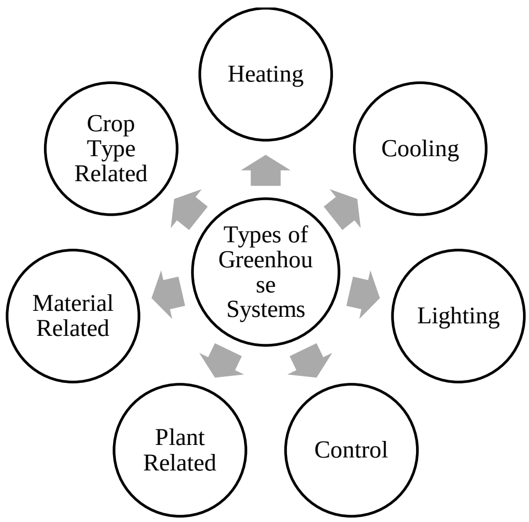


Figure 2.1. Targeted types of greenhouse systems [12].

The heating greenhouse technologies are classified into passive systems that are focused on solar collectors and concentrators, and active systems operating through heat pumps and boilers. The cooling technologies are also classified into passive systems through natural ventilators and shading, and active systems through mechanical ventilators and earth to air heat exchangers. The lighting technologies work by either reinforcing natural lighting sources or using supplementary systems such as HPS and LED lamps. The control systems are integrative systems that allow for the management of the various systems manually or automatically. Other systems related to plant-related needs are also implemented for the control of humidity, carbon dioxide, and soil temperature. The material related systems are concerned with the greenhouse type: envelope, insulation, or energy storage types. Moreover, the crop type systems monitor and control the greenhouse environment according to the needs of the crop as open, semi-closed, or closed environments [12].

Atomization and control technologies in greenhouses are deemed to be one of the most important research topics due to their implication on food securities and qualities. Thus, research in recent years is focused on specific systems that can enhance greenhouse conditions to maximize crop yield in one or more aspects [13]. Pennisi & Orsini [14] established the artificial lighting specifications that can maximize the yield of lettuce crops in accordance with the absorption peaks of the chlorophylls in the plant. However, the required lighting conditions vary according to the age and growth cycle of lettuce, which requires an automated system to identify and adjust based on inputs from the greenhouse. Atilgan et al. [15] investigated the response of tomato plants to the different conditions of LED lighting and its role in altering nutrient concentrations, while certain lighting conditions were recommended through a controlled system. Shang and Fu [16] studied the possibility of using Internet of Things (IoT) and Zigbee in providing the best conditions for a crop, which led into increasing yield and enhancing quality, in addition to lowering material consumption and providing the needed protection for the crop. Furthermore, several studies are attempting to integrate one or more system into an automated, monitored, and controlled greenhouse set up for those identified goals.

2.2. IMAGE PROCESSING AND MACHINE LEARNING

Image processing and machine learning are among the most important technologies that are used in the atomization of greenhouses. Some real-life implementations and research cases can be found to support the existence of these systems, while most of them are considered in the development stage with several enhancements required for their functionality and accuracy [5]. A Greenhouse system based on autonomous plant cultivation, which is currently being operated at a German base in Antarctica. Scientists have developed a unique system, which allows the greenhouses autonomously operate with no human intervention using image processing of plants. In the afore-mentioned system, the monitoring of crop growth and the automation of agricultural production is carried out respectively. A collection of images from different automated systems through approach that is based on neural networks, thus outperforming other most popular codecs, is conducted by reducing the size of the image with no apparent degradation in image quality. The researchers have used the obtained information from images that are reconstructed in order to train the computer vision algorithm capable of classifying almost 18 different kinds of plants by their respective species at different development stages. The obtained accuracy is approximately 92.6% [17].

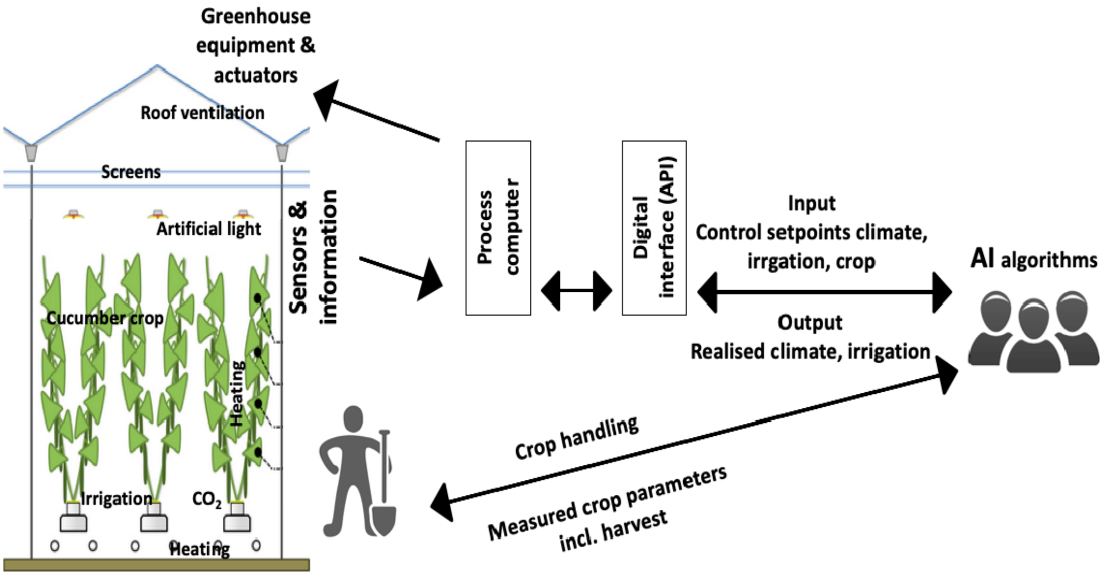


Figure 2.2. Example of atomization systems in greenhouses [12].

2.2.1. Types of Machine Learning

Machine learning provides an automatic learning ability that develops with the aid of system experience with no programming performed explicitly. It focuses on computer program development which can access a data and learn its patterns to provide an automated output through learning. It is considered to be a statistical tool, which gives computer systems the complete ability to improve, by learning without being programmed all the time. The learning process of the machines start with the data or observations such as direct experience, examples, and instructions etc. Thereafter, they make decisions based on previous examples with certain results that are provided. In such systems, human intervention is not required, and the systems adjust their actions according to the situation [18]. The extent of human intervention into the system determines the type of machine learning, as shown in Figure 2.3.

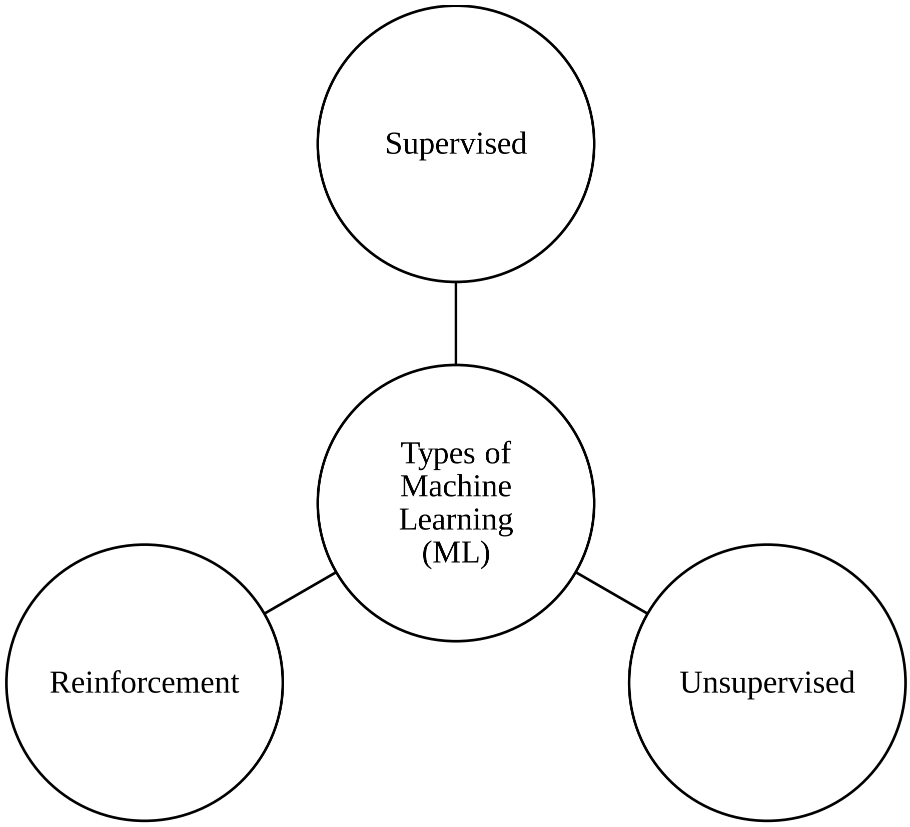


Figure 2.3. Types of machine learning according to human intervention [19].

There are three main types that care classified according to their human intervention extent and the type of learning that is implemented [20]:

- Supervised machine learning: a training data is given to the system in order to learn the relationship between the inputs and outputs through regression modelling and classification algorithms. Adjustment to the system is performed through comparisons with training data and feedback from a human operator.
- Unsupervised machine learning: An analysis is performed by the algorithm to figure out patterns and trends in the data using clustering and dimension reduction. The system is able to implement adjustment through its outputs without needing feedback from a human operator.
- Reinforcement machine learning: the returns of the system are maximized as the algorithm learns the data deeper and the return outputs reward its inputs using either a model-free or model-based algorithms.

2.2.1.1. Supervised Learning Algorithm

Supervised learning is a type of machine learning that maps input, and output based on the pairs of sample inputs-outputs. During supervised learning, the output values are given to input values which in turn are given to a network. Then, the network instantly updates its weights to generate the output hence desired for given inputs. Error between the expected outputs and outputs of network is calculated and the data are arranged by this error margin. During the calculation of error margin, differences between all expected outputs and the outputs obtained from the network are calculated. Looking at the obtained difference, error per cell margin is found. After that, each cell gets its own weight updated [21]. The methods by which supervised learning operates is illustrated in Figure 2.4.

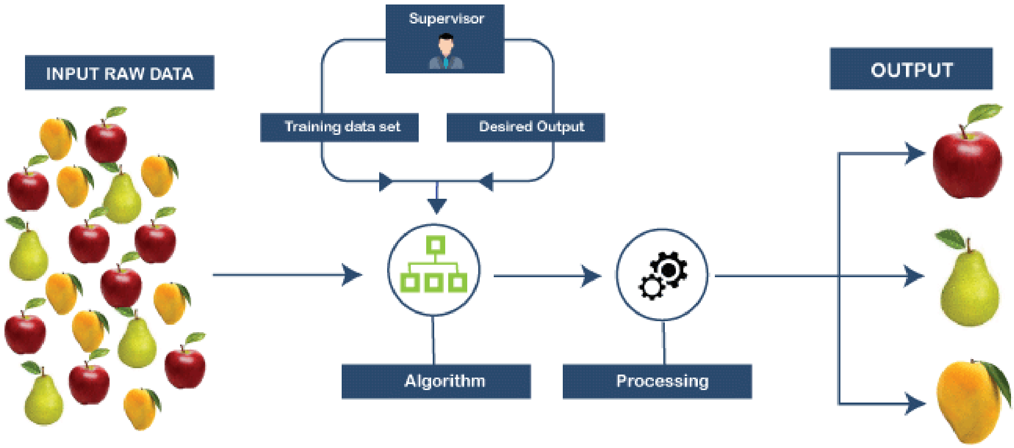


Figure 2.4. Supervised learning [22].

There are several algorithms that are used in supervised learning. The simplest method is linear regression where a prediction is made through projecting the most fitted line for all data points between the two variables. The formula of linear regression is given by the following equation (1) [21]:

$$y = X\beta + \varepsilon \tag{2.1}$$

Where,

y is the output variable matrix
X is the input variable matrix
β is the regression coefficient matrix
ε is the error value matrix

Logistic regression is another supervised learning algorithm that is used to calculate the probability of a set of output values based on a dataset. The outputs of the system are of a binary nature, as the algorithm measures the conformity of the output on either a zero (0) or one (1) scale. The algorithm has three main steps in providing an output [21]:

- Calculating the log value of probability.
- Calculating the value of probability.

- Then, calculating the output probability of a certain event.

K-nearest neighbor (KNN) is another supervised algorithm that classifies data based on their proximity. The model calculates the consistency of the data and its distribution and establishes an optimal weighting scheme to learn data patterns. Other algorithms are used in supervised learning, such as support vector machine (SVM), decision tree, random forest, gradient boosting machine, neural networks, and naïve Bayesian networks, which is used for huge datasets and operates according to the following algorithm in equations (2) and (3) [21]:

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)} \tag{2.2}$$

Where,

$P(c|x)$ is posterior probability

$P(x|c)$ is likelihood

$P(c)$ is class prior probability

$P(x)$ is predictor prior probability

$$P(c \mid x) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c) \tag{2.3}$$

2.2.1.2. Unsupervised Learning Algorithm

The unsupervised learning comprises the trained data which are neither labeled nor classified when compared to afore-mentioned supervised learning algorithms. This type of machine learning examines the given untagged data, tries to look for the commonalities and eventually reveal the data structure. In some cases, the system may not be able to find correct output, however, it can look for the data and try to find it out from the given data sets in order to identify from the untagged data [23]. The general process of unsupervised learning is illustrated in Figure 2.5.

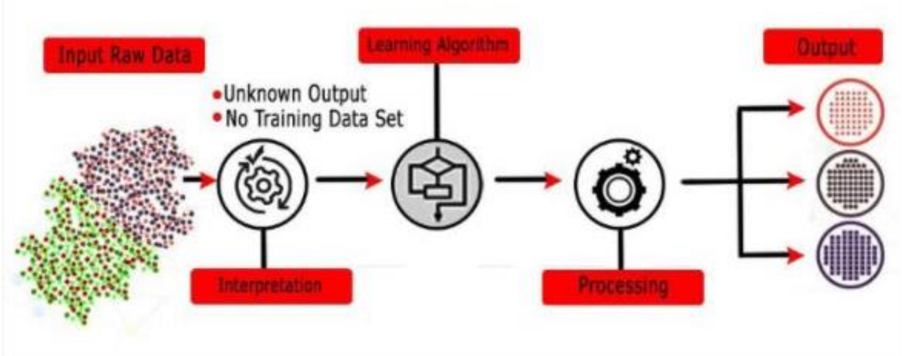


Figure 2.5. Unsupervised learning [22].

There are several techniques that has been used for unsupervised learning, which are mainly based on either linear or non-linear models to learn data patterns and process them according to an algorithm to provide the output, such as [23]:

- Deep belief networks (DBNs): this method trains the algorithm for maximum likelihood. The weights are calculated using equation (4) using the probability of visible vectors calculated by equation (5). The method keeps updating the hidden units using equation (6).

$$\omega_{ij}(t+1) = \omega_{ij}(t) + \eta \frac{\partial \log(p(\upsilon))}{\partial \omega_{ij}} \tag{2.4}$$

$$p(\upsilon) = \frac{1}{Z} \sum_{h} e^{-E(\upsilon,h)} \tag{2.5}$$

Where Z is the partition function

$$\Delta\omega_{ij} \alpha \left\langle \upsilon_i h_j \right\rangle_{data} - \left\langle \upsilon_i h_j \right\rangle_{reconstruction} \tag{2.6}$$

- Data clustering: the method divides the data set into clusters and applies the algorithm to different parts of it and it self-learns the patters and adjusts accordingly.
- Hierarchical matching pursuit (HMP): a complex type of algorithms that are structured like DBNs but for huge datasets [24].

- K-means comparison: the dataset is divided into clusters based on their proximity of patterns. Then, outputs are created and iterated several times to update the observations.
- Feature subset selection expectation maximization (FSSEM)
- Locally Linear Embedding (LLE)

2.2.1.3. Reinforcement Learning Algorithm

This type of learning algorithm can be defined as, “a learning method interacting with environment through generation of some actions and finds the errors in the system”. The most common features found in reinforcement learning algorithms are trial, delayed reward and error searching. Reinforcement learning method enables software agents and machines to determine ideal behavior automatically within a specified context in order to maximize the performance. To find out the best action, a simple reward feedback can be sufficient for the software agent. The feedback obtained by the software agent is called the reinforcement signal [25]. The process of reinforcement learning is illustrated in Figure 2.6.

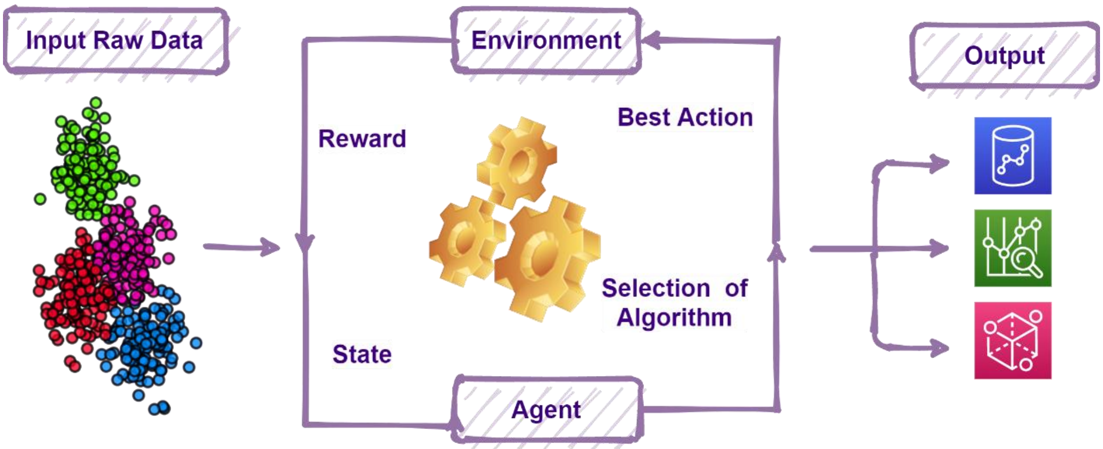


Figure 2.6. Reinforcement learning [26].

2.2.2. Image Processing

A digital image transformation is expressed as geometric transformation in image coordinate system. Each point in (x, y) coordinates of image A in a spatial transformation ought to be mapped in the new coordinate system as (u, v) points, as shown in Figure 2.7. The spatial transformation is required in order to perform the following [27]:

- Alignment of the images taken at different intervals or using different sensors.
- Correction of images in case of lens distortion.
- Correction of effects during camera orientation.
- Some special effects and image morphing.

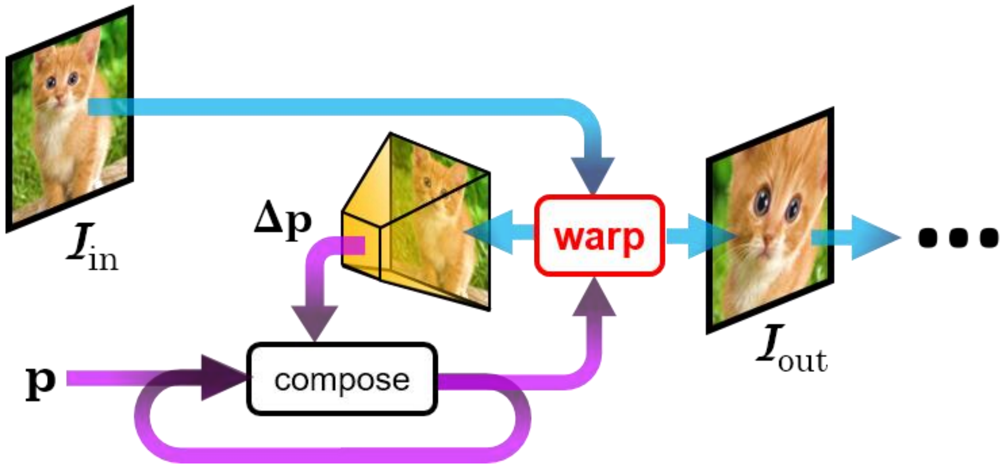


Figure 2.7. Spatial transformation [29].

Each (x, y) point in image A in a spatial transformation ought to be mapped in the new coordinate system as (u, v) point.

2.2.2.1. Filtering

2.2.2.1.1. Digital Filters

In order to sharpen and blur the digital images, digital filtering technique is used. The filtering process may be performed using the filter array given in the specific frequency regions of the spatial domain masking found in the Fourier domain [28].

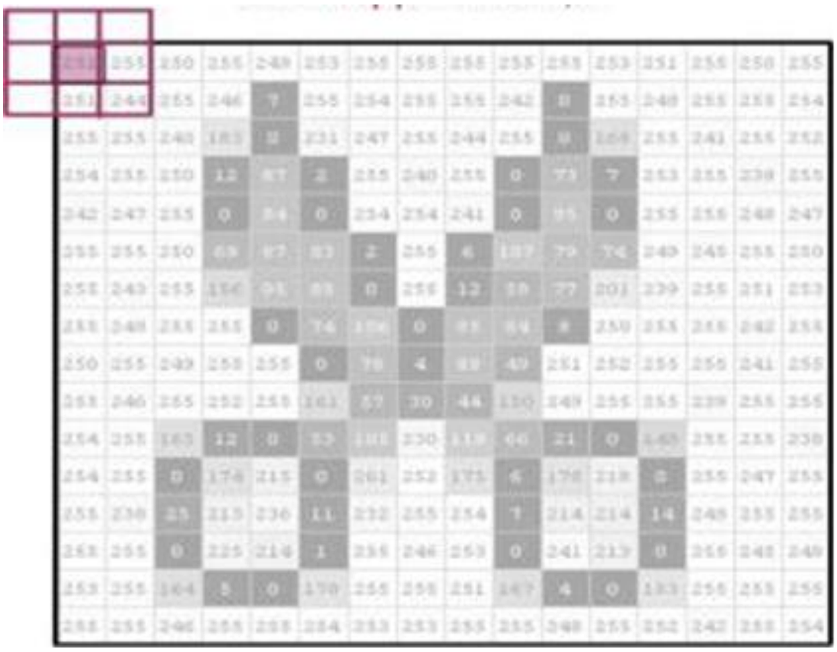


Figure 2.8. Filtering applied at (1,1) [29].

The example given in Figure 2.8 above has the filter applied to location (1,1) as there is known to be an inherent problem while working with the edges and corners. Problem to be considered in this case is that the neighbors are missing. The location (1,1) needs to be considered. In the Figure 2.8 above, there are no neighbors at the upper side or neighbors at the left side. The proposed solution for it is either replicating or zero padding, which is given in Figure 2.9 below. The blue highlighted pixels are added to the original image in this scenario [28].

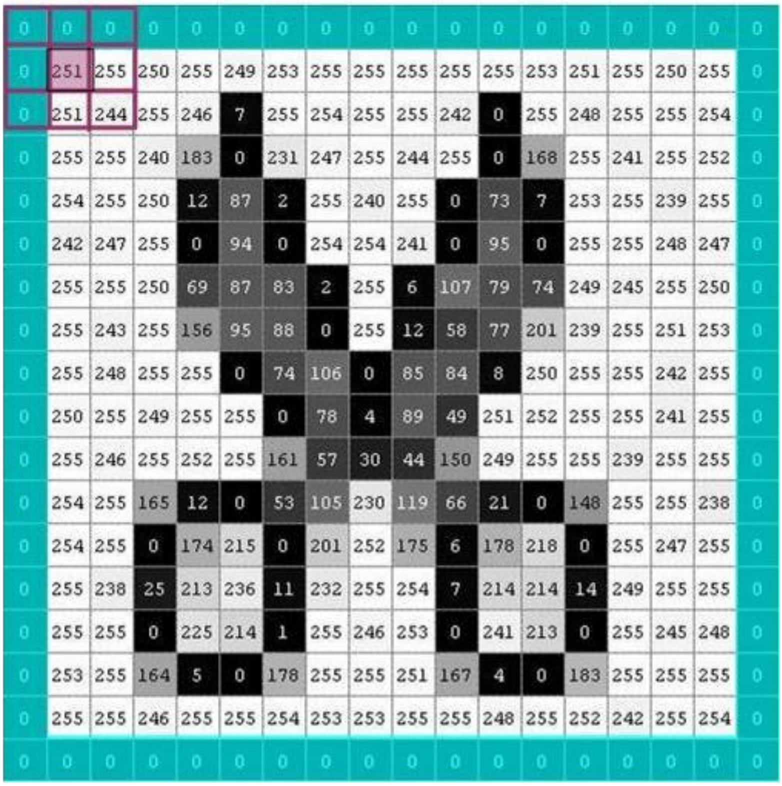


Figure 2.9. Zero padding [29].

2.2.2.1.2. Image Padding

Before the transformation to Fourier space, the images need to be typically padded. Filtered High-pass images below show different padding techniques as well as their results [30]. Figure 2.10 below shows the high-pass filtered images.

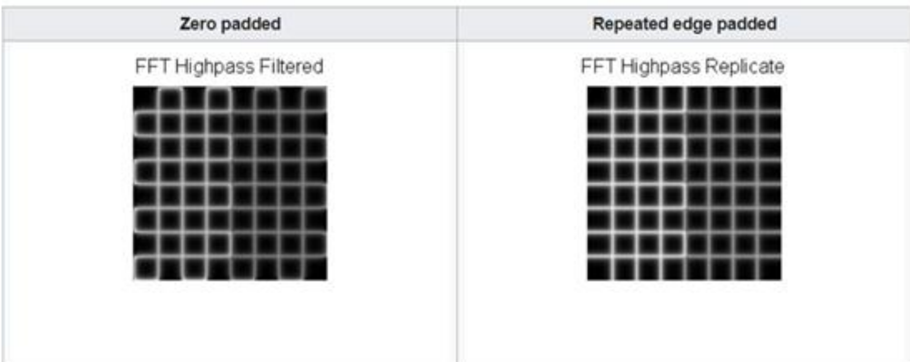


Figure 2.10. High-pass filtered images.

The point to be noticed is that when zero padded image is compared to edge padding, the high-pass filter is observed to have some extra edges [30].

2.2.2.1.3. Filtering Code Example

In the MATLAB example code given below in Figure 2.11, spatial high pass filtering domain can be observed.

```
img=checkerboard(20);                              % generate checkerboard
% ***************************  SPATIAL DOMAIN  ****************************
klaplace=[0 -1 0; -1 5 -1;  0 -1 0];                % Laplacian filter kernel
X=conv2(img,klaplace);                             % convolve test img with
                                                   % 3x3 Laplacian kernel
figure()
imshow(X,[])                                       % show Laplacian filtered
title('Laplacian Edge Detection')
```

Figure 2.11. Example code for High-pass Filtering.

2.2.2.2. Affine Transformations

This enables the basic transformations of images including rotate, scale, mirror, translate and shear etc. The step-by-step explanation of this transformation is as follows [31]:

- Image needs to be converted to matrix where each entry relates to a pixel intensity.
- Location of each pixel is denoted as vector showing the coordinates.
- Coordinates need to be multiplied using the affine-transformation matrix.
- Position is required in the output image for the pixel value to be copied.

In order to allow transformations, which require translation, 3D homogeneous coordinates are required. Third dimension is generally set to a constant, which is non-zero and usually 1. Therefore, the coordinate obtained is (x, y, 1) thus allowing the coordinate vector multiplication by 3x3 matrix and providing translation shifts. Hence, the third dimension allows the translation process [32].

As matrix multiplication is known to be associative, by multiplying each individual's transformation in a specific order, several affine transformations may be combined into one affine transformation. The resultant single matrix when applied to the point vector, instantly comes up with the same result as seen in all individual transformations that are performed in sequence on the vector (x,y,1). Therefore, an affine transformation matrices sequence can be transformed into single affine transformation matrix [31]. Figure 2.12 illustrates affine transformation.

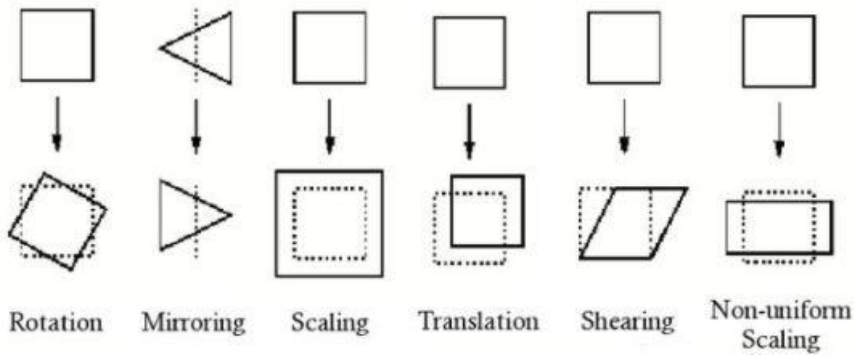


Figure 2.12. Affine transformation [33].

2.3. SURVEY OF LITERATURE APPLICATIONS IN GREENHOUSES

Modeling of the greenhouse is quite a valuable method in order to understand the effects of different parameters which influence the cooling or heating demand. In a study conducted by Choab et al. [34], the greenhouse model was used to obtain the optimal operating conditions of the greenhouse, and this carries the fundamental importance while selecting some parameters of greenhouse design as well as making management decisions during practical production. Greenhouse modeling has different aspects and may be evaluated according to functionality, accuracy, applicability, and portability. Therefore, designers ought to choose the features to examine or ignore thus depending on the purpose of the model. In their study, Zhang et al. managed to develop a precise model, which involved solar radiation equations consisting of high-resolution in order to aid a particular solar greenhouse [35]. In a study performed by Boulard et al. [36], the Computational Fluid Dynamics (CFD) method was employed to develop another highly reliable high-resolution model, which needed a quite high computational load. The study by Li et al. [37] proposes the mitigation of computational cost through an optimization scheme based on adopting a Proper Orthogonal Decomposition (POD). In the study conducted by Zhang et al. [35] on glass greenhouse, an unsteady-one-dimensional model was assumed, and the elements used for indoor climate were considered to be uniform. Their model determined the dynamic cover transmittance and absorbance which were caused by variations in the position of the sun and also the combined effects of soil, cover and air, making it a detailed model to be used in temperature simulation.

In another study, Zhao et al. [38] developed a model that could be rapidly computed and was able to be widely applicable. The proposed model can be evaluated for control and as a tool for humidity and temperature prediction in typical single-slope greenhouses. According to a report presented by Liu et al. [39], the aforementioned model is the main greenhouse model being used in the production of cold-season vegetable in an area of 1.96 million hectares in China. However, in a study by Wang et al. [40], these greenhouses being widely used lack the standardization considering their structure, size, and materials. When divided into three, two third of them are made up of soil walls, which are rammed, and the length and the width of the greenhouse changes from 50 to 100m and 7 to 12 m, respectively, thus making it difficult to employ the model to a greenhouse. In a study performed by Ahamed et al. [41], the reason behind difficulties in implementation of the model has been discussed. According to their data, the wall heat flux needs to be simulated based on case-by-case approach rather than being imposed as an experimental value and it was found that the gain of solar radiation changes depending on the size, structure, the material used and the tightness of the lid which directly influence the air leakage degree known as infiltration.

There have been numerous studies reported based on machine learning models thus predicting the temperature as well as humidity [42]. In addition, in a study by Jung et al. [43], different types of black box models have been proposed in order to simulate the humidity and temperature in glass greenhouses and such models have been reported to be using machine learning methods, for instance, deep learning and neural networks techniques. According to research, the black box models may easily be implemented to real greenhouses, and they can make some predictions regarding future climate conditions. Firstly, the data need to be gathered and then trained independently in order to make them applicable in different greenhouse types. However, in a study conducted by Righini et al. [44], it was mentioned that for white box models or mechanistic which are based on physical laws, the boundary conditions can limit the functioning of proposed models.

The budget calculation of the greenhouse would be almost impossible without knowing temperature on either sides of wall and without measuring the heat flux of the wall directly. In separately conducted studies by Sanchez-Molina et al. [45] and Rodriguez et al. [46], it was determined that many greenhouse models were designed by implementing the traditional modeling methods based on energy-balance concept and they were found to be generally performing well and accurate.

In another study performed by Liu et al. [47], a new transient model of greenhouse using mechanistic method was introduced in order to estimate the humidity and temperature in greenhouses. The proposed model was an easy-to-use and a novel method based on energy balance thus estimating the wall temperature was able to predict the future climate conditions of greenhouse by using the weather forecast only. In a report presented by Ravishankar et al. [48], an energy-balance model (tailored) was introduced for OSC-greenhouses in order to gain an understanding of potential that such systems carry to meet the needs of energy in greenhouses. Using developed energy model, which is able to predict power generation, energy load and light entering the greenhouse, the scope to use the data in exploring options for cell optimization, energy storage, supplemental lighting and optimization of plant growth is immense.

In a study performed by Lijun et al. [49], an adaptive predictive control thus combined with feedback linearization technique was proposed for greenhouse temperature system. Their predictive control, which incorporates feedback linearization, is able to cope with non-linearity in model, energy of running system and last but not the least set point tracking. According to a study by Speetjens et al. [50], the greenhouse layout and design can have modifications and the properties of cover material might change with time because of aging or either damage. Furthermore, the greenhouse is considered to be a living system which will almost never be the exact same thing as previously and originally planned.

In other studies on greenhouses conducted by Bang et al. [51] and Lascu et al. [52], it was determined that in order to deal non-linearity of the greenhouse temperature, there are some methodologies, which can prove to be useful while designing the controller based on non-linear geometric control theory, and this can transfer non-linear model accurately to linear model using the feedback control law. In a study carried out by Zangina et al. [53], robotic application of the path optimization was conducted for greenhouse by using experimentally evaluated functions of time, INSGAIII and NSGAIII approaches. According to the study, the required dosage was considered to be a variable for each plant when the robot capacity was fixed, and the refilling was necessary.

When a vast literature review is performed, mechanism models proposed in the past are found which provide a crystal-clear explanation of the concerned greenhouse environment. For example, the early dynamic and static model which was proposed by Bot et al. [54], while improved models proposed by Van Henten et al. [55] and static and dynamic models by De Zwart et al. [56] are being used for the purpose of finding the function related to metrological conditions as well as the greenhouse component parameters. In a study conducted by Çakır and Şahin [57], a mathematical model was developed in order to select the optimum greenhouse type in accordance with the size, location and position. Their results confirmed that greenhouses were not only usable but suitable to be used in colder climate regions so that to increase productivity. Moreover, Taki et al. [58] made a comparison between some mathematical models such as regression and innovative method known to be ANN (Artificial Neural Network), and they selected among them the best one which could predict roof temperature, inside air and energy transfer of the greenhouse.

The paper presented by Mohammadi et al. [59] proposed an experimental validation along with a dynamic model in order to estimate the variables of inside environment in semi-solar greenhouse. Therefore, they designed and constructed a semi-solar greenhouse. In another study conducted by Taki et al. [60], a comparison was made between SVM (Support Vector Machine) models and ANN (Artificial Neural Network) in order to predict three temperature points denoted as Ta, Ts and Tp. Hence, some variables, which were used as input, were examined as well as the relation between them. In a study by He and Ma [61], neural network with Principal Component Analysis (PCA) was applied in order to model the internal humidity of greenhouse and their results emphasized that the regression method implemented stepwise was less accurate as compared to PCA based BPNN. Another study focused on such model was presented by Yu et al. [62], proposing a completely new temperature prediction model which was based on least squares SVM (support vector machine) model having parameters thus optimized by an IPSO (improved particle swarm optimization). According to their results, IPSO was found to predict minimum and maximum temperature with greater accuracy when compared with BPNN. There are many algorithms which can be used in the afore-mentioned models. In another study conducted by Elsoragaby et al. [63], multi-objective genetic algorithm (MOGA) model was demonstrated to have an excess of energy inputs and the potential reduction in emission was found to be 19.6% and 46.37%, respectively in order to broadcast and transplant seeding methods.

In the literature review, there were many studies conducted by researchers such as Shamshirband et al. [64], Elhami et al. [65], Nabavi-Pelesaraei et al. [66], Tabatabaie et al. [67], Nabavi-Pelesaraei et al. [68], Khoshnevisan et al. [69], and Mousavi-Avval et al. [70] that focused on the calculation of input and output energies, and some of them had used SPSS software program in order to develop a linearity between inputs and outputs and some of them had used MATLAB software in order to develop MOGA, thus finding the best genetic algorithm to be used in optimization problem. The literature has many examples of greenhouses connected to technological devices and being controlled remotely. In a study conducted by Xiaohui et al. [71], a greenhouse environment was connected with the web server in order to acquire data from the parameters using mobile phone terminal and the greenhouse environment was controlled in order to achieve the most appropriate condition for the growth of crops.

In another similar study by Zhao et al. [72], it was stated that applying the Android system can increase the output, reduce energy consumption, and also can reduce the labor costs of the greenhouse environment. The real-time observation while connected to the smartphone terminal may adjust the greenhouse environment instantly to keep it in the best suitable condition. Thanks to the popularity of technology and human-computer interaction, the greenhouse control system is able to serve its user in the best possible way, however, it does not function independently. The study performed by Wang et al. [73] presented a mechanism based on data communication in order to achieve the data transmission protocol matching and also to provide the control parameters as well as the configuration of the system implemented in a greenhouse.

Liang et al. [74] presented a paper proposing a dynamic method of monitoring the WIFI based greenhouse environment and they realized the remote monitoring of humidity, temperature, and the light intensity of greenhouse through a designed sensor and a server software. In another study by Munoz et al. [75], a cloud-based solution was presented as a service provided to greenhouse models and their proposed system was able to be used as a real-time virtual sensor or as a simulator for research purposes. The use of sensors in greenhouses has been an efficient approach. Garcia-Manas et al. [76] developed a soft sensor used for tomato crop by using dynamic models in order to reproduce biological as well as physical phenomena in the greenhouse. In their study, weather forecasts obtained externally were utilized to estimate the crop growth. Another study conducted on control and monitoring function of greenhouse environment by Wang et al. [77] showed that Web application can configure the information adaptively in accordance with the control and monitoring parameters.

2.4. SYSTEM MAIN COMPONENTS

2.4.1. Microcontrollers

Microcontrollers are the heart of machine learning systems, where all operations are performed according to the selected algorithm. Several advantages arise from using microcontrollers, including space and power savings, cost saving, ability to operate on a stand-alone basis without the need to connect to a network or a power source, and the ability to process data locally without relying on a cloud source. There are several microcontrollers that can perform several tasks according to their capabilities [78]. Table 2.1 shown examples of different microcontrollers and their specifications.

Table 2.1. Examples of common microcontrollers used for machine learning applications.

Name	Illustration	Features and Specs
Coral Dev Board		A single-board computer with a removable system-on-module (SOM) that contains eMMC, SOC, wireless radios, and Google’s Edge TPU. RAM: 1 GB LPDDR4
NVIDIA		Allow multi stream video analytics and can process up to 8 HD full motion video streams in real time. RAM 4 GB 64-bit LPDDR4 25.6 GB/s
Sipeed MAIX GO Suit		It has high performance compared to its space and power footprint with the ability of delivering very accurate results. RAM: 8 MB high speed SRAM 400 MHz.

Raspberry Pi 4		One of the MCs that offer high processing speed, increased memory, and multimedia compatibility. RAM: 1 to 4 GB LPDDR4
Arduino Uno		It is effective in connectivity with several hardware types, which facilitates good environment creation. It has several input and output pins allowing a wide range of hardware to be integrated. RAM: 2 KB SRAM

2.4.2. Sensors

There are several sensors that can be used in automated greenhouse systems. The main sensor types are used for temperature, humidity, carbon dioxide, light, and soil moisture. Different sensor types are shown in Figure 2.13.

Figure 2.13. Different sensor types used in greenhouse systems.

Temperature, lighting, carbon dioxide, soil moisture, soil pH, wind, and precipitation Temperature sensors contain thermistor that are installed depending on the temperature range in the greenhouse environment, while the ideal sensor types contain wireless features to communicate with microcontrollers. Humidity sensors need to be waterproof and have dust protection features in order to keep their functionality effective. Carbon dioxide sensors depend on programmed tolerances to send notifications to the automation system, while lighting sensors use spectrums to identify the current lighting conditions within the greenhouse environment. There are other types of sensors that can detect pH levels, wind speed, and precipitation. All sensor components depend mainly on the complexity and functionality of the system, and according to the needs of the greenhouse and its crops [79].

PART 3

MATERIALS AND METHODS

3.1. MATERIALS

3.1.1. Image Processing Systems

An image processing system has three essential components: camera, computer, and controlling system. The camera is the image acquisition tool that is used to feed the system with the current situation of an object, where there are two main types: IP cameras and machine vision cameras. The type of camera used in the system depends mainly on the minimum quality of pictures that need to be fed for the machine learning algorithm to work. Thus, the camera type used in the system is a designer's choice, as a VGA camera can be sufficient for some applications or high resolutions might be required to identify more specific characteristics of the object. The image processing system may or may not have the controlling system integrated into it, depending on its purpose and final design intent. However, a computer or a microprocessor is essential to process, enhance, and analyze image obtained from the camera.

There are different microprocessors that can be used for image processing. Hercik et al. [80] used a small low-power microprocessor (DSP 56F805, Figure 3.1) for traffic image processing application and succeeded into analyzing images in 100 ms, while no indication of system accuracy was performed. The used microcontroller is energy efficient and has 16-bit processor with 80 MHz core frequency. It has very flexible configuration capabilities that allows it to handle a maximum of 6 operations per instruction cycle. Its use in image processing application is popular, as well as other applications, including automation, power control, metering, engine management, and motion control [81].

Figure 3.1. Freescale DSP 56F805 microprocessor.

Nvidia is another microprocessor type that is used for image processing applications, Figure 3.2. Saha et al. [82] used a Nvidia GPU and succeeded in reaching a 3000 x 3000 image resolution and attributed their success in achieving a recursive ray tracing that requires a high computational algorithm to using this type of microprocessor. Mounir et al. [83] tested the most common image processing algorithms in different microprocessors. They compared the results from NVIDIA GPUs to Raspberry Pi, CPUs, and FPGAs. The Nvidia microprocessor achieved an acceptable execution time (1.79s) compared to Raspberry Pi 3 (13.05s), while FGPAs had the lowest execution time (0.09s). Vasan et al. [84] achieved significant enhancement in image processing systems in terms of timing, which made them suggest Nvidia as one of the key microprocessors that can be used in image processing systems. Nvidia GeForce GTX 400 GPU is one of the common GPUs that has a high-speed processing and made also for video image processing. It has a 128-bit HDR range that supports interactive ray tracing and video transcoding. Its system supports a wide range of application programming interfaces, including Python, Java, C++, and Fortran. It also supports resolutions for up to 2560 x 1600 according to the manufacturer [85].

Figure 3.2. NVIDIA graphic processing unit (GPU).

Raspberry Pi is another type that is mainly used in image detection applications and provides adequate results. Hrbcek and Bubenikova [86] used Raspberry Pi for image processing and confirmed the ability of the microprocessor to achieve detection against a database library. Sokullu et al. [87] designed a mailbox face detection system using Raspberry Pi. Pixia and Xiangdong [88] succeeded in detecting disease in cucumber crops using the microprocessor and the pigmentation discoloring trait of the disease with an accuracy rate reaching to 96%. Therefore, Raspberry Pi is one of the common microprocessors for detection applications and it has proven good accuracy in this application. Raspberry Pi 4 model B (Figure 3.3) is one of the latest microprocessors that has high RAM capacity with good I/O, GPU, and CPU performance. It has a 64-bit 1.5 GHz processing capability, supports various hardware compatibility, and has the ability to provide good image processing results [89].

Figure 3.3. Raspberry Pi 4 Model B.

3.1.2. Control Systems

Agriculture control systems depend mainly on microcontrollers as the heart of the system, then integrated with sensor modules and control modules. Arduino microcontrollers are among the most preferred choices for different application, especially for automation in agriculture. It is one of the most flexible boards out of the available range, as well as its ability to provide connectivity to several component types and being relatively easy to program. Arduino Uno (Figure 3.4) has a power jack, USB connection port, ICSP header, 16 MHz crystal oscillator, six analog inputs, 6 PWM outputs, and a button for reset. The microcontroller can serve as a stand-alone controlling device that does not require a connection to a computer. There are two options for powering Arduino: external power and USB connection, with a minimum supply requirement of 6 volts and a maximum of 12 volts. The software programming is performed on the board without the need for an external computer use [90].

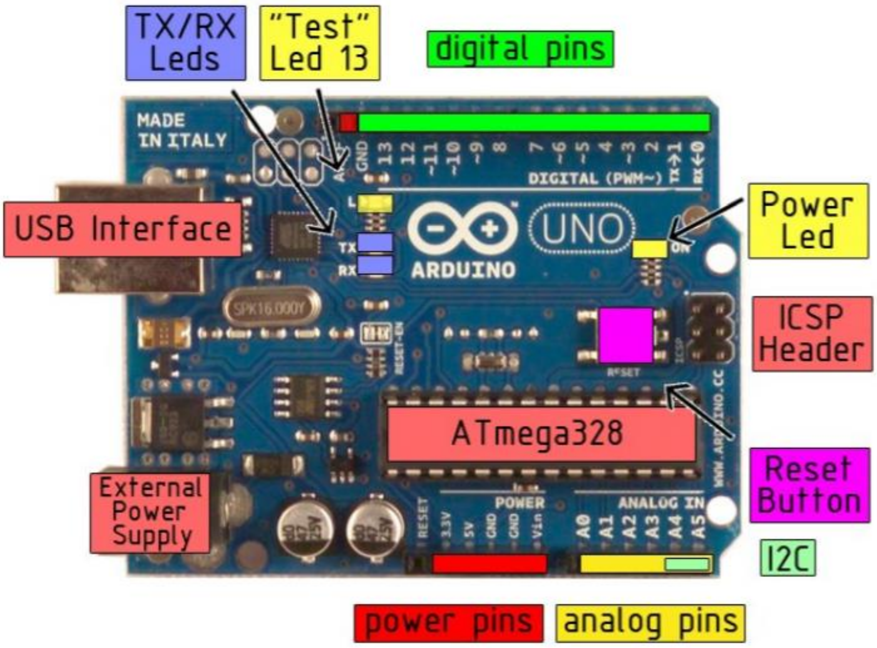


Figure 3.4. Board components of Arduino Uno microcontroller.

Several studies on automated agriculture systems used Arduino as a microcontroller. Hasiri et al. [91] developed an automated sprinkler irrigation system using Arduino. The authors divided the simulation field into blocks with soil moisture sensor, humidity sensors, and rain sensors. The microcontroller was also connected to a water pump that feeds sprinklers in each field segment. The system was connected through ZigBee to a smart phone application for monitoring and control. Bolu et al. [92] developed a similar system that is powered with solar power and regulated electric supply for the functionality of the system. Sruthi et al. [93] added a fertilization automation system to the irrigation system through a pH sensor and a fertilizer pump, as shown in the block map of the design in Figure 3.5. The authors used an LCD display without the addition of a ZigBee protocol to the design.

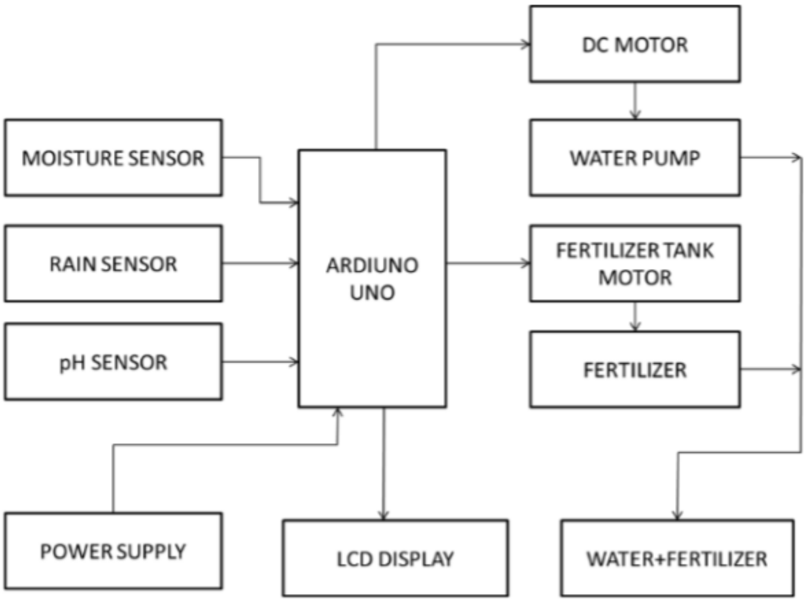


Figure 3.5. A block diagram for automated irrigation and fertilization control system [93].

3.2. METHODS

3.2.1. Image Processing

Image processing systems are built for two purposes: inspection and control, which their outputs are used for the decision-making process. The two types are performed through an image acquired and analyzed against a pre-defined database. For both types, the acquired image is classified into categories that enable further action by a processor or a control system [94]. There are four steps that are used for image processing systems, as illustrated in Figure 3.6:

- Acquisition: an image of the object is taken and translated into a digital format.
- Processing: the image is enhanced and modified to highlight characteristics that are looked for by the system.

- Extraction: the processed image is analyzed to identify the classification of the case of the object and empowers machine learning algorithm to take the necessary action.
- Control: a further step in case of a control system integration, where the object is passed through the system, or a calculated action is taken to correct its condition.

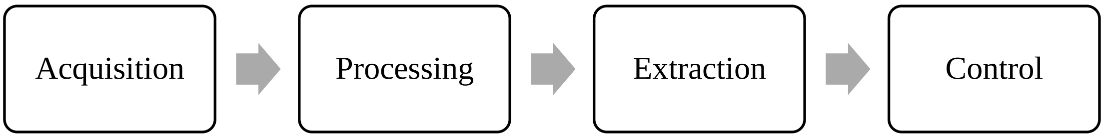


Figure 3.6. Method flowchart for image processing systems [94].

The processing step is the most crucial part of the method as it determines the adequacy of the images taken by the camera, as well as the ability of the algorithm to produce an accurate output. Therefore, the processing part of the algorithm is required to remove unnecessary information and noise from the image, enhance its quality, convert it into a required form, and analyze its elements to identify the targeted issues [94], as illustrated in Figure 3.7.

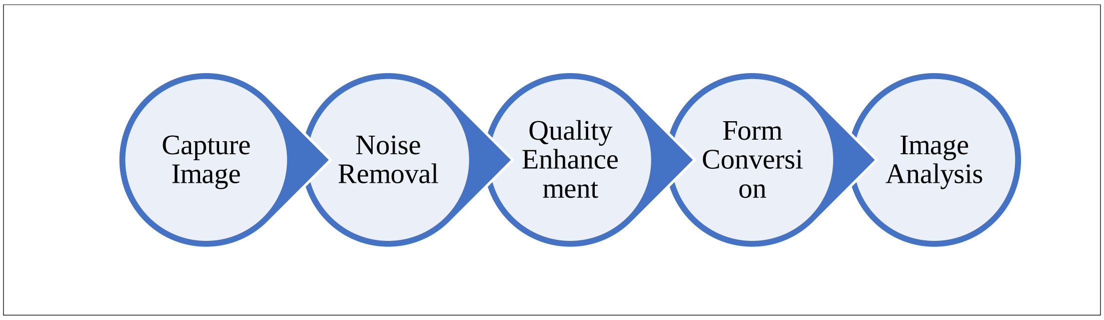


Figure 3.7. Processing flowchart [94].

In agricultural applications, the main characteristic targeted by image processing systems is color. The following action to the system depends on its purpose, whether it was for identification of growth, disease, environmental conditions, pest control, or production volumes [95]. Figure 3.8 shows the image processing flowchart for agricultural systems.

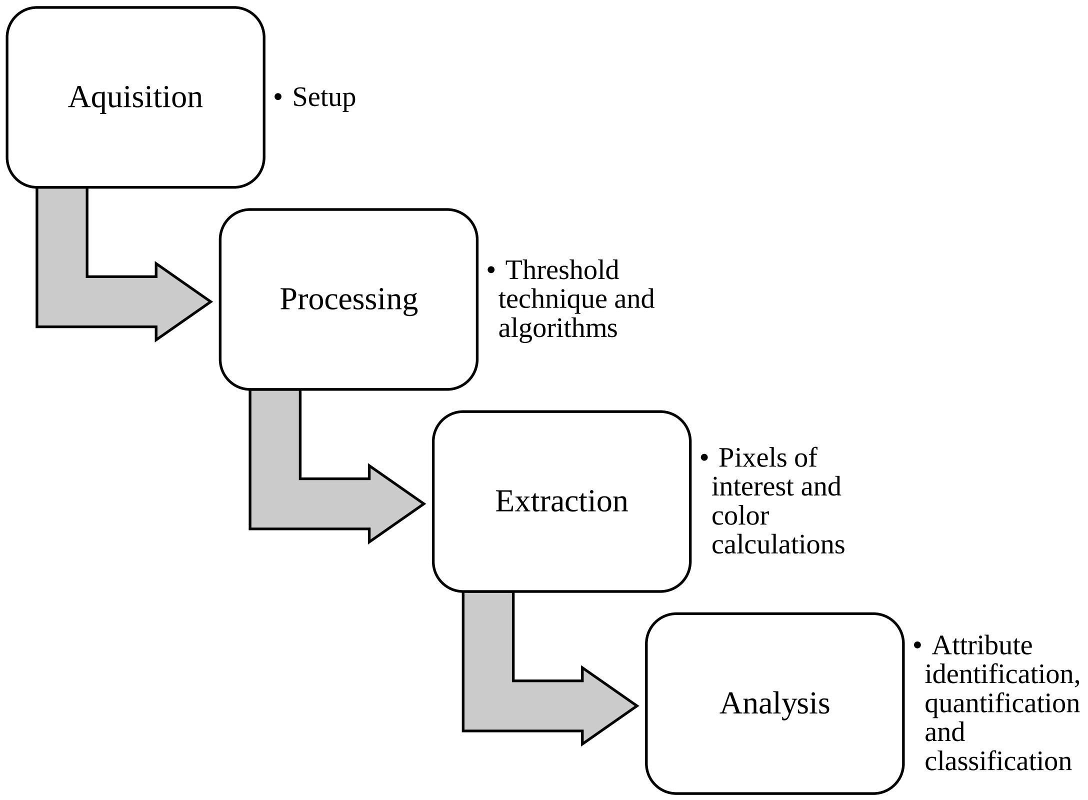


Figure 3.8. Image processing flowchart for agricultural systems [95].

The processing of the image depends mainly on the ability of the algorithm to manipulate the acquired images and extract useful information from them. A key step is identifying the region of interest and separating it from the background. Issues can be identified through understanding their nature is targeted: colorimetric, texture, or morphological. The algorithm is setup to identify the different elements of the image, such as its location and orientation, on a three-dimensional array, while the colors are classified on an RGB system [95], as shown in Figure 3.9.

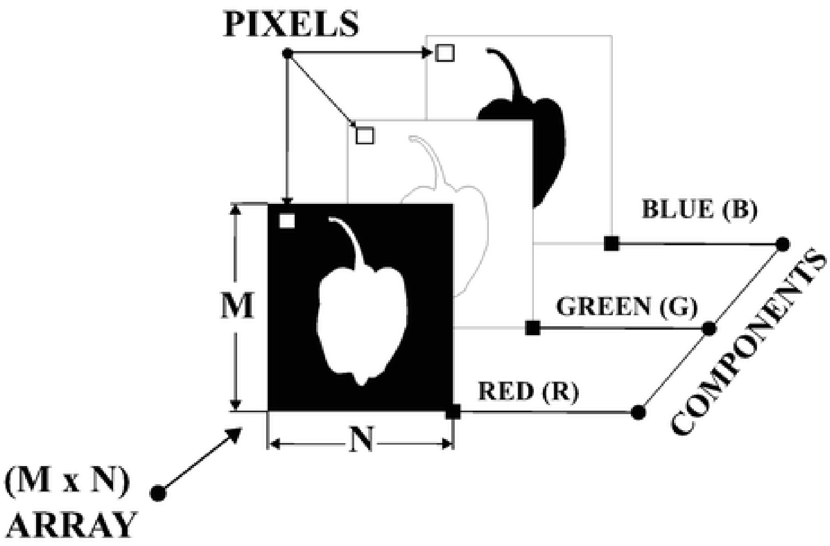


Figure 3.9. RGB components extraction [95].

Thereafter, the image is divided into two main zones and each one of them are filled with a different coding using an algorithm to achieve color segmentation. Color calculation is performed on the zone of interest in the image and attributes are extracted through three types: lightness (L), chroma (C_{ab}), and hue (h_{ab}). The algorithm matches these attributes through machine learning to the database to identify the case of the image [95].

3.2.2. Control System

In agriculture, control systems vary depending on the volume and sophistication of operations that are required to be performed. The system can be a fixed system or a robotic system for irrigation, harvesting, seeding, weeding, or tracking. In the previously mentioned study of Sruthi et al. [93], the system had no ZigBee communication and it was meant to provide readings through an LCD display screen, while another systems in the literature developed it to be monitored and controlled through a mobile application [91]. Figure 3.10 shows the logic of the control system developed by Sruthi et al. [93], which was intended for irrigation and fertilization control. The moisture sensor took periodic readings, and the programmed system had a minimum value to identify need for irrigation. If the minimum value mark was crossed, the irrigation system is activated to provide the suitable amount of water to the soil. Moisture was measured again to provide feedback to the system. The fertilizer system worked in a similar manner, only measuring for pH level instead of moisture. A predefined pH level activates the system to supply or hinder fertilizer to the soil, accordingly.

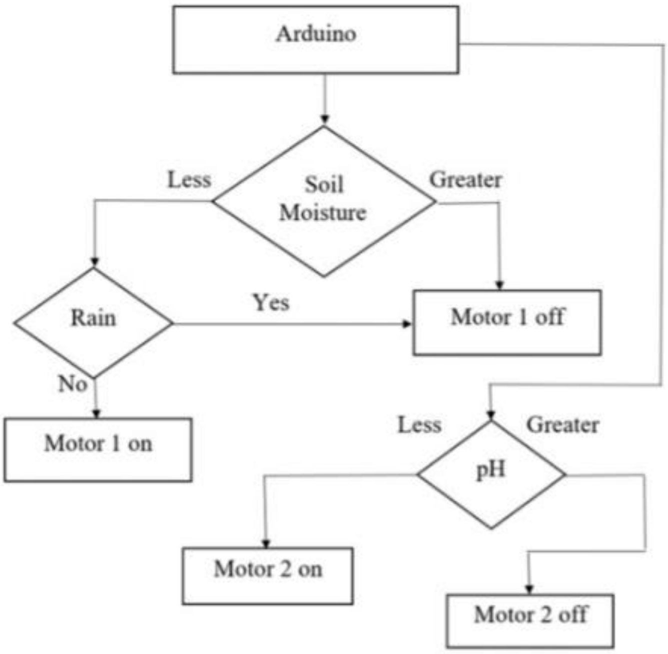


Figure 3.10. Software flowchart for control system [95].

PART 4

SYSTEM DESIGN AND RESULTS

4.1. SYSTEM DESIGN

The designed system is formed by two main modules: the monitoring system using image processing technique, and the control system for vital environment control through sensor techniques. The following sections discuss the design of each module in terms of their components, integration, mechanisms, and expected outcomes.

4.1.1. Monitoring System

The system for monitoring is an image processing assembly that is formed by the image acquisition components and the machine learning algorithm that performs judgements based on database patterns. The hardware of the system contains two main components, as shown in Figure 4.1:

- A microprocessor: Raspberry Pi 4 Model B
- A camera: Raspberry Camera Rev 1.0
- Power supply through the USB port

The Raspberry camera is connected to the CSI port in order to feed the Raspberry Pi 4 Model B microprocessor with images from the greenhouse simulator. The camera has a still resolution of 5 Megapixels and have video modes reaching up to 1080p30. Sensor on the camera is OmniVision OV5647 and images have a pixel size of 1.4 x 1.4 microns. The camera supports different picture formats, including RGB888, YUV420, PNG, BMP, and JPEG. Several features are added to the camera, such as OpenCV integration, motion detection, DRC and HDR customizations, and user-definability image effect.

It has obvious compatibility with Raspberry microprocessors and all camera tuning aspects can be altered, which provides high flexibility in image processing applications.

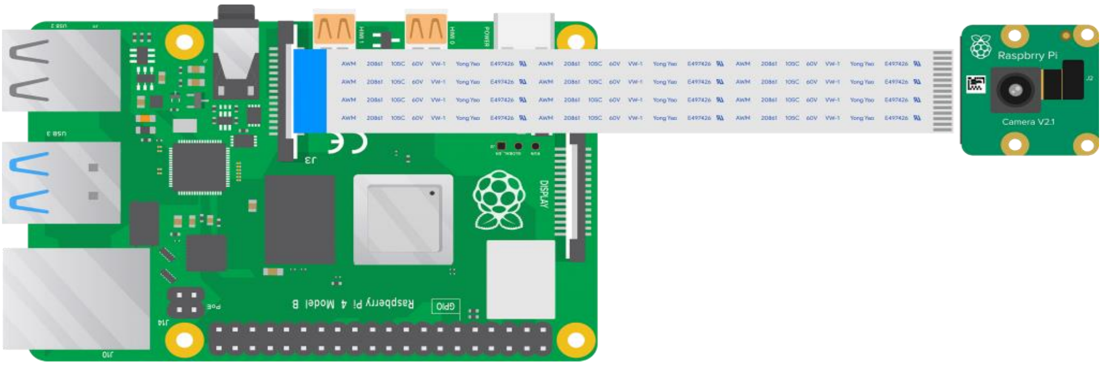


Figure 4.1. Monitoring system schematic setup.

A machine learning algorithm is designed to identify the compliance of plant samples based on their color compatibility. Coding is performed on Python3 and the code with the used algorithm is provided in appendix A. A database of 607 samples was used to train and test the database with a CNN technique using the following steps:

- Creation of convolutional filter through feature maps with equation (7):

$$(f * g) \overset{def}{=} \int_{-\infty}^{\infty} f(\tau) g(t-\tau) d\tau \qquad (4.1)$$

- Add Rectified Linear Unit layer (ReLU) with the function shown in equation (8):

$$\phi(x) = \begin{cases} 0 & for \quad x<0 \\ x & for \quad x \geq 0 \end{cases} \qquad (4.2)$$

- Use max pooling to create pooled feature maps
- Creation of flattened layer
- Merge to create connection

- Output layer and class recognition
- Use Sigmoid function (9):

$$S(x) = \frac{1}{1+e^{-x}} \tag{4.3}$$

- Distances from true values are measured using crossed-entropy function shown in equation (10):

$$H(p,q) = -\sum_{x} p(x)\log q(x) \tag{4.4}$$

The algorithm and results on the accuracy of fault detection was calculated. Testing the code starts of by calling Keras model for image processing from the greenhouse environment, which filters database pictures for unusable images. A final data set is generated for training and testing the algorithm, using the following Keras code:

```
tf.keras.preprocessing.image_dataset_from_directory(
    directory,
    labels="inferred",
    label_mode="int",
    class_names=None,
    color_mode="rgb",
    batch_size=32,
    image_size=(64, 64),
    shuffle=True,
    seed=None,
    validation_split=None,
    subset=None,
    interpolation="bilinear",
    follow_links=False,
    crop_to_aspect_ratio=False,
    **kwargs)
```

The used machine learning algorithm for greenhouse samples takes the following steps:

- Reading the acquired image.
- Resizing the image into 64 x 64 pixels.

- Filter image coloring according to RGB system.
- Saving image for 32-bit compatibility single precision float.
- Perform prediction based on CNN algorithm
- Classify image using a 1 or 0 system into “wanted” and “unwanted”

Finalized images are divided through the algorithm into smaller sections for pattern comparison with the training set. Through this process, the code was able incompatibilities in other characteristics, besides color, such as shapes, angles, and orientations in order to help the machine learning algorithm identify unconformities. Each pixel is then assigned to a number (either 1 or 0). If more than 95% of the pixels were assigned 1 (compatible), the whole image was deemed compatible. A sample of daisy flower images that were used to train the algorithm are shown in Figure 4.2.

Figure 4.2. Sample from the database set used to train and test algorithm.

4.1.2. Control System

An Arduino UNO microcontroller is used as the heart of an agricultural control system, which is designed to control the environment within the greenhouse through vital indicators and intervention measures, as shown in the schematic in Figure 4.3. Three sensor types are used for the system:

- Temperature and humidity sensor (DHT11): the sensor carries the function of measuring the temperature and humidity inside the greenhouse and provides detailed information regarding the inside environment.
- Rain sensor: the main function of this sensor is to detect water drops hitting the ceiling of the greenhouse.
- Soil moisture sensor: In agricultural mediums, a minimum and maximum level of moisture is permitted based on the crop type and growing stage. Therefore, this sensor provides this information to the microcontroller for further action.

Based on the sensor types that are used in the design, it is evident that the main controlled parameters are temperature, humidity, and soil moisture. Lighting is also controlled by the system through simulating daylight. Subsequently, these parameters are controlled using:

- Louvers that open to the side and at the top of the greenhouse for ventilation, reduction of temperature, when needed, connected to the system with a servo motor.
- Infrared lamp to heat air within greenhouse if temperature drops below the minimum limit.
- Fans that circulate air and their speed is controlled through an L298N (U1) controller, which supplies the fans with power according to the needed speed.
- A Spectrum led that generates UV lights with 220-volt power to simulate sunlight. Since the greenhouse is not exposed to sunlight, the microcontroller is programmed to provide a steady rate of UV rays for certain hours per day.
- A water pump with a reservoir to supply soil with water as needed by the system.

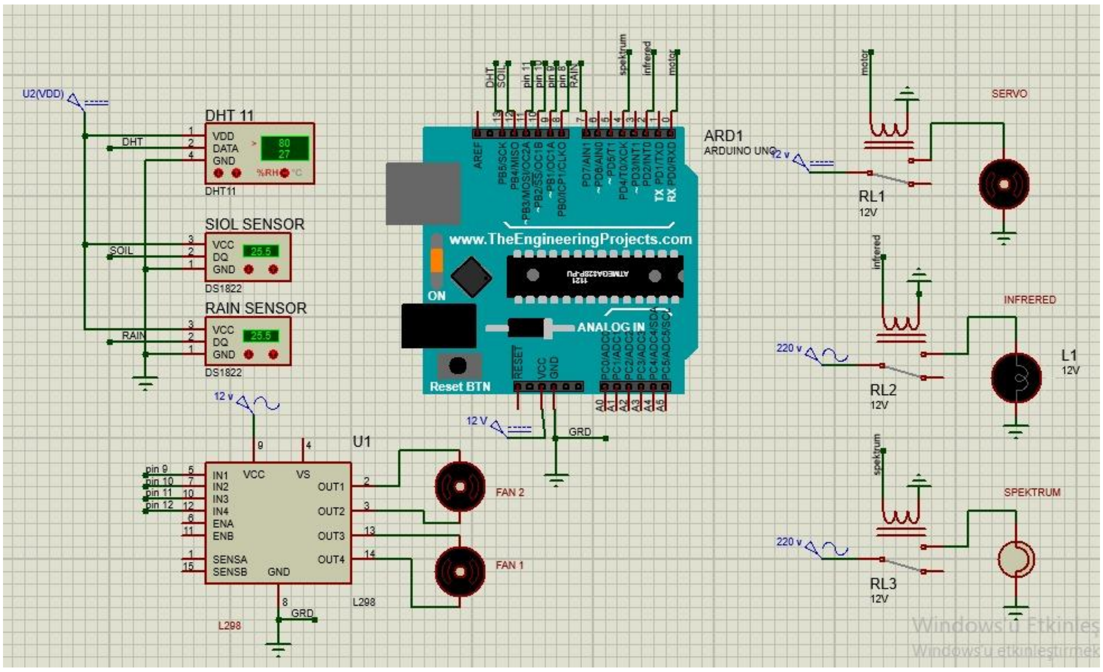
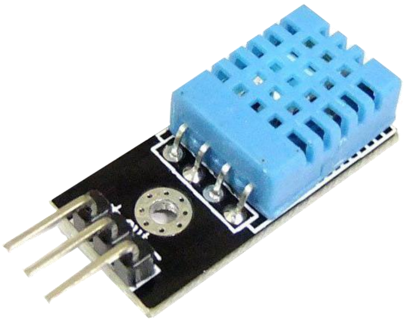
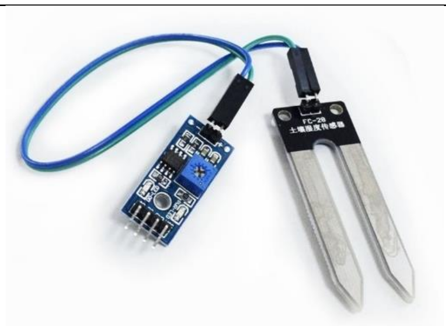
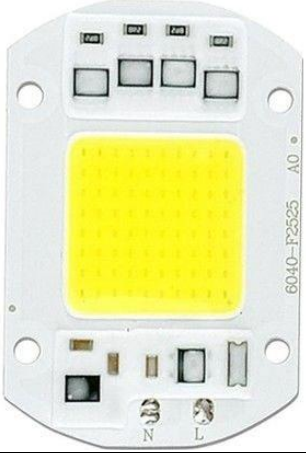


Figure 4.3. Schematic diagram of the designed greenhouse control system.

Each component within the system is connected to switch relays for control by the microcontroller and to earthing for safety. Sensors are supplied with power with U2 (VDD) low voltage source (5 volts). The Arduino microcontroller, the servo motor, and the U1 controller are fed with 12 volts, while the infrared lamp and spectrum led lamp are fed with 220 volts. Table 4.1 provides illustrations and specifications for the components of the control system.

Table 4.1. Specifications for components used in greenhouse control system.

Component and model	Illustration	Specifications
DHT11 Guangzhou Aosong Electronics Co.		Voltage: 3.5 – 5.5 V Current: 0.3 mA Temp. range: 0 – 50 °C Humidity range: 20% - 90% Resolution: 16-bit Accuracy: ±1
Rain sensor FC-37		Electronic and collector boards Voltage: 5V Current: 15 mA
Soil moisture sensor FC-28		Voltage: 3.5 – 5 V
Infrared lamp Haining Xushi R30/ R95		Voltage: 220V Watt: 150 W Size: 95 x 128 mm

Servo motor Tower Pro SG90		Voltage: 4.8 – 6 V Turn: 90° in each direction
Fans		Voltage: 12 V Current: 0.07 A
L298N Dual H Bridge Driver		Voltage: 46 V Current: 2 A Logical voltage: 5 – 35 V Logical current: 0 – 36 mA
PowerLed COB LED Grade 50		Energy efficiency: A++ Color temperature: 5000 K Power: 100 W Voltage: 220 V Current: 1500 mA

Water pump		Voltage: 2.5 - 6 V Current: 220 mA Capacity: 120 L/hr
Ningbo Songle Switch Relays		Nominal voltage: 12V Nominal current: 37.5 mA Pull-in voltage: max 75% Drop out voltage: min 10% Max voltage: 110%

The control of the system is performed through an Arduino UNO code provided in Appendix B. The code provides the control of all sensor and intervention components connected to the microcontroller. Feedback is sent by the microcontroller to an android application, as shown in Figure 4.4. The modes of control are implemented into the code; however, the mobile application allows the operator to monitor the conditions of the greenhouse and overwrite changes as judged suitable. The final greenhouse setup is presented in Figure 4.5.

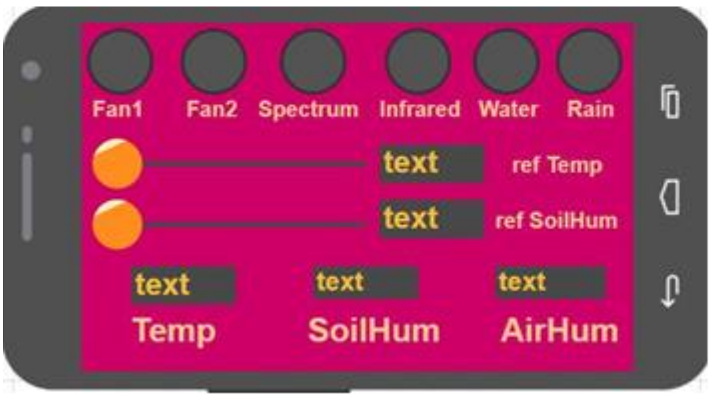


Figure 4.4. Interface screen of greenhouse control system.

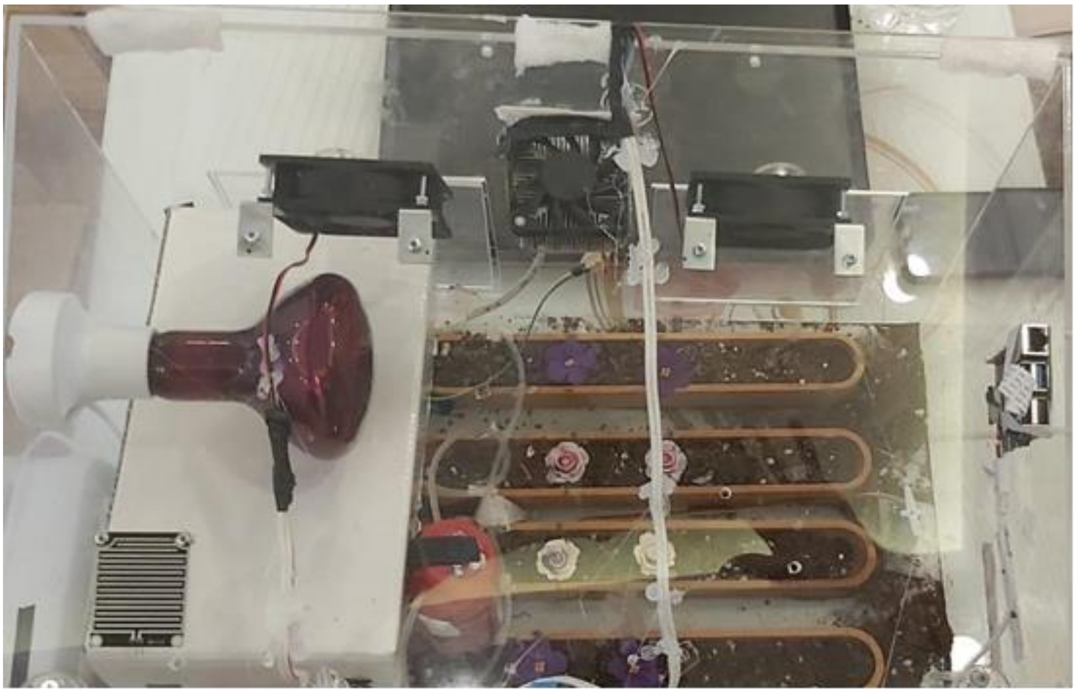


Figure 4.5. Greenhouse simulation setup.

4.2. RESULTS

4.2.1. Accuracy and Precision of Monitoring System

The monitoring system depends on an image processing technique through a machine learning code. Thus, the database of pictures is divided into a 60% training group and a 40% testing group. The accuracy of the prediction is calculated using equation (7):

$$Accuracy = \frac{|TU| + |TW|}{|FU| + |FW| + |TU| + |TW|} \tag{4.5}$$

Where,

False wanted (FW): samples from the wanted class predicted as unwanted.
False unwanted (FU): samples predicted as unwanted with true class wanted.
True wanted (TW): samples predicted correctly from the wanted class.
Ture unwanted (TU): samples predicted correctly from the unwanted class.

A total of 772 daisy flower samples were used, where the set was divided into three train-test splits: 60-40, 70-30, and 80-20. The maximum epoch number was set to 5 and the model accuracy reached for the three splits were 96.32%, 98.43%, and 99.14%, respectively (Figure 4.6). The model losses were recorded at epoch 5 for the three models as 0.153, 0.072, and 0.056, respectively.

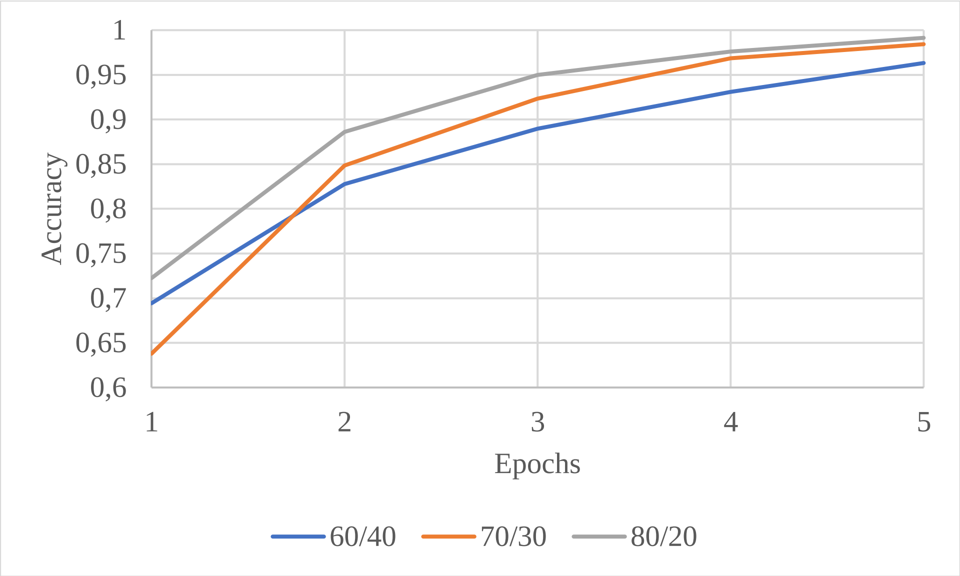


Figure 4.6. Machine learning model accuracy for daisy flowers.

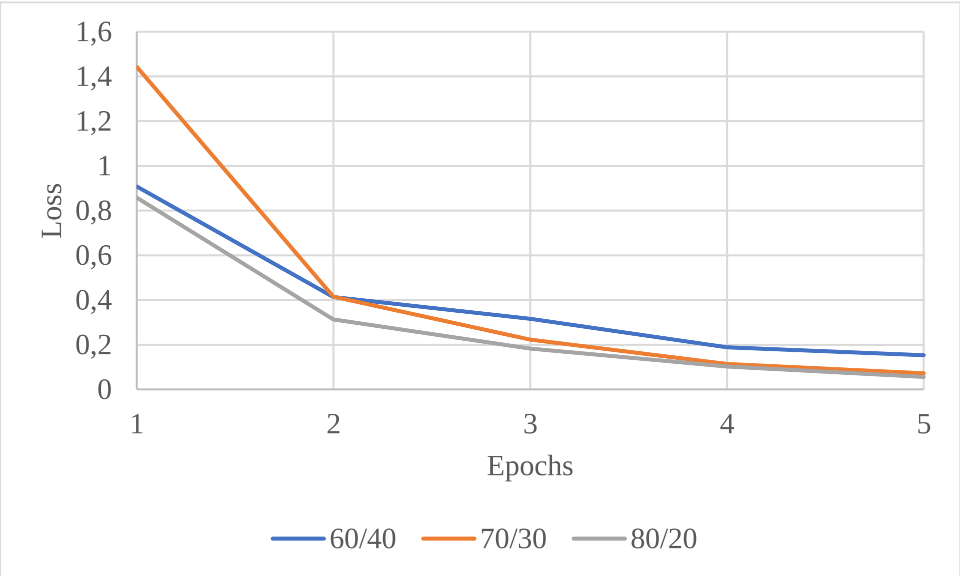


Figure 4.7. Machine learning model loss for daisy flowers.

4.2.1. Performance of Control System

The control system functionality and performance were tested through four modes of change forced through manually manipulating the values of the indicators on the mobile application. The control system has a manual and automatic modes, where in the latter values were fixed, as shown in Table 4.3. The four testing modes are:

- Mode 1: increasing the temperature in the greenhouse to 30°C. The system responded to user order after switching it to manual mode and the infrared lamp heated the closed greenhouse to the set temperature.
- Mode 2: decreasing the temperature in the greenhouse after switching off the control system by using a block of ice and leaving the environment to cool to 18°C.
- Mode 3: Drying the soil using a hair dryer after switching off the system to reduce the soil moisture level to below 20%.
- Mode 4: Drying greenhouse environment with hot airflow from hair dryer while system is off until humidity is below 20%.

Table 4.2. Set of parameter values built in the control system for the greenhouse.

Parameter	**Set range**
Temperature	22 – 22.5 °C
Humidity	60% - 80%
Soil moisture	85% to 100%

After applying testing mode 1, the temperature inside the greenhouse was recorded as 30.2 °C. The control system was returned back to the automatic mode and started, where the fans started operating and side ventilation louvers were open. The system response was recorded 6 seconds after the change to the automatic mode. As shown in Figure 4.9, the temperature in the greenhouse set up started decreasing in response to testing mode 1 and achieved in the range in 83 seconds.

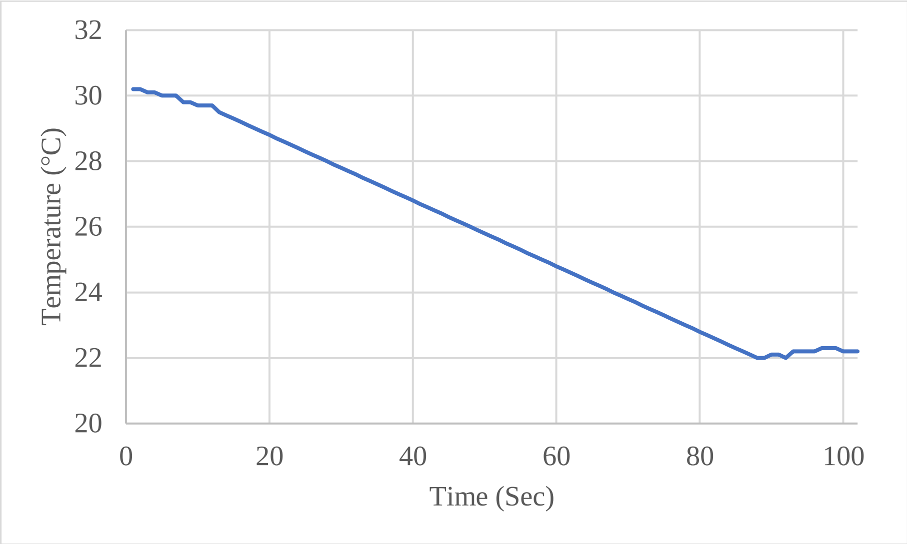


Figure 4.8. Temperature adjustment by control system in response to testing mode 1.

After applying testing mode 2, the temperature inside the greenhouse was recorded as 18.1 °C. The ice block was kept in the greenhouse and the control system was returned back to the automatic mode and started, where the louvers were completely closed, and the infrared lamp started working. The system response was recorded 2 seconds after the change to the automatic mode. As shown in Figure 4.10, the temperature in the greenhouse set up started increasing in response to testing mode 2 and achieved in the range in 57 seconds.

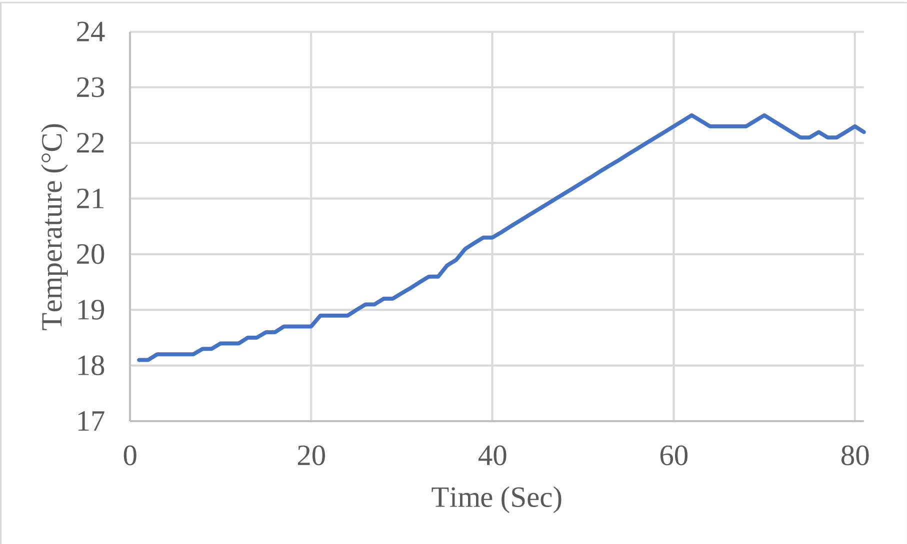


Figure 4.9. Temperature adjustment by control system in response to testing mode 2.

After applying testing mode 3, the soil moisture level was reduced to 19.4% while the control system was set to manual mode. The automatic mode was switched on, where the water pump started working to adjust moisture level after 5 seconds. As shown in Figure 4.11, water started flowing into the planters in the greenhouse, which increased the soil moisture level to the 85% minimum and continued increasing after 30 seconds.

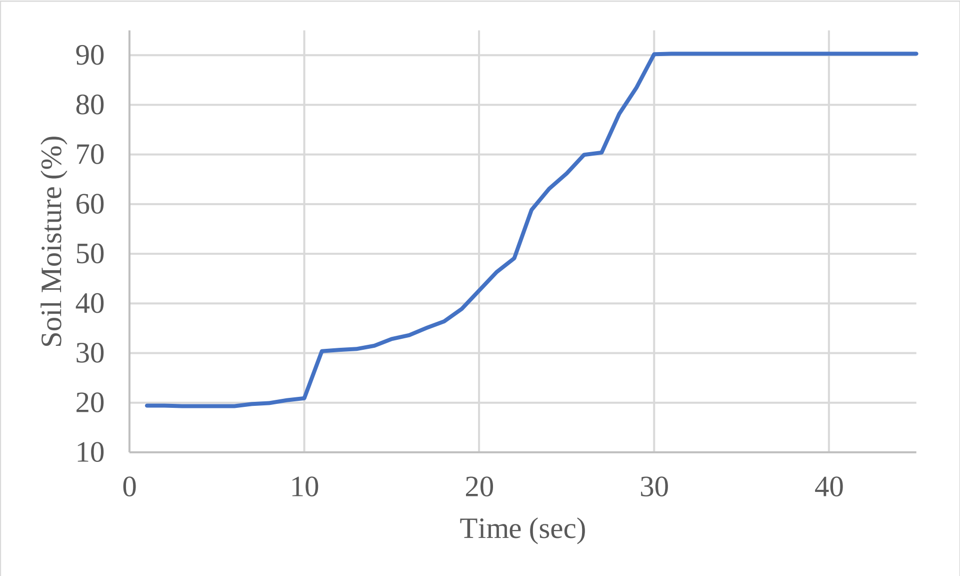


Figure 4.10. Soil moisture adjustment by control system as a response to testing mode 3.

After applying testing mode 4, the humidity inside the greenhouse was reduced to 18.3% while the control system was set to manual mode/ switched off. The automatic mode was switched on, where the water pump started working to remoisturize the soil planters and infrared lamp worked to increase temperature to simulate more water evaporation. The system responded to the initial humidity reduction in 7 seconds after it was switched on. As shown in Figure 4.12, water started flowing into the planters in the greenhouse and the increased heat assisted evaporation, which increased humidity level to the 60% minimum threshold after 326 seconds and continued increasing up to 65%, when system switched to adjust other parameters.

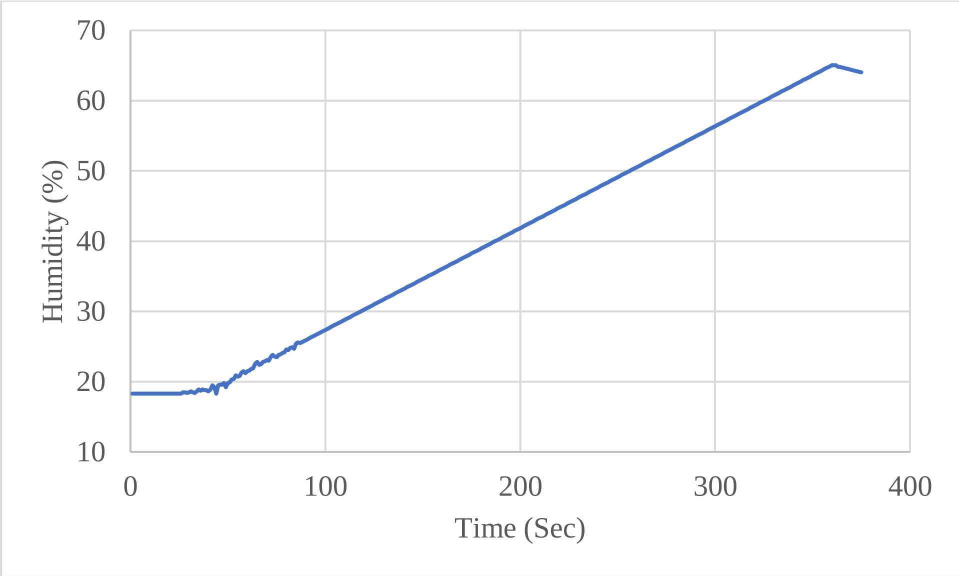


Figure 4.11. Humidity adjustment by control system as a response to testing mode 4.

PART 5

DISCUSSION AND CONCLUSION

5.1. DISCUSSION

The current research designed and implemented a monitoring system and a control system for greenhouse agriculture with the coordination of several complex components. The monitoring system used a Raspberry Pi 4 Model B microprocessor to empower a machine learning algorithm. The accuracy of the system was calculated as 96.32% with a 60% training set, 98.43% with a 70% training set, and 99.14% with a 80% training set. The system was able to identify all samples presented within the greenhouse, which demonstrates the potential of the created solution. The implemented solution identified the healthiness of the plant sample through analyzing pictures acquired through a camera and provided a binary result of the qualification or disqualification of the plant sample based on its appearance. The accuracy rate obtained in both cases exceed the 96% achieved by Pixia and Xiangdong [88] in detecting disease pigmentations in cucumber grown in a greenhouse simulation. Zhou et al. [96] developed a similar system to the one implemented in the current research for the detection of ripening of three species of strawberry. The accuracy measure used the same methods through a training and testing sets, where the highest accuracy was calculated as 89.6%. Table 5.1 shows a comparison of accuracy rates between the monitoring system in the current research and similar systems in the literature.

Table 5.1. Comparison of accuracy for image processing and machine learning studies.

Study	Technique	Best Accuracy
Current Monitoring System	CNN Technique	99.14%
Pixia and Xiangdong [88]	Supervised learning	96%
Zhou et al. [96]	Unsupervised learning	89.6%

The accuracy of a monitoring system dependent on an machine learning technique is dependent mainly on two factors: the size of the training set and the complexity of the criteria that are identified through the system. Zhang and Xu [97] tested an unsupervised algorithm similar to the one implemented in the current research in enhancing analysis techniques based on an image database. The system was tested on tomato and used ten picture images to analyze the image. The best accuracy value of 94.09% was achieved on one of the images using the machine learning technique. An accuracy reaching to 98.24% was achieved using the same technique on the increase of the size of the training set. Similarly, Xu et al. [98] claimed the achievement of a high accuracy using the same technique that exceeded the achieved 97.4% in the current research. Nonetheless, the authors of the research designed their system to recognize the existence of the crop in a certain position. Thus, the intent of their design was to locate the plant rather than analyze its characteristics. The monitoring system implemented in the current research demonstrated higher complexity through analyzing the RGB patterns on the acquired plant images from the greenhouse, which is considered a more complex task to be fulfilled through CNN machine learning techniques. Therefore, the accuracy level that was achieved by the system can be deemed high in comparison with similar systems and it exceeds the accuracy levels of systems with the same level of complexity.

The control system presented in the current research expanded this approach to a user interface and interventive control system that is able to adjust greenhouse environmental parameters based on ideal set of values. The efficiency of the system was tested through forced testing modes in order to understand the system's response to extreme change conditions. Wudneh and Vanitha [99] developed a monitoring system with the use of Raspberry Pi 3 as a microcontroller. The system had an architecture that included a monitoring camera, soil humidity sensor, and a DHT11 sensor. The study included graphs of temperature changes and soil moisture changes over a twenty-minute period. The picture acquisition through the camera did not serve a real function other than providing pictures to the system to save. It is evident that the system was solely designed for monitoring purposes, which makes the control system developed in this research more superior. Lakhiar et al. [100] presented an architecture of a control system that implements IoT for the control irrigation, humidity, temperature, and carbon dioxide levels. The research developed a similar design to the control system presented in the current research. However, the system was not implemented nor tested for performance. Additionally, the proposed architecture did not indicate an automated mode but only a manual control using a mobile application. A similar system was designed by Osama et al. [101], where the system demonstration used LED lights to simulate growth for tomatoes in a greenhouse. Other sensors were added to the system for temperature, humidity, and soil pH level. However, these systems were for monitoring purposes without any intervention frameworks like the one imposed for LED lights. The system was also dependent on user controls without automation.

5.2. CONCLUSION

The problem addressed in the current research is the development of a smart greenhouse monitoring and control systems that enables agricultural applications for image classification using processing techniques, as well as monitor vital parameters within the greenhouse environment with an automated mode with the ability for user intervention. The current systems presented in the literature lack the ability to develop such systems with adequate application that can be used in greenhouse settings. Therefore, the current study aimed for designing a monitoring system for the conditions of plantations in greenhouses using image processing techniques and allowing for controlled conditions based on human judgement through a controlling system with sensory techniques adjusted using an android phone application.

The designs targeted by the current research allows for alarming farmers of their plant conditions, monitoring greenhouse environment, and adjusting this environment to enhance plant conditions. Furthermore, these systems allow the remote interaction with these systems through a mobile application, which facilitates the scalability factor that forms a challenge in agriculture. This means that instead of designating a technician to each greenhouse, a single technician can monitor those systems and assign the necessary actions, which reduces costs, efficiency, and productivity in greenhouses.

A monitoring system was designed through using image processing and machine learning techniques. The system was developed through the use of Raspberry Pi 4 Model B as a microprocessor, a Raspberry Camera rev 1.0, and a machine learning code. The outcomes of image processing were judged on a binary system of 1 = wanted and 0 = unwanted for image classification. The accuracy of the system was tested through two arrangements: 60% training/ 40% testing with an accuracy of 96.32%, 70% training/ 30% testing with an accuracy of 98.43%, and 80% training/ 30% testing with an accuracy of 99.14%. The greenhouse setting was put to test and the system was able to correctly identify all samples according to their true classification. A comparison with the literature showed that the designed system is superior based on the complexity of criteria through an RGB classification.

The control system was designed to monitor vital greenhouse parameters: temperature and humidity, rain, and soil moisture. An Arduino UNO microcontroller was used to control the system. Control components included ventilation fans, infrared lamp, water pump, side and top louvers, and spectrum LED light. A mobile application was designed as a user interface to monitor changes in the environment, in addition to functions of system components in response to changes in the vital greenhouse parameters. A simulation of the setup was performed through forced testing modes: increase in temperature (mode 1), decrease in temperature (mode 2), decrease in soil moisture level (mode 3), decrease in humidity (mode 4). Response to mode 1 decreased temperature in greenhouse from 30.2 °C to the allowable range of 22 – 22.5 °C in 83 seconds through activating the fans and opening the side louvers. System responded to the change in 6 seconds. Response to mode 2 increased temperature in greenhouse from 18.1 °C to the same allowable range in 57 seconds through the activation of the LED lamp. System was activated in 2 seconds.

Response to mode 3 increased soil moisture level from 19.4% to the allowable range of 85% to 100% in 30 seconds. System was activated in 5 seconds. The efficiency of the system is proven through the testing modes and the comparison to similar systems in the literature showed the extensivity and dependability of the designed control system. Response to mode 4 increased humidity from 18.3% to 60% in 326 seconds, while system was activated in 7 seconds.

5.3. RECOMMENDATIONS FOR FUTURE RESEARCH

The current systems can be altered to test its efficiency with different types of crops and plants. Changes in the machine learning code seem to be necessary in order to adjust the system to changes in attributes. The accuracy and precision of the system can be obtained and compared with the results of the research. Moreover, the current designed systems can be enhanced in future research through several additions. It would be significant to integrate the monitoring and control system into one system in order to turn the monitoring system into a sensor that allows for intervention using the control system. However, such an integration requires major development for the monitoring system to communicate with the control system for a suitable intervention through the available components. Actions such as removal of plant, fertilization, and pest control is required to be added to the control system in order for it to become effective. The adequacy of the used microprocessor (Raspberry Pi 4 Model B) and microcontroller (Arduino UNO) need to be re-evaluated in such as system.

REFERENCES

1. Jha, K., Doshi, A., Patel, P., & Shah, M., "A comprehensive review on automation in agriculture using artificial intelligence", ***Artificial Intelligence in Agriculture,*** 2(2): 1-12 (2019).

2. Lunner-Kolstrup, C., Hörndahl, T., & Karttunen, J. P., "Farm operators' experiences of advanced technology and automation in Swedish agriculture: A pilot study", ***Journal of Agromedicine,*** 23(3): 215-226 (2018).

3. Rotz, S., Gravely, E., Mosby, I., Duncan, E., Finnis, E., Horgan, M. & Fraser, E., "Automated pastures and the digital divide: How agricultural technologies are shaping labour and rural communities", ***Journal of Rural Studies,*** 6(8): 112-122 (2019).

4. Talaviya, T., Shah, D., Patel, N., Yagnik, H., & Shah, M., "Implementation of artificial intelligence in agriculture for optimisation of irrigation and application of pesticides and herbicides", ***Artificial Intelligence in Agriculture,*** 4(6): 58-73 (2020).

5. Shamshiri, R., Weltzien, C., Hameed, I. A., J Yule, I., E Grift, T., Balasundram, S. K. & Chowdhary, G., "Research and development in agricultural robotics: A perspective of digital farming", ***Int J Agric & Biol Eng***, 11(4): 312-318 (2018).

6. Tian, H., Wang, T., Liu, Y., Qiao, X., & Li, Y., "Computer vision technology in agricultural automation A review", ***Information Processing in Agriculture,*** 7(1): 1-19 (2020).

7. Shafi, U., Mumtaz, R., García-Nieto, J., Hassan, S. A., Zaidi, S. A. R., & Iqbal, N., "Precision agriculture techniques and practices: From considerations to applications", ***Sensors,*** 19(17): 3796 (2019).

8. Rasti, S., Bleakley, C. J., Holden, N. M., Whetton, R., Langton, D., & O'Hare, G., "A survey of high resolution image processing techniques for cereal crop growth monitoring", ***Information Processing in Agriculture,*** Germany, 12-19 (2021).

9. Syeda, I. H., Alam, M. M., Illahi, U., & Su'ud, M. M., "Advance control strategies using image processing, UAV and AI in agriculture: a review", ***World Journal of Engineering,*** 18(4): 579-589 (2021).

10. Vibhute, A., & Bodhe, S. K., "Applications of image processing in agriculture: a survey", ***International Journal of Computer Applications,*** *52*(2):417-447 (2012).

11. Pandey, S. & Pandey, A., "Greenhouse Technology," ***International Journal of Research***, 3(9): 218-229 (2015).

12. Harjunowibowo, D. E. W. A. N. T. O., Ding, Y., Omer, S., & Riffat, S., “Recent active technologies of greenhouse systems—a comprehensive review”, ***Bulg J Agric Sci,*** 24(1): 158-170 (2018).

13. Strickler, J., "High-Tech Greenhouses Could Be The Future Of Agriculture," **https://www.forbes.com/sites/jordanstrickler/2020/08/28/high-tech greenhouses-could-be-the-future-of-agriculture/?sh=6d68fd6e380f**/(2021).

14. Pennisi, G., & Orsini, F., “Improving light management in lettuce crops”, ***Nature Food,*** 2(6): 394-395 (2021).

15. Atılgan, A., Erdal, İ., & Aktaş, H., “Effects of The Use of Colored Cover Materials and Led Lighting in Greenhouses on Plant Nutrient Concentration: Case of Tomato Plant (Solanum lycopersicum L.)”, ***Turkish Journal of Agriculture-Food Science and Technology,*** 8(12): 2550-2555 (2020).

16. Shang, Y., & Fu, A. Y., “Research on Intelligent Agriculture Greenhouses Based on Internet of Things Technology”, ***In ITM Web of Conferences,*** China, 34-49 (2017).

17. Nesteruk, S., Shadrin, D., Pukalchik, M., Somov, A., Zeidler, C., Zabel, P., & Schubert, D., “Image compression and plants classification using machine learning in controlled-environment agriculture: Antarctic station use case”, ***IEEE Sensors Journal,*** 21(16): 676-689 (2021).

18. Khan, A. I. and Al-Habsi, S. "Machine learning in Computer Vision," ***Procedia Computer Science***, 16(7): 1444-1451 (2020).

19. Dey, A"Machine Learning Algorithms: A Review," ***International Journal of Computer Science and Information Technologies,*** 7(3): 1174-1179 (2016).

20. Sarker, I. H., “Machine learning: Algorithms, real-world applications and research directions”, ***SN Computer Science,*** 2(3): 1-21 (2021).

21. Somani, P. and Kaur, G. "A review on Supervised Learning Algorithms," ***International Journal of Advanced Science and Technology***, 29(10): 2551-2559 (2020).

22. Internet: Singh, A. & Dani, A., "Universalising Leading Healthcare Budgeting Systems by Applying Supervised Machine Learning Through Integrating Recurrent Neural Networks,"**https://ysjournal.com/universalising-leading-healthcare-budgeting-systems-by-applying-supervised-machine-learning-through-integrating-recurrent-neural-networks/**(2021).

23. Khanum, M., Mahboob, T., Imtiaz, W., Ghafoor, H. A., & Sehar, R., “A survey on unsupervised machine learning algorithms for automation, classification and maintenance”, ***International Journal of Computer Applications,*** 119(13): 145-167 (2015).

24. Bo, L., Ren, X., & Fox, D., “Hierarchical Matching Pursuit for Image Classification: Architecture and Fast Algorithms”, ***In Nips*** 1(2): 6-11 (2011).

25. Wani, M. A., Khoshgoftaar, T. M. & Palade, V., “Deep Learning Applications”, ***Springer,*** New York, 45-67 (2021).

26. Internet: Inwinstack, "Reinforcement Learning," **https://www.inwinstack.com/ blog-en/blog_ai-en/6262**/(2021).

27. Lin, H. & Lucey, S., "Inverse Compositional Spatial Transformer Networks," in ***IEEE Conference on Computer Vision and Pattern Recognition (CVPR),*** Honolulu, USA, 34-56 (2017).

28. Kayva, C., Darsini, P., Padmaja, T. S. & Sankeerthana, R. "Performance Analysis of Different Filters for Digital Image Processing," ***Turkish Journal of Computer and Mathematics Education,*** 12(2): 2572-2576 (2021).

29. Internet: Shamsudin, N. F., "[Lab03] Intensity Transformation and Spatial Filter Part 2", **http://biti3313matlabtutorial. blogspot.com/2012/05/lab03-intensity-transformation-and.html**/ (2021).

30. Hashemi, M., “Enlarging smaller images before inputting into convolutional neural network: zero-padding vs. İnterpolation”, ***Journal of Big Data,*** 6(1): 1-13 (2019).

31. Grag, A., Negi, A. and Chauhan, G. "Analysis of Iterated Affine Transformation Function and Linear Mappingvol. for Content Preservation," ***International Journal of Engineering & Technology,*** 7(19): 50-57 (2018).

32. Huang, H. H., & Yang, J., “Affine-transformation invariant clustering models”, ***Journal of Statistical Distributions and Applications,*** *7*(1):1-24 (2020).

33. De Vries, J., “Object recognition: a shape-based approach using artificial neural networks”, ***Department of Computer Science, University of Utrecht,*** Neitherlnad, 34-51(2006).

34. Choab, N., Allouhi, A., El Maakoul, A., Kousksou, T., Saadeddine, S., & Jamil, A., “Review on greenhouse microclimate and application: Design parameters, thermal modeling and simulation, climate controlling technologies”, ***Solar Energy,*** 19(1): 109-137 (2019).

35. Zhang, Y., Henke, M., Li, Y., Yue, X., Xu, D., Liu, X., & Li, T., "High resolution 3D simulation of light climate and thermal performance of a solar greenhouse model under tomato canopy structure", ***Renewable Energy,*** 16(0): 730-745 (2020).

36. Boulard, T., Roy, J. C., Pouillard, J. B., Fatnassi, H., & Grisey, A., "Modelling of micrometeorology, canopy transpiration and photosynthesis in a closed greenhouse using computational fluid dynamics", ***Biosystems Engineering,*** 15(8): 110-133 (2017).

37. Li, K., Sha, Z., Xue, W., Chen, X., Mao, H., & Tan, G., "A fast modeling and optimization scheme for greenhouse environmental system using proper orthogonal decomposition and multi-objective genetic algorithm", ***Computers and Electronics in Agriculture,*** 16(8): 105-196 (2020).

38. Zhao, C. J., Li, M., Yang, X. T., Sun, C. H., Qian, J. P., & Ji, Z. T., "A data-driven model simulating primary infection probabilities of cucumber downy mildew for use in early warning systems in solar greenhouses", ***Computers and Electronics in Agriculture,*** 76(2): 306-315 (2011).

39. Liu, H., Yin, C., Gao, Z., & Hou, L., "Evaluation of cucumber yield, economic benefit and water productivity under different soil matric potentials in solar greenhouses in North China", ***Agricultural Water Management,*** 24(3):106-442 (2021).

40. Wang, L., Qiu, L. & Guo, S. "The Present Situation and Countermeasure Development of Greenhouse in China", ***Journal of Agricultural Mechanization,*** 3(4): 565-6142008.

41. Ahamed, M. S., Guo, H., & Tanino, K., "Development of a thermal model for simulation of supplemental heating requirements in Chinese-style solar greenhouses", ***Computers and Electronics in Agriculture,*** 15(10): 235-244 (2018).

42. Zou, W., Yao, F., Zhang, B., He, C., & Guan, Z., "Verification and predicting temperature and humidity in a solar greenhouse based on convex bidirectional extreme learning machine algorithm", ***Neurocomputing,*** 24(9): 72-85 (2017).

43. Jung, D. H., Kim, H. S., Jhin, C., Kim, H. J., & Park, S. H., "Time-serial analysis of deep neural network models for prediction of climatic conditions inside a greenhouse", ***Computers and Electronics in Agriculture,*** 17(3): 105-402 (2020).

44. Righini, I., Vanthoor, B., Verheul, M. J., Naseer, M., Maessen, H., Persson, T., & Stanghellini, C., "A greenhouse climate-yield model focussing on additional light, heat harvesting and its validation", ***Biosystems Engineering,*** 19(4): 1-15 (2020).

45. Sanchez-Molina, J. A., Ming, L., Rodriguez, F., Guzman, J. L., Hui, W., & Xinting, Y., “Development and test verification of air temperature model for Chinese solar and Spainish Almeria-type greenhouses”, ***International Journal of Agricultural and Biological Engineering,*** 10(4): 66-76 (2017).

46. Rodríguez, F., Berenguel, M., Guzmán, J. L., & Ramírez-Arias, A., “Modeling and control of greenhouse crop growth”, ***Springer International Publishing,*** Switzerland, 250 (2015).

47. Liu, R., Li, M., Guzmán, J. L., & Rodriguez, F., “A fast and practical one-dimensional transient model for greenhouse temperature and humidity”, ***Computers and Electronics in Agriculture,*** 18(6): 106-186 (2021).

48. Ravishankar, E., Booth, R. E., Saravitz, C., Sederoff, H., Ade, H. W., & O’Connor, B. T., “Achieving net zero energy greenhouses by integrating semitransparent organic solar cells”, ***Joule,*** *4*(2): 490-506 (2020).

49. Lijun, C., Shangfeng, D., Meihui, L. & Yaofeng, H. "Adaptive feedback linearization-based predictive control for greenhouse temperature," ***International Agricultural Engineering Journal,*** 27(2): 360-367 (2018).

50. Speetjens, S. L., Stigter, J. D., & Van Straten, G., “Towards an adaptive model for greenhouse control”, ***Computers and Electronics in Agriculture,*** 67(2): 1-8 (2009).

51. Bang, H., Myung, H. S., & Tahk, M. J., “Nonlinear momentum transfer control of spacecraft by feedback linearization”, ***Journal of Spacecraft and Rockets,*** 39(6): 866-873 (2002).

52. Lascu, C., Jafarzadeh, S., Fadali. S. & Blaabjerg, F. "Direct torque control with feedback linearization for ınduction motor drives," ***IEEE Transactions on Power Electronics,*** 32(3): 2072-2080 (2017).

53. Zangina, U., Buyamin, S., Abidin, M. S. Z., & Mahmud, M. S. A., “Agricultural rout planning with variable rate pesticide application in a greenhouse environment”, ***Alexandria Engineering Journal,*** 60(3): 3007-3020 (2021).

54. Bot, G. P., “Greenhouse climate: from physical processes to a dynamic model”, ***Doctoral Dissertation, Bot Universty,*** Nietherland, 67-99 (1983).

55. Van Henten, E. J., Greenhouse climate management: an optimal control approach”, ***Van Henten,*** Neitherland, 67-88 (1994).

56. De Zwart, H. F., Analyzing energy-saving options in greenhouse cultivation using a simulation model”, ***Wageningen University and Research,*** Nietherland, 34-45 (1996).

57. Çakır, U., & Şahin, E., “Using solar greenhouses in cold climates and evaluating optimum type according to sizing, position and location: A case study”, ***Computers and Electronics in Agriculture,*** 11(7): 245-257 (2015).

58. Taki, M., Ajabshirchi, Y., Ranjbar, S. F., Rohani, A., & Matloobi, M., “Heat transfer and MLP neural network models to predict inside environment variables and energy lost in a semi-solar greenhouse”, ***Energy and Buildings,*** 1(10): 314-329 (2016).

59. Mohammadi, B., Ranjbar, S. F., & Ajabshirchi, Y., “Application of dynamic model to predict some inside environment variables in a semi-solar greenhouse”, ***Information Processing in Agriculture,*** 5(2): 279-288 (2018).

60. Taki, M., Mehdizadeh, S. A., Rohani, A., Rahnama, M., & Rahmati-Joneidabad, M., “Applied machine learning in greenhouse simulation; new application and analysis”, ***Information Processing in Agriculture,*** 5(2): 253-268 (2018).

61. He, F., & Ma, C., “Modeling greenhouse air humidity by means of artificial neural network and principal component analysis”, ***Computers and Electronics in Agriculture,*** 7(1): 19-23 (2010).

62. Yu, H., Chen, Y., Hassan, S. G., & Li, D., “Prediction of the temperature in a Chinese solar greenhouse based on LSSVM optimized by improved PSO”, ***Computers and Electronics in Agriculture,*** 12(2): 94-102 (2016).

63. Elsoragaby, S., Yahya, A., Mahadi, M. R., Nawi, N. M., Mairghany, M., Elhassan, S. M. M., & Kheiralla, A. F, “Applying multi-objective genetic algorithm (MOGA) to optimize the energy inputs and greenhouse gas emissions (GHG) in wetland rice production”, ***Energy,*** 6(2): 2988-2998 (2020).

64. Shamshirband, S., Khoshnevisan, B., Yousefi, M., Bolandnazar, E., Anuar, N. B., Wahab, A. W. A., & Khan, S. U. R., “A multi-objective evolutionary algorithm for energy management of agricultural systems a case study in Iran”, ***Renewable and Sustainable Energy Reviews***, 4(4): 457-465 (2015).

65. Elhami, B., Akram, A. and Khanali, M. "Optimization of energy consumption and environmental impacts of chickpea production using data envelopment analysis (DEA) and multi objective genetic algorithm (MOGA) approaches," ***Information Processing in Agriculture***, 3(3): 190-205 (2016).

66. Nabavi-Pelesaraei, A., Hosseinzadeh-Bandbafha, H., Qasemi-Kordkheili, P., Kouchaki-Penchah, H., & Riahi-Dorcheh, F., “Applying optimization techniques to improve of energy efficiency and GHG (greenhouse gas) emissions of wheat production”, ***Energy,*** 10(3): 672-678 (2016).

67. Tabatabaie, S. M. H., Rafiee, S., Keyhani, A., & Heidari, M. D., “Energy use pattern and sensitivity analysis of energy inputs and input costs for pear production in Iran”, ***Renewable Energy,*** 5(1): 7-12 (2013).

68. Nabavi-Pelesaraei, A., Abdi, R., Rafiee, S., & Mobtaker, H. G., “Optimization of energy required and greenhouse gas emissions analysis for orange producers

using data envelopment analysis approach”, ***Journal of Cleaner Production,*** 6(5): 311-317 (2014).

69. Khoshnevisan, B., Rafiee, S., Omid, M., & Mousazadeh, H., “Applying data envelopment analysis approach to improve energy efficiency and reduce GHG (greenhouse gas) emission of wheat production”, ***Energy,*** 5(8): 588-593 (2013).

70. Mousavi-Avval, S. H., Rafiee, S., Jafari, A., & Mohammadi, A., “Optimization of energy consumption for soybean production using Data Envelopment Analysis (DEA) approach”, ***Applied Energy,*** 88(11): 3765-3772 (2011).

71. Xiaohui, Q., Shangfeng, D., Yaofeng, H., & Meihui, L., “Development and design of mobile terminal APP for greenhouse environment control”, ***IFAC-Papers On Line,*** 51(17): 822-825 (2018).

72. Zhao, J. W. Gao, T. & Chan, J. L. "Greenhouse Monitoring System Based on Android Platform," ***Proceedings of the 2018 2nd International Conference on Big Data and Internet of Things,*** Beijing, China, 23-35 (2018).

73. Wang, J., Chen, M., Zhou, J., & Li, P., “Data communication mechanism for greenhouse environment monitoring and control: An agent-based IoT system”, ***Information Processing in Agriculture,*** 7(3): 444-455 (2020).

74. Liang, M. H., He, Y. F., Chen, L. J., & Du, S. F., “Greenhouse Environment dynamic Monitoring system based on WIFI”, ***Ifac-Papersonline,*** 51(17): 736-740 (2018).

75. Muñoz, M., Guzmán, J. L., Sánchez, J. A., Rodríguez, F., & Torres, M., “Greenhouse models as a service (GMaaS) for simulation and control”, ***IFAC-Papers On Line,*** 52(30): 190-195 (2019).

76. Garcia-Mañas, F., Rodríguez, F., & Berenguel, M., “Leaf area index soft sensor for tomato crops in greenhouses”, ***IFAC-Papers On Line,*** 53(2): 15796-15803 (2020).

77. Wang, J., Zhou, J., Gu, R., Chen, M., & Li, P., “Manage system for internet of things of greenhouse based on GWT”, ***Information Processing in Agriculture,*** *5*(2): 269-278 (2018).

78. Swathi, K., Sandeep, U. and Roja, R. A. "Performance Analysis of Microcontrollers Used In Iot Technology," ***International Journal of Scientific Research in Science, Engineering and Technology***, 4(4): 1268-1273 (2018).

79. Patel, P. M. and Chauhan, T. H., "A Survey on various Sensors used in Greenhouse," ***International Journal for Scientific Research & Development***, 1(10): 2082-2084 (2013).

80. Hercik, R., Slaby, R. and Machacek, Z. "Image Processing for Low-power Microcontroller Application," ***Przeglad Elektrotechniczny***, 88(12): 343-346 (2012).

81. Freescale, "56F805 Data Sheet: Preliminary Technical Data," ***Freescale,*** Texas, USA, 34-46 (2007).

82. Saha, M. D., Darji, M. K., Patel, N., & Thakore, D., "Implementation of image enhancement algorithms and recursive ray tracing using CUDA", ***Procedia Computer Science,*** *7(9):*516-524 (2016).

83. Mounir, T. A., Boumerdassi, S. Benhamada, A., Allal, A. M. & Kherarba, M. "Performance evaluation of basic image processing algorithms in CPU, GPU, Raspberry Pi and FPGA," ***International Journal of Computer Science Engineering,*** 9(4): 312-325 (2020).

84. Vasan, S. S., Ionita, C. N., Titus, A. H., Cartwright, A. N., Bednarek, D. R., & Rudin, S., "Graphics processing unit implementation of image processing algorithms to improve system performance of the control acquisition, processing, and image display system of the micro-angiographic fluoroscope", ***International Society for Optics and Photonics,*** 83(13): 83-134 (2012).

85. NVIDIA, "NVIDIA GeForce GTX 400 GPU Datasheet," ***NVIDIA Corporation,*** USA, 45-89 (2010).

86. Hrbček, J., & Bubeníková, E. "Embedded image processing on Raspberry Pi connected to the industrial control system," ***Multidisciplinary Aspects of Production Engineering,*** 2(1): 62-71 (2019).

87. Sokullu, R., Özkan, A. K. I. N., & Aslan, E., "Face Recognition Based Multifunction IoT Smart Mailbox", ***Journal of Emerging Computer Technologies,*** 1(2): 34-37. (2021).

88. Pixia, D., & Xiangdong, W., "Recognition of greenhouse cucumber disease based on image processing technology", ***Open Journal of Applied Sciences,*** 3(1): 27-31 (2013).

89. Raspberry, P., "Datasheet: Raspberry Pi 4 Model B," ***Raspberry Pi (Trading) Ltd,*** Cambridge, UK, 23-56 (2019).

90. Arduino, A.,"Arduino UNO - Datasheet," ***Arduino LLC,*** Boston, 67-101 (2011).

91. Hasiri, E. M. & Asniati, M. A., "The Implementation of Smart Farming Application Based on the Microcontroller and Automatic Sprinkler Irrigation System of Agricultural Land," ***Advances in Science****,* ***Technology and Engineering Systems Journal,*** 5(2): 174-179 (2020).

92. Bolu, C. A., Azeta, J., Alele, F., Daranijo, E. O., Onyeubani, P., & Abioye, A. A., "Solar Powered Microcontroller-based Automated Irrigation System with Moisture Sensors", ***In Journal of Physics: Conference Series,*** 13(3): 320-367 (2019).

93. Sruthi, R., Swytha, S., Varsha, K. & Kavitha, P., "Microcontroller Based Automatic Irrigation and Fertilisation System Using Soil Moisture Sensor and Ph Sensor," ***Journal of Physics: Conference Series (Advances in Computational Electrical Engineering)***, 1(9):64-78 (2021).

94. Eissa, A. A., Khalik, A. A., & Abdel, A. A., "Understanding color image processing by machine vision for biological materials", ***Structure and Function of Food Engineering,*** 2(3): 227-274 (2012).

95. Sandoval, J. R. M., Rosas, M. E. M., Sandoval, E. M., Velasco, M. M. M., & De Ávila, H. C., "Color Analysis and Image Processing Applied in Agriculture", ***Colorimetry and Image Processing,*** 4(5): 71-78 (2018).

96. Zhou, C., Hu, J., Xu, Z., Yue, J., Ye, H., & Yang, G., "A novel greenhouse-based system for the detection and plumpness assessment of strawberry using an improved deep learning technique", ***Frontiers in Plant Science,*** 1(1): 5-59 (2020).

97. Zhang, P., & Xu, L., "Unsupervised segmentation of greenhouse plant images based on statistical method", ***Scientific Reports,*** 8(1): 1-13 (2018).

98. Xu, Z., Shi, X., Ye, H., & Hua, S., "Geometric Positioning and Color Recognition of Greenhouse Electric Work Robot Based on Visual Processing", ***International Journal of Pattern Recognition and Artificial Intelligence***, 35(2): 215-905 (2021).

99. Wudneh, T. B. and Vanitha, V., "Implementation of IoT with ımage processing in greenhouse monitoring system," ***International Journal of Innovative Technology and Exploring Engineering,*** 8(9): 509-514 (2019).

100. Lakhiar, I. A., Jianmin, G., Syed, T. N., Chandio, F. A., Buttar, N. A., & Qureshi, W. A., "Monitoring and control systems in agriculture using intelligent sensor techniques: A review of the aeroponic system", ***Journal of Sensors,*** 2(3): 12-19 (2018).

101. Osama, R., El-Huda Ashraf, N., Yasser, A., AbdelFatah, S., El Masry, N., & AbdelRaouf, A., "Greenhouse Plant Growth Supervision with the LED Lights using Machine Learning", ***In Proceedings of the 2020 9th International Conference on Software and Information Engineering,*** Egypt, 169-173 (2020).

102. Hemming, S., de Zwart, F., Elings, A., Righini, I., & Petropoulou, A., "Remote control of greenhouse vegetable production with artificial intelligence Greenhouse climate, irrigation, and crop production", ***Sensors,*** 19(8): 18-70 (2019).

103. Internet: Sharma, V. "Unsupervised Learning an Angle for Unlabelled Data World," **https://www.datasciencecentral.com/profiles/blogs/unsupervised-learning -an-angle-for-unlabelled-data-world**/(2021).

APPENDIX A

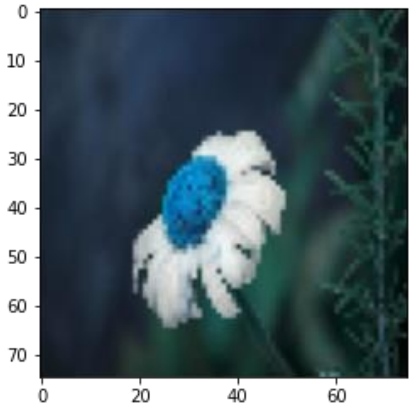
MONITORING SYSTEM CODE

<u>CODE STARTS HERE</u>

```
import numpy as np
import matplotlib.pyplot as plt
import os
import cv2
DATADIR = "C:\Daisy Flowers 2"
CLASS = ["Unwanted", "Wanted"]
#Define path of directory
for category in CLASS:
   path = os.path.join(DATADIR, category)
   for img in os.listdir(path):
      img_array = cv2.imread(os.path.join(path,img))
      plt.imshow(img_array)
      break
   break
```


```
IMG_SIZE = 75
new_array = :::cv2.resize(img_array, (IMG_SIZE, IMG_SIZE))
```

```
plt.imshow(new_array)
plt.show()
```




```
daisy_data = []
def create_daisy_data(,):
    for category in CLASS:
        path = os.path.join(DATADIR, category)
        class_num = CLASS.index(category)
        for img in os.listdir(path):
            try:
                img_array = cv2.imread(os.path.join(path,img))
                new_array = cv2.resize(img_array, (IMG_SIZE,, IMG_SIZE))
                daisy_data.append([new_array, class_num])
            except Exception as e:
                pass
create_daisy_data()
X = []
y=[]
for faetures, label in daisy_data:
    X.append(faetures)
    y.append(label)
X = np.array(X).reshape(-1, IMG_SIZE, IMG_SIZE, 3)
y = np.array(y)
import pickle
pickle_out = open("X.pickle","wb")
```

```
pickle.dump(X, pickle_out)
pickle_out.close()
pickle_out = open("y.pickle","wb")
pickle.dump(y, pickle_out)
pickle_out.close()
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Dropout, Activation, Flatten, Conv2D,
MaxPooling2D
import pickle
from tensorflow.keras.callbacks import TensorBoard
import time
NAME = "Daisy-Training-Model-Conv32/64".format(int(time.time()))
tensorboard = TensorBoard(log_dir="logs/{}".format(NAME))
X = pickle.load(open("X.pickle", "rb"))
y = pickle.load(open("y.pickle", "rb"))

X = X/255.0
model = Sequential()
model.add(Conv2D(32, (3,3), input_shape = X.shape[1:]))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size = (2,2)))
model = Sequential()
model.add(Conv2D(64, (3,3)))
model.add(Activation("relu"))
model.add(MaxPooling2D(pool_size = (2,2)))
model.add(Flatten())
model.add(Dense(64))
model.add(Activation("relu"))
model.add(Dense(1))
model.add(Activation("sigmoid"))
model.compile(loss="binary_crossentropy",
          optimizer="adam",
```

```
metrics=["accuracy"])
history = model.fit(X, y, batch_size=20, epochs=5, validation_split=0.2,
callbacks=[tensorboard])
import keras
import matplotlib.pyplot as plt
plt.plot(history.history['acc'])
plt.plot(history.history['test_acc'])
plt.title('model accuracy')
plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
plt.plot(history.history['loss'])
plt.plot(history.history['test_loss'])
plt.title('model loss')
plt.ylabel('loss')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

 # Image processing code

```
from keras.models import Sequential
from keras.layers import Conv2D, MaxPool2D, Flatten, Dense,
Dropout,BatchNormalization
import tensorflow as tf
from keras import optimizers
import datetime, os
import time
import numpy as np
from PIL import Image
import cv2
```

 # from tqdm import tqdm

```
import glob
```

```
import pickle
from image_splitter import image_split
#from os import system
# take new snapshot
#system("ls -la")
import os
command ="raspistill -vf -hf -o cam2.jpg"
os.system(command)
time.sleep(10)
path_of_image = "cam2.jpg"
image_split(path_of_image)
model = tf.keras.models.load_model('model')
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
processed_images = []
# The five steps (algorithm) for image processing
sub_images_number = 8
for i in range(sub_images_number):
    sub_image_path = f"sub_image_{i + 1}.jpg"
    sub_image = cv2.imread(sub_image_path)
    sub_image = cv2.resize(sub_image, (64, 64))
    sub_image = cv2.cvtColor(sub_image, cv2.COLOR_BGR2RGB)
    sub_image = sub_image.astype('float32')
    sub_image = cv2.normalize(sub_image, None, alpha=0, beta=1, norm_type=cv2.NORM_MINMAX, dtype=cv2.CV_32F)
    # add it to the array
    processed_images.append(sub_image)
    imx = np.array(processed_images)
    classes = model.predict_classes(imx)
    # Image processing output
    processed_images = []
    print(f"image {i}:", end=" ")
```

```
if classes == 1:
    print("WANTED")
else:
    print("UNWANTED")
    n=1
    for x in range(0, n):
        print("click space")
        img = cv2.imread(f"sub_image_{i+1}.jpg")
        cv2.imshow(f"sub_{i+1}", img)
        cv2.waitKey()
```

CODE ENDS HERE

APPENDIX B

CONTROL SYSTEM CODE

CODE STARTS HERE

```
#include <Servo.h>
#include <LiquidCrystal.h>
#include <Adafruit_Sensor.h>
#include <DHT.h>
#include <DHT_U.h>
#include <Wire.h>
#define REMOTEXY_MODE__SOFTSERIAL
#include <SoftwareSerial.h>
#include <RemoteXY.h>
// RemoteXY connection settings
#define REMOTEXY_SERIAL_RX 12
#define REMOTEXY_SERIAL_TX 13
#define REMOTEXY_SERIAL_SPEED 9600

int relay1 = 3 ;
int relay2 = 4 ;
int fan1 = 8 ;
int fan2 = 9 ;
int watering = 7 ;
Servo servo ;
int x,y;
int ref1;
int ms,fs;
bool openedWindow = false;
DHT dht1(2, DHT11);

int nemtoprak1=A1;
// RemoteXY configurate
#pragma pack(push, 1)
uint8_t RemoteXY_CONF[] =
  { 255,2,0,61,0,1,1,11,245,0,
  129,0,19,14,9,4,17,70,97,110,
  50,0,129,0,72,14,11,4,17,87,
  97,116,101,114,0,129,0,2,14,9,
  4,17,70,97,110,49,0,129,0,32,
  14,18,4,17,83,112,101,99,116,114,
  117,109,0,129,0,54,14,15,4,17,
```

```
  73,110,102,114,97,114,101,100,0,129,
  0,88,14,9,4,17,82,97,105,110,
  0,4,128,2,21,54,9,2,26,4,
  128,2,32,54,9,2,26,67,4,75,
  44,20,6,2,26,11,67,4,42,44,
  19,6,2,26,11,129,0,78,24,16,
  4,17,114,101,102,32,84,101,109,112,
  0,129,0,75,34,22,4,17,114,101,
  102,32,83,111,105,108,72,117,109,0,
  67,4,9,44,19,7,2,26,11,67,
  4,54,32,19,7,2,26,11,67,4,
  54,22,19,7,2,26,11,65,4,1,
  1,12,12,65,4,18,1,12,12,65,
  4,36,1,12,12,65,4,55,1,12,
  12,65,4,71,1,12,12,65,4,86,
  1,12,12,129,0,76,53,18,6,17,
  65,105,114,72,117,109,0,129,0,41,
  53,18,6,17,83,111,105,108,72,117,
  109,0,129,0,9,53,18,6,17,84,
  101,109,112,0 };

// this structure defines all the variables and events of your control interface
struct {

    // input variables
  int8_t slider_temp; // =0..100 slider position
  int8_t slider_soilhum; // =0..100 slider position

    // output variables
  uint8_t fan1_led_r; // =0..255 LED Red brightness
  uint8_t Relay1_led_r; // =0..255 LED Red brightness
  uint8_t fan2_led_r; // =0..255 LED Red brightness
  uint8_t Relay2_led_r; // =0..255 LED Red brightness
  uint8_t water_led_r; // =0..255 LED Red brightness
  uint8_t rain_led_r; // =0..255 LED Red brightness
  char Airhum_indicator[11];  // string UTF8 end zero
  char Soilhum_indicator[11];  // string UTF8 end zero
  char Temp_indicator[11];  // string UTF8 end zero
  char ref_temp[11];  // string UTF8 end zero
  char ref_hum[11];  // string UTF8 end zero

    // other variable
  uint8_t connect_flag;  // =1 if wire connected, else =0

} RemoteXY;
#pragma pack(pop)
void setup()
{
Serial.begin(9600);
```

```
dht1.begin();

pinMode(relay1, OUTPUT);
pinMode(relay2, OUTPUT);
pinMode(fan1,OUTPUT);
pinMode(fan2,OUTPUT);
pinMode(nemtoprak1,INPUT);
pinMode(watering,OUTPUT);
  RemoteXY_Init ();
  RemoteXY.slider_temp = 50;
  }
void loop() {
  delay(100);
/////////////////////////////////////
    float h1 = dht1.readHumidity();
    //float t1 = dht1.readTemperature();
    float t1 = dht1.readTemperature();

///////////////////////////////////////
  Serial.print(F("  TemP: "));
  Serial.print(t1);
  Serial.print(F(" AIR HUM "));
  Serial.print(h1);
  Serial.print(F(" SOIL HUM "));
  Serial.print(nemtoprak1);
  Serial.print(F(" slider "));
  Serial.print(ref1);
  Serial.print("\n");

  digitalWrite(relay1,LOW);
  RemoteXY.Relay1_led_r =255;
int nem1=analogRead(nemtoprak1);

    delay(1000);
     Serial.print(openedWindow);
if(t1>ms){
 digitalWrite(relay2,HIGH);
 RemoteXY.Relay2_led_r =0;
 if(openedWindow == false)
 {
  servo.detach();
  delay(200);
  servo.attach(5);
  servo.write(180);
  delay(200);
  servo.detach();
  openedWindow = true;
 }
  digitalWrite(fan1,HIGH);
```

```
  digitalWrite(fan2,HIGH);
  RemoteXY.fan1_led_r =255;
  RemoteXY.fan2_led_r =255;
}
else {
  if(openedWindow == true)
 {
  servo.attach(5);
    delay(200);
  servo.write(0);
    delay(200);

  servo.detach();
  openedWindow = false;
 }
 digitalWrite(relay2,LOW);
 RemoteXY.Relay2_led_r =255;
  digitalWrite(fan1,LOW);
  digitalWrite(fan2,LOW);
  RemoteXY.fan1_led_r =0;
  RemoteXY.fan2_led_r =0;

}
if(nemtoprak1>fs){
  digitalWrite (watering,LOW);
  RemoteXY.water_led_r =0;
}
else  {
  digitalWrite (watering,HIGH);
  RemoteXY.water_led_r =255;
}
  RemoteXY_Handler ();

RemoteXY.slider_soilhum = min(0,100);
dtostrf(t1,0,1, RemoteXY. Temp_indicator); //temp
dtostrf(h1,0,1, RemoteXY.Airhum_indicator);
dtostrf(nemtoprak1,0,1, RemoteXY.Soilhum_indicator);
 ms =1*0.5* RemoteXY.slider_temp;
//  ms =50;

  fs =1*0.5* RemoteXY.slider_soilhum;


dtostrf(ms,0,1, RemoteXY.ref_temp);
dtostrf(fs,0,1, RemoteXY.ref_hum);

}
```

CODE ENDS HERE

RESUME

Abdalrazg Fathi BENNASER was born in Derna and he graduated primary, elementary, and high school in this city, after that, he started the bachelor degree at Omar Almukhtar University, Department of Communication Engineering in 2008. Then in 2019, he started at Karabuk University, Electrical and Electronics Engineering to complete his M. Sc. education.