GREEN FIELDS CONNECT WEBSITE USING DEEP LEARNING

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

J.ESWAR SAI NITHESH	(21UECM0101)	**(VTU 20324)**
I.SAI VAMSI	(21UECS0234)	**(VTU 19738)**
S.TANUJ REDDY	(21UECS0584)	**(VTU 19798)**

Under the guidance of
Dr.P.Siva Prakash,M.E.,Ph.D.,
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

January, 2024

GREEN FIELDS CONNECT WEBSITE USING DEEP LEARNING

Minor project-1 report submitted
in partial fulfillment of the requirement for award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

By

J.ESWAR SAI NITHESH	(21UECM0101)	**(VTU 20324)**
I.SAI VAMSI	(21UECS0234)	**(VTU 19738)**
S.TANUJ REDDDY	(21UECS0584)	**(VTU 19798)**

Under the guidance of
Dr.P.Siva Prakash,M.E.,Ph.D.,
Associate Professor

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF COMPUTING

VEL TECH RANGARAJAN DR. SAGUNTHALA R&D INSTITUTE OF SCIENCE & TECHNOLOGY

(Deemed to be University Estd u/s 3 of UGC Act, 1956)
Accredited by NAAC with A++ Grade
CHENNAI 600 062, TAMILNADU, INDIA

January, 2024

CERTIFICATE

It is certified that the work contained in the project report titled "GREEN FIELDS CONNECT WEBSITE USING DEEP LEARNING" by J.ESWAR SAI NITHESH (21UECM0101), S.TANUJ REDDY (21UECS0584), I.SAIVAMSI (21UECS0234) has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

Signature of Supervisor
Dr.P.Siva prakash
Associate Professor
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr.Sagunthala R&D
Institute of Science & Technology
January, 2024

Signature of Head of the Department
Dr.M.S. Murali Dhar
Associate Professor & HOD
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
January,2024

Signature of the Dean
Dr. V. Srinivasa Rao
Professor & Dean
Computer Science & Engineering
School of Computing
Vel Tech Rangarajan Dr. Sagunthala R&D
Institute of Science & Technology
January, 2024

DECLARATION

We declare that this written submission represents our ideas in our own words and where others' ideas or words have been included, we have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in our submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

J.ESWAR SAI NITHESH

Date: / /

S.TANUJ REDDY

Date: / /

I.SAI VAMSI

Date: / /

APPROVAL SHEET

This project report entitled ”GREEN FIELDS CONNECT WEBSITE USING DEEP LEARNING” by J.ESWAR SAI NITHESH (21UECM0101), S.TANUJ REDDY (21UECS0584), I.SAIVAMSI (21UECS0234) is approved for the degree of B.Tech in Computer Science & Engineering.

Examiners **Supervisor**

Dr.P.Siva Prakash, M.E.,Ph.D.,

Date: / /

Place:

ACKNOWLEDGEMENT

We express our deepest gratitude to our respected **Founder Chancellor and President Col. Prof. Dr. R. RANGARAJAN B.E. (EEE), B.E. (MECH), M.S (AUTO),D.Sc., Foundress President Dr. R. SAGUNTHALA RANGARAJAN M.B.B.S.** Chairperson Managing Trustee and Vice President.

We are very much grateful to our beloved **Vice Chancellor Prof. S. SALIVAHANAN,** for providing us with an environment to complete our project successfully.

We record indebtedness to our **Professor & Dean, Department of Computer Science & Engineering, School of Computing, Dr. V. SRINIVASA RAO, M.Tech., Ph.D.,** for immense care and encouragement towards us throughout the course of this project.

We are thankful to our **Head, Department of Computer Science & Engineering, Dr.M.S. MURALI DHAR, M.E., Ph.D.,** for providing immense support in all our endeavors.

We also take this opportunity to express a deep sense of gratitude to our **Internal Supervisor Dr.P.SIVA PRAKASH, M.E.,Ph.D.,** for his cordial support, valuable information and guidance, he helped us in completing this project through various stages.

A special thanks to our **Project Coordinators Mr. V. ASHOK KUMAR, M.Tech., Ms. C. SHYAMALA KUMARI, M.E., Mr. SHARAD SHANDHI RAVI, M.Tech.,** for their valuable guidance and support throughout the course of the project.

We thank our department faculty, supporting staff and friends for their help and guidance to complete this project.

J.ESWAR SAI NITHESH (21UECM0101)
S.TANUJ REDDY (21UECS0584)
I.SAIVAMSI (21UECS0234)

ABSTRACT

The "Green Fields Connect Website Using Deep Learning" represents a comprehensive solution for modern agriculture by integrating Convolutional Neural Networks (CNNs) into a user-friendly web application. This innovative project leverages deep learning techniques to predict suitable crops based on environmental and soil conditions, providing farmers with valuable insights for optimized cultivation practices.The project initiates with the exploration and preprocessing of an agricultural dataset obtained from an official crop database. Data cleaning procedures handle duplicates and missing values, while Exploratory Data Analysis (EDA) offers insights into the distribution and relationships among critical agricultural parameters such as nitrogen, phosphorus, potassium, temperature, humidity, pH, and rainfall.Unlike traditional machine learning models, this project employs a Convolutional Neural Network (CNN) implemented using the Keras library. The CNN architecture allows the model to learn hierarchical representations from input images, in this case, the agricultural parameters. The model is trained to recognize patterns that indicate optimal crop recommendations based on the input conditions.

Keywords:

Convolutional Neural Network, Data analysis, Web Application ,Machine Learning, Predictive Modeling,

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

SNO	ABBREVIATION	ACRONYM
1	API	Application Programing Interface
2	CNN	Convolutional Neural Networks
3	CSS	Cascading Style Sheet
4	EDA	Exploratory Data Analysis
5	HTML	Hyper Text Markup Language
6	ROI	Return On Investment
7	UI	User Interface

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

1.1 Introduction

The ”Green Fields Connect Website Using Deep Learning” is an innovative and intelligent application designed to revolutionize modern agriculture by harnessing the power of Convolutional Neural Networks (CNNs) for crop prediction. Agriculture, a critical sector for global sustenance, faces challenges in optimizing crop selection based on diverse environmental and soil conditions. This project addresses these challenges by combining data science, machine learning, and web development to provide farmers with accurate and tailored recommendations for crop cultivation.Traditional crop recommendation systems often rely on conventional machine learning models. However, this project adopts a novel approach by implementing a CNN, a specialized neural network architecture renowned for its prowess in image recognition tasks. By treating agricultural parameters as an image, the CNN can capture intricate patterns and relationships among the variables, ultimately enhancing the accuracy of crop predictions.

1.2 Aim of the project

The primary aim of the ”Green Fields Connect Website Using Deep Learning ” is to revolutionize and modernize agricultural practices by integrating advanced machine learning, specifically Convolutional Neural Networks (CNNs), into a user-friendly web application. The project seeks to address the challenges faced by farmers in selecting the most suitable crops based on diverse environmental and soil conditions, ultimately enhancing the efficiency and sustainability of crop cultivation.

1.3 Project Domain

The ”Green Fields Connect Website Using Deep Learning” operates within the domain of Precision Agriculture and Crop Management. Precision Agriculture, also known as precision farming, involves the use of advanced technologies to optimize crop yields, reduce waste, and enhance overall farm efficiency. In this context, the project focuses on leveraging data science and machine learning techniques to provide farmers with precise recommendations for crop cultivation.

Key Components of the Project Domain:
Data-Driven Decision Making Harnessing data to make informed decisions in agriculture. Utilizing machine learning models to analyze and interpret complex datasets related to soil and environmental conditions.

1. Crop Prediction Models:
Implementing advanced machine learning models, such as Convolutional Neural Networks (CNNs), for accurate and nuanced crop predictions.Enhancing the prediction capabilities by treating agricultural parameters as image-like data, allowing for the extraction of intricate patterns. Environmental and Soil Factors: Considering key environmental factors, including temperature, humidity, and rainfall.

2.Web-Based User Interface:
Developing a user-friendly web application for seamless interaction with the recommendation system. Allowing farmers to input specific agricultural parameters through an intuitive interface. Sustainable Agriculture Practices: Contributing to sustainable farming practices by optimizing crop selection based on environmental conditions. Reducing resource waste and promoting efficient land use through tailored crop recommendations.

3.Precision Agriculture Technologies:
Aligning with the broader field of precision agriculture, which integrates technologies such as sensors, GPS guidance, and automated machinery. Positioning the project as a component of the evolving landscape of precision agriculture technologies.

4.Optimized Crop Cultivation:
Empowering farmers to make data-driven decisions leading to optimized crop cultivation. Enhancing overall farm productivity and profitability through targeted and efficient farming practices.

1.4 Scope of the Project

The "Green Fields Connect Website Using Deep Learning" envisions a comprehensive scope within the domain of Precision Agriculture, aiming to provide farmers with advanced tools for optimized crop selection. The system analyzes crucial environmental parameters such as temperature, humidity, and rainfall to recommend crops suited to specific climatic conditions. Incorporating soil composition factors including nitrogen, phosphorus, potassium, and pH to tailor recommendations based on soil health and nutrient levels. Integration of Convolutional Neural Networks Leveraging CNN's to process agricultural parameters as image-like data. Developing an intuitive web interface that enables farmers to input agricultural parameters seamlessly, making the system accessible to users with varying technical backgrounds. Dynamic and Real-time Predictions: Providing real-time predictions for crop recommendations. Rigorously training the machine learning model with a diverse dataset and evaluating its performance to ensure reliable and accurate crop predictions. Presenting the recommendations in a user-friendly format, allowing farmers to easily interpret and implement the suggested crop cultivation strategies. Adaptability to Diverse Agricultural Practices: Designing the system to accommodate a variety of crops and agricultural practices, ensuring its applicability across different regions.

Chapter 2

LITERATURE REVIEW

[1] Dr.V. Geetha, et al., (2023) proposed a paper in which they have used a Machine Learning algorithm named as Random Forest algorithm for analyzing the growth of the crop in relation to the different climatic conditions like dried period, increasing in temperatures and biophysical changes. Here, for the correct prediction of crop, crop growth dataset has been collected from different sources. This dataset has been used for both training and testing purposes and Random Forest algorithm is applied for crop prediction. It is observed that the Random Forest algorithm gives more accurate results as compared to other algorithms.

[2] Zeel Doshi, et al., (2022) proposed a paper in which they come up with an intelligent system called AgroConsultant. This system aims to support the Indian farmer in making a concise decision about the growth of a crop on the basis of sowing season, geographical location of the farmer, characteristics of the soil and environmental factors like temperature and rainfall. They used two subsystems in their proposed intelligent system. The first system concerns the recommendation of crops to the farmer and the second system is related to the prediction of the rainfall for a particular region which is fed to the first system for crop sustainability prediction.

[3] A.Priyadharshini, et al.,(2022) proposed a paper named in which they have proposed a system which will assist the farmer to choose the right crop by providing the intuition which cannot be tracked. It decreases the chances of crop failure and increases productivity. It helps the farmer by preventing them from suffering losses. For efficient yield forecasting prediction real time monthly weather data is taken. Profit analysis of various crops was done based on the previous year's data. Here, a sequential model is implemented with 3 input layers and 15 output layers. The proposed system is implemented by using Linear Regression algorithm and Neural Network by using Pandas, NumPy, TensorFlow, Keras and Scikitlearn libraries, tools, and Python as the programming language.

[4] P.Vaishnavi, et al., (2018) proposed a paper in which the proper prediction of crops is informed to the agriculturists based on real time by considering the various parameters like production and season. Different Data Mining techniques and Big Data techniques are followed by the system for precise prediction of crops. Then personalized and relevant recommendations are given by the system to the farmers which results in yielding good volume production.

[5] Dr.Y.Jeevan Nagendra Kumar, et al., (2016) implemented a Supervised Machine Learning approach for Crop Yield Prediction in the Agriculture Sector. In this paper, authors take the past data and use it to predict the future yield. Mainly crop yield depends on weather and pesticides. The dataset used from Kaggle repository which consists of parameters like rainfall, perception, temperature, and production. The data set has 3101 instances that have taken from the past historic data. Only two-third of the data is used for training. Random Forest algorithm used which has 3 factors like n tree, m try, Node size. Accuracy of the Random Forest algorithm is compared with other algorithms. It will provide more accuracy. Here, the Random Forest algorithm provides the best crop yield as output. Information provided is accurate so results about crop yield are more accurate and farmers have more profit. The proposed system covers maximum types of crops. This system will predict accurate results of different crops so it will be helpful for Indian farmers.

[6] Ms Kavita, et al., (2021) implemented a system in which they have predicted the crop yield for India by using the data from 1950 to 2018. For the prediction five crops were used which are Rice, Wheat, Jowar, Bajra, Tobacco, and Maize. The data consists of 745 instances in which 70 system delivers the farmer with a detailed set of recommendations to optimize their selection of crops based on various factors like location, farm size, temperature, rainfall, and various crop dataset.

[7] A. L. Ismail et al., (2018) created a framework to predict preparedness of a country to face the climate change using machine learning approach. The study is done for South East Asia. Steps for calculating the predictive index are data acquisition, data training, data testing, index predicting, index validation and index visualisation. The study is a precautionary measure to alert the regions and verify its vulnerable index using deep learning.

[8] Zhen Nan Liu, et al., (2018), In this paper, authors have compared different machine learning algorithms for calculating, Standardised Precipitation Index (SPI) and SPEI. After data collection, Extreme learning methods, Online sequential extreme learning machine, Selfadaptive evolutionary extreme learning machine. Authors claimed that all three algorithms can be applied successfully on drought forecasting. However, OS-ELM and SADE- ELM performs better than ELM.

[9] Chaoyun Zhang, et al., (2018) presented an ample survey of the crossovers between the two areas. A brief study of applications of networking using deep learning techniques is done. We then discuss several techniques and platforms that facilitate the efficient deployment of deep learning onto mobile systems. Authors focus on how can deep learning can be useful for mobile and wireless networking. This is a suvey paper surfacing the issues and challenges in deep learning in wireless and mobile networks.

[10] K.G. Liakos et al., (2018) presented a comprehensive review of research dedicated to machine learning applications in agriculture domain. Various parameters on which work was analysed were: crop management, livestock management, water management and soil management. ML models have applied for crop yield prediction and disease detection. ML based detection can be extracted without the need of fusion of data from other resources. Author claims that farm management systems are evlvong into real artificial intelligent systems, with the ultimate scope of production improvement. Author motivates to use ML for the benefit of agriculture as it is the basic need amongst all other needs for survival.

Chapter 3

PROJECT DESCRIPTION

3.1 Existing System

The existing agricultural system in many regions relies heavily on traditional and manual practices for crop selection. Farmers often make decisions based on their experience, local knowledge, and sometimes limited advice from agricultural experts. The traditional approach may involve trial and error, leading to suboptimal crop choices and reduced overall productivity. Additionally, external factors such as changing climate conditions, varying soil types, and unpredictable weather patterns make it challenging for farmers to consistently make informed decisions about the crops they cultivate.

In some cases, farmers may seek guidance from agricultural extension services, which provide recommendations based on general guidelines and historical data. However, these recommendations may not always consider the specific and dynamic conditions of individual farms, leading to less accurate and personalized suggestions.

3.2 Proposed System

The proposed system introduces an innovative approach to crop recommendation, leveraging advanced technologies such as data mining, Deep learning, and precision agriculture to address the limitations of traditional farming practices. The primary goal is to provide farmers with accurate and personalized recommendations for crop selection based on site-specific parameters, ultimately enhancing agricultural productivity.

The system incorporates a comprehensive ensemble model that utilizes machine learning algorithms, including Random Tree, CHAID, K-Nearest Neighbor, and Naive Bayes. By employing a majority voting technique, the system ensures robust and reliable crop recommendations, taking into account a wide range of factors such as soil characteristics, soil types, and historical crop yield data.

3.3 Feasibility Study

The feasibility study for the proposed crop recommendation system involves a comprehensive assessment of various aspects to determine the practicality and viability of implementing the system. The study encompasses several key dimensions:

3.3.1 Economic Feasibility

Cost-Benefit Analysis:
Conducting a detailed analysis of the costs associated with system development, implementation, and maintenance against the anticipated benefits, including increased crop yield and financial returns for farmers.

Return on Investment (ROI):
Calculating the projected ROI to determine the economic viability of the proposed system.

3.3.2 Technical Feasibility

System Architecture:
Assessing the technical requirements and architecture needed for the implementation of the ensemble model, precision agriculture techniques, and data mining algorithms.

Data Availability:
Evaluating the availability and accessibility of relevant data, including historical crop yield data, soil characteristics, and real-time environmental variables.

3.3.3 Social Feasibility

User Acceptance:
Gauging the willingness and acceptance of farmers to adopt and use the proposed system, considering factors such as user-friendliness and accessibility.

Training Requirements:
Identifying the training needs for farmers and stakeholders to effectively use the system.

3.4 System Specification

Use a certain project in an efficient way, we need hardware components and software components to be present in a computer. These components are been used as a guideline for a project. While the Increase Of higher processing power and new versions of software it increases time management. By this, we can conclude that it plays a bigger role in computer systems

3.4.1 Hardware Specification

• Processor : I3/I5 Intel Processor

• RAM : 8 GB

3.4.2 Software Specification

• PYCHARM

• PYTHON

• PRE-TRAINED MODELS

• DEEP LEARNING MODELS

• WEBCAM

3.4.3 Standards and Policies

Data Privacy and Security:
All user data, including personal information and farm-related data, must be treated with the utmost confidentiality. Implement encryption mechanisms for data transmission and storage. Regularly audit and assess security protocols to identify and address potential vulnerabilities.

Regulatory Compliance:
Ensure compliance with regional and international data protection regulations. Adhere to agricultural standards and policies set by relevant government bodies.

Ethical Use of Data:
Collect data transparently, informing users about the purpose and use of their information. Obtain explicit consent from users for data collection and utilization. Avoid any discriminatory practices in crop recommendations.

Algorithmic Transparency:
Provide clear explanations of the algorithms and models used in making crop recommendations. Disclose the sources of data and the criteria influencing the recommendations.

Accessibility and Inclusivity:
Ensure that the crop recommendation system is accessible to farmers with diverse needs and abilities. Provide support for multiple languages and dialects.

Accuracy and Reliability:
Regularly update and retrain machine learning models to improve accuracy. Implement mechanisms for feedback from users to enhance the system's reliability.

User Education and Awareness:
Educate users about the capabilities and limitations of the system. Provide resources to help users interpret recommendations and make informed decisions.

Interoperability:
Strive for compatibility with existing agricultural systems and databases. Facilitate seamless data exchange with other platforms and technologies.

Community Engagement:
Foster engagement with local farming communities to understand their needs and challenges. Seek feedback from users to continually improve the system.

Environmental Impact:
Consider the environmental impact of the technology, promoting sustainable and eco-friendly practices. Encourage farmers to adopt practices that contribute to environmental conservation.

Chapter 4

METHODOLOGY

4.1 General Architecture

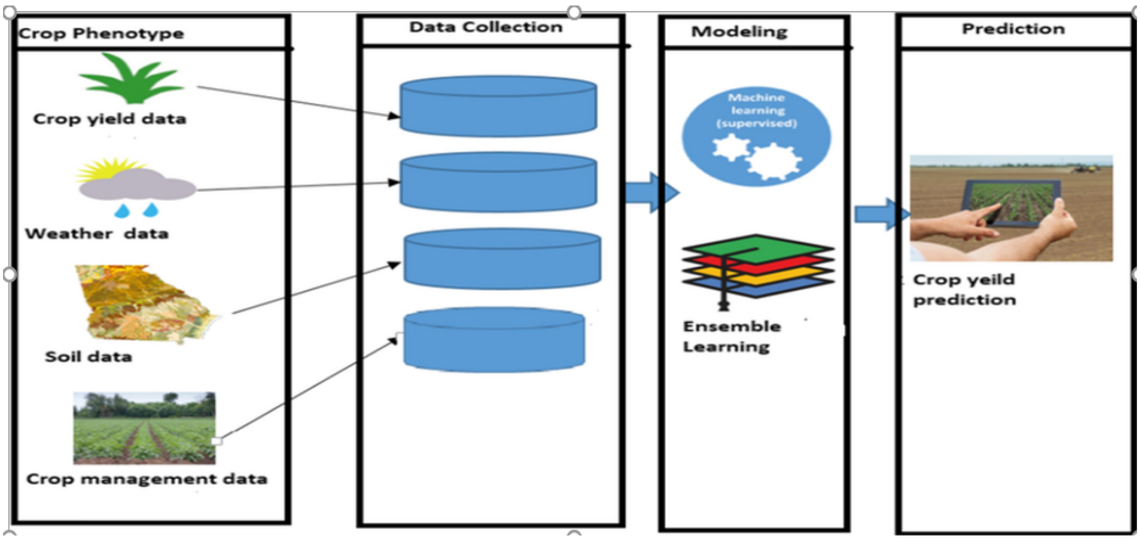

Figure 4.1: **Architecture**

Figure 4.1 explains about the crop suggesting website's architecture typically involves a user interface for input, a backend for data processing, and integration with external databases or APIs. Users input parameters like location, soil type, and climate, which are processed by the backend algorithms. The system then recommends suitable crops based on the analyzed data. The architecture should prioritize scalability, real-time data updates, and user-friendly interfaces for a seamless experience. Security measures are essential to protect sensitive user information and maintain the integrity of the system. Regular updates and collaboration with agricultural experts ensure accurate and relevant crop recommendations.The client side, accessible through a web browser, provides a user interface for farmers to input relevant information like soil type, climate, and preferences. On the server side, a database stores agricultural data, while algorithms analyze user inputs to recommend suitable crops. This architecture ensures efficient data processing and personalized recommendations for optimal crop selection based on various factors.

4.2 Design Phase

4.2.1 Data Flow Diagram

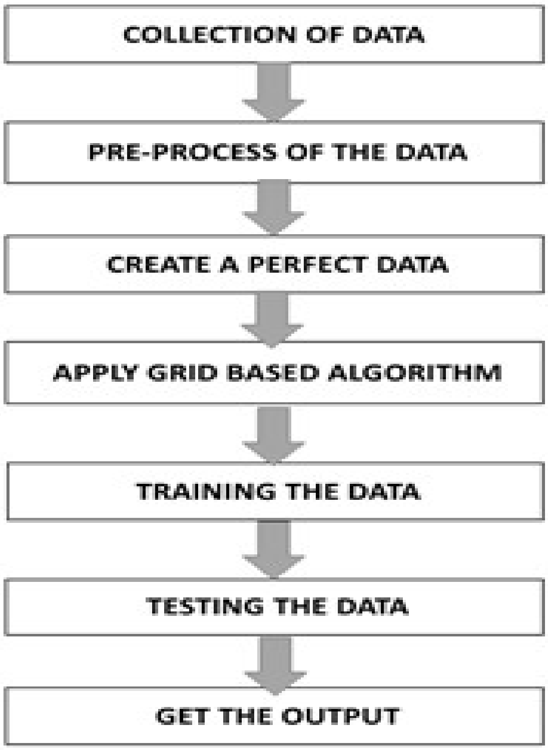

Figure 4.2: **Data Flow Diagram**

Figure 4.2 explains about the Green Fields Connect website employs a Data Flow Diagram (DFD) to illustrate the flow of information. Users interact with the system, initiating processes such as browsing or uploading images. Uploaded images undergo data preprocessing, ensuring compatibility with deep learning models. This step may involve resizing or normalization to optimize data for analysis.The central process involves deep learning inference, where sophisticated models analyze the preprocessed data. For instance, if the website incorporates deep learning for image recognition, this stage entails the model identifying patterns, objects, or features within the images. The results are then communicated back to users through the website interface. Throughout this data flow, the Green Fields Connect Server serves as the nexus, coordinating the interactions and facilitating the seamless movement of information. External entities, such as users, remain engaged with the website, while the underlying processes leverage deep learning to enhance functionalities. In summary, the DFD encapsulates the dynamic flow of data, depicting how user inputs traverse through preprocessing and deep learning inference.

4.2.2 Use Case Diagram

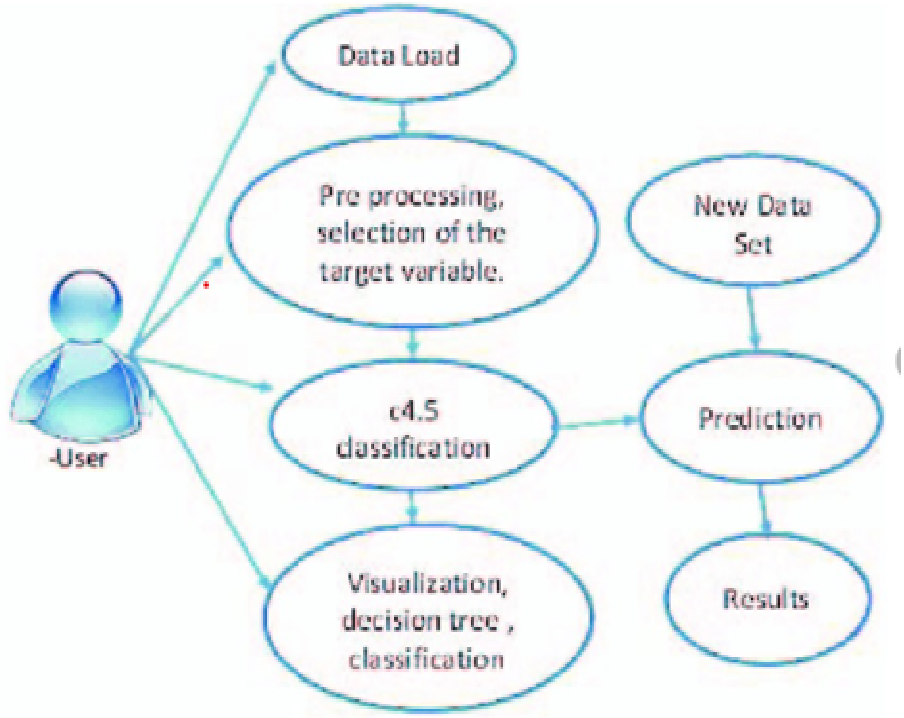

Figure 4.3: **Use Case Diagram**

Figure 4.3 explains about the use case diagram for the Green Fields Connect website with deep learning integration outlines key interactions between users and the system. The primary actors include regular users and administrators, each with distinct roles. Users can engage in use cases such as browsing content, uploading images for analysis through deep learning, and searching for specific content. The "Browse Content" use case allows users to explore the website's offerings, while "Upload Image" enables them to submit images for deep learning processing. The "Search for Content" use case empowers users to initiate searches, leveraging the capabilities of deep learning for more refined results. Administrators, as privileged users, may have additional use cases like content management or user account administration. This use case diagram offers a high-level view of how various actors interact with the Green Fields Connect website, emphasizing the integration of deep learning for enhanced user experiences and functionalities.

4.2.3 Class Diagram

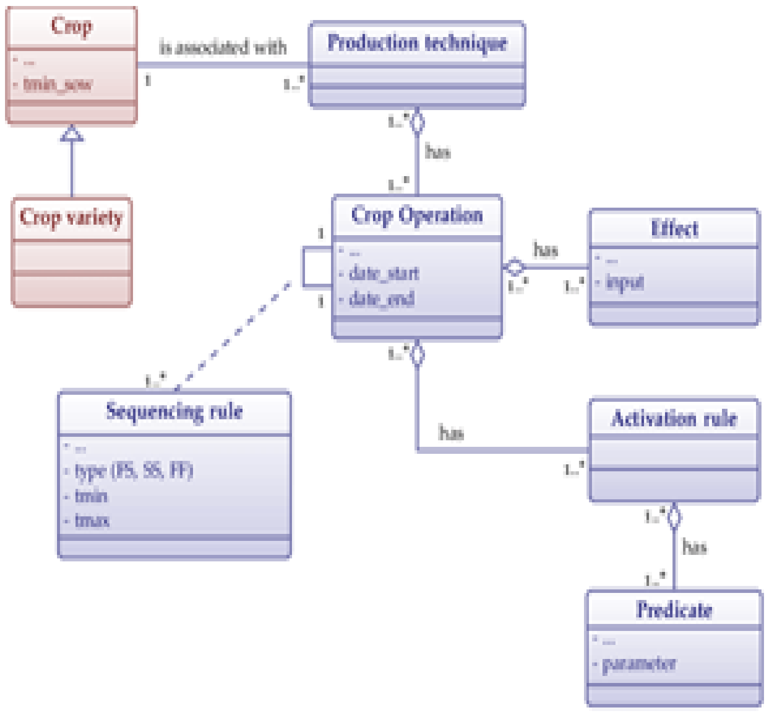

Figure 4.4: **Class Diagram**

Figure 4.4 explains about the class diagram for the Green Fields Connect website, incorporating deep learning, provides a visual representation of the static structure of the system. Key classes include ”User” and ”Admin,” each encapsulating attributes such as username, email, and preferences. Both classes may inherit from a shared class representing basic user functionalities, promoting code reuse.The ”Image” class signifies uploaded images with attributes like file name, date, and tags, forming associations with the ”User” class to establish the link between users and their uploaded content. Additionally, the ”Content” class represents various media types available on the website, serving as a parent class for images and potentially other content types. To depict the integration of deep learning, the diagram introduces the ”DeepLearningProcessor” class. This class encapsulates the functionality of processing images using deep learning algorithms, featuring methods like processImage(image). The associations between users, administrators, and content, along with the inclusion of the deep learning component, collectively present a comprehensive snapshot of the Green Fields Connect website’s underlying class structure.

4.2.4 Sequence Diagram

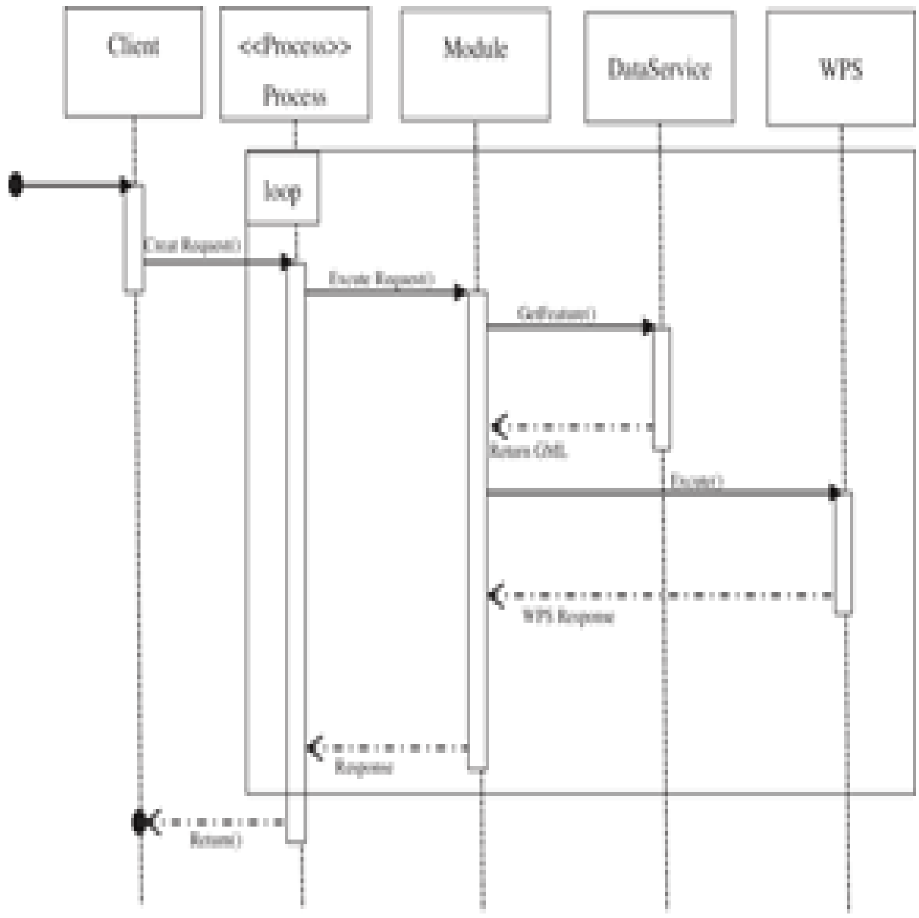

Figure 4.5: **Sequence Diagram**

Figure 4.5 explains about the sequence diagram for the Green Fields Connect website, integrating deep learning capabilities, illustrates the chronological flow of interactions among key components. The sequence initiates with the user uploading an image, triggering a message to the server indicating the image upload request. Subsequently, the server forwards the image to the DeepLearning Processor, symbolizing the commencement of deep learning analysis.Within the deep learning process, various messages may delineate stages such as preprocessing, feature extraction, and classification. Following this analysis, the server sends a response to the user, conveying the outcomes of the deep learning processing. This response may include notifications, updates, or the presentation of processed information on the user interface.The sequence diagram captures the dynamic nature of user-server interactions, emphasizing the seamless integration of deep learning into the website's functionality. The user, having received the server's response.

4.2.5 Collaboration diagram

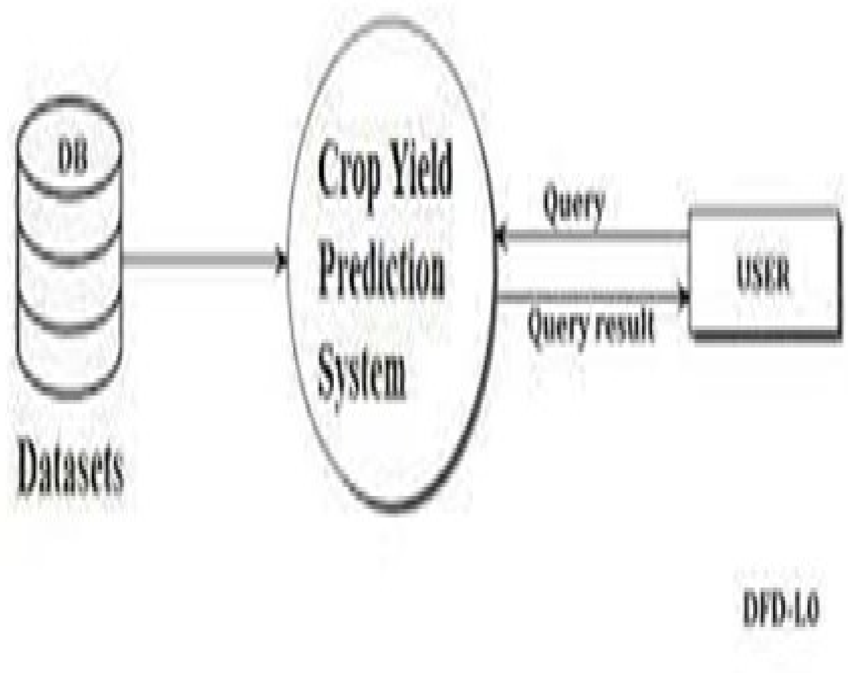

Figure 4.6: **Collaboration Diagram**

Figure 4.6 explains about the collaboration diagram for the Green Fields Connect website with deep learning integration, key components interact to demonstrate the flow of communication and relationships within the system. Objects include the ”User,” engaging with the website, the ”Server,” hosting the platform, and the ”DeepLearning Processor,” responsible for image analysis using deep learning algorithms.Communication lines, often represented by arrows, illustrate the flow of messages and interactions between these components. For instance, when a user initiates an image upload, a communication link conveys this request to the server, which subsequently communicates with the DeepLearning Processor for image processing. The collaboration diagram captures the dynamic relationships among these components, emphasizing the seamless communication essential for deep learning integration on the Green Fields Connect website.

4.2.6 Activity Diagram

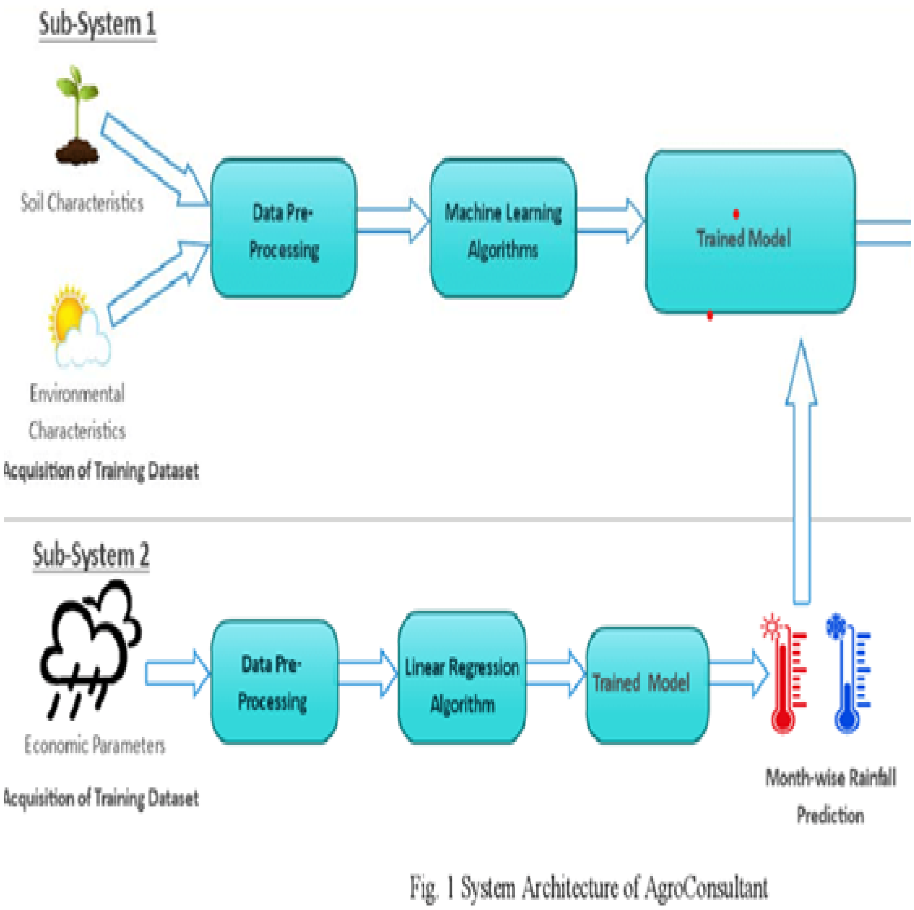

Figure 4.7: **Activitty Diagram**

4.3 Algorithm & Pseudo Code

4.3.1 Algorithm

The crop recommendation system employs a combination of machine learning algorithms, including Convolutional Neural Network (CNN) and Random Forest, to predict the optimal crop based on input parameters such as soil composition, weather conditions, and other relevant factors.

Convolutional Neural Network (CNN):

CNN is a deep learning algorithm primarily used for image classification but adapted in this context for its ability to extract hierarchical features from input data. In the system, CNN is employed to analyze and learn patterns from the environmental parameters provided as input. It consists of layers such as convolutional layers, pooling layers, and dense layers. Convolutional layers extract features.

Random Forest:

Random Forest is an ensemble learning method that operates by constructing multiple decision trees during training and outputs the mode of the classes as the prediction. In the crop recommendation system, Random Forest is utilized to analyze the input parameters and make predictions based on the collective decision-making of multiple trees. It is known for its versatility, efficiency, and resistance to overfitting.

Ensemble Model with Majority Voting:

The system implements an ensemble model that combines the predictions from multiple learners, including Random Forest, CHAID, K-Nearest Neighbor (KNN), and Naive Bayes. Majority voting is employed to determine the final recommendation. Each learner contributes its prediction, and the crop with the most votes is selected as the recommended crop. This approach leverages the strengths of different algorithms, potentially improving accuracy and robustness.

Data Preprocessing:

Before inputting data into the algorithms, a data preprocessing step is implemented. This involves handling missing values, removing duplicates, scaling features using StandardScaler, and encoding categorical labels using LabelEncoder. The preprocessing ensures that the input data is in a suitable format for the machine learning models.

Prediction:

After training, the models are used to make predictions on new input data. For the web-based application, when a user inputs soil and climate parameters, the models process the data and predict the most suitable crop for cultivation.

4.3.2 Pseudo Code

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Load dataset
dataset = pd.read_csv('crop_dataset.csv')

# Data preprocessing
# Handle missing values, remove duplicates
dataset = dataset.drop_duplicates().dropna()

# Define features (X) and target variable (y)
X = dataset.iloc[:, :-1].values
y = dataset.iloc[:, -1].values

# Encode categorical labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(y)

# Split dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Train Random Forest model
random_forest = RandomForestClassifier(n_estimators=100, random_state=42)
random_forest.fit(X_train, y_train)

# Evaluate model on the test set
accuracy = random_forest.score(X_test, y_test)
print(f"Accuracy on test set: {accuracy}")

# User input for prediction
new_input = np.array([24, 59, 18, 23.70, 54.21, 6.64, 124.77])
new_input_scaled = scaler.transform(new_input.reshape(1, -1))

# Make prediction
prediction = random_forest.predict(new_input_scaled)
predicted_crop = label_encoder.inverse_transform(prediction)

print(f"Predicted crop: {predicted_crop[0]}")
```

4.4 Module Description

4.4.1 Convolutional Neural Network (CNN)

CNN is a deep learning algorithm primarily used for image classification but adapted in this context for its ability to extract hierarchical features from input data. In the system, CNN is employed to analyze and learn patterns from the environmental parameters provided as input. It consists of layers such as convolutional layers, pooling layers, and dense layers. Convolutional layers extract features, pooling layers reduce dimensionality, and dense layers make predictions.. The model is trained to recognize patterns that indicate optimal crop recommendations based on the input conditions.The ”Crop Recommendation System” represents a comprehensive solution for modern agriculture by integrating Convolutional Neural Networks (CNNs) into a user-friendly web application. This innovative project leverages machine learning techniques to predict suitable crops based on environmental and soil conditions, providing farmers with valuable insights for optimized cultivation practices.Pooling layers then downsample the output of the convolution layers to reduce the dimensionality of the data and make it more computationally efficient.Using multiple convolution layers in a CNN allows the network to learn increasingly complex features from the input image or video. The first convolution layers learn simple features, such as edges and corners. The deeper convolution layers learn more complex features, such as shapes and objects.

4.4.2 Testing

Deploy the application in a real-world setting and collect feedback from actual users. Monitor the application’s performance and address any issues that arise.Implement a feedback loop for continuous improvement. Collect feedback from users and stakeholders to identify areas for enhancement. Consider incorporating additional features or expanding the dataset to improve the model’s recommendations. Precision agriculture techniques play a pivotal role in the proposed system, allowing for the collection and analysis of real-time data from the field. This includes factors like temperature, rainfall, humidity, and other environmental variables, contributing to more accurate predictions and personalized recommendations.Using multiple convolution layers in a CNN allows the network to learn increasingly complex features from the input image or video. The first convolution layers learn simple features, such as edges and corners. The deeper convolution layers learn more complex features, such as shapes and objects.

4.4.3 Collecting and Processing Data

Gather a comprehensive dataset containing information on various crops and environmental factors.Ensure the dataset is diverse, representative, and includes relevant features. Handle missing values, duplicate entries, and outliers. Encode categorical variables (e.g., crop labels) using Label Encoding. Split the dataset into features (X) and target variable (y). Visualize data distribution and relationships using plots and graphs. Understand correlations between different environmental factors. Deploy the application in a real-world setting and collect feedback from actual users. Monitor the application's performance and address any issues that arise.Implement a feedback loop for continuous improvement. Collect feedback from users and stakeholders to identify areas for enhancement. Consider incorporating additional features or expanding the dataset to improve the model's recommendations. Precision agriculture techniques play a pivotal role in the proposed system, allowing for the collection and analysis of real-time data from the field. This includes factors like temperature, rainfall, humidity, and other environmental variables, contributing to more accurate predictions and personalized recommendations'

4.5 Steps to execute/run/implement the project

4.5.1 Problem Understanding and Requirement Analysis

Clearly define the problem: Helping farmers in crop selection based on environmental factors. Understand user requirements: Input features (nitrogen, phosphorus, etc.), desired output (recommended crop).

4.5.2 Data collection

Gather a comprehensive dataset containing information on various crops and environmental factors. Ensure the dataset is diverse, representative, and includes relevant features.

4.5.3 Data Pre processing

Handle missing values, duplicate entries, and outliers. Encode categorical variables (e.g., crop labels) using Label Encoding. Split the dataset into features (X) and target variable (y).

4.5.4 Exploratory Data Analysis (EDA)

Visualize data distribution and relationships using plots and graphs. Understand correlations between different environmental factors. Identify patterns that may influence crop recommendations.

4.5.5 Model Selection

Choose appropriate machine learning models. In your case, you've used a Random Forest Classifier. Consider neural network models if applicable(e.g,CNN)

4.5.6 Flask Web Application

Develop a Flask web application (app.py) for user interaction. Load the trained model, scaler, and label encoder in the Flask application. Create HTML templates (index.html) for user input and result display.

4.5.7 User Interface (UI) Design

Design a user-friendly interface for users to input environmental factors. Implement a "Predict" button to trigger the recommendation process.

4.5.8 Testing

Test the web application thoroughly, including input validation and error handling. Verify that the model provides accurate crop recommendations.

4.5.9 Documentation

Document the entire project, including dataset details, model architecture, and web application structure. Provide clear instructions on how to reproduce and execute the project.

4.5.10 Future Improvements

Identify potential enhancements, such as incorporating more features or expanding the dataset. Consider continuous model training to adapt to changing agricultural conditions.

4.5.11 Reporting

Create a comprehensive report summarizing the project, methodologies, results, and potential future work. Following these steps should help you successfully execute the crop recommendation system project.

Chapter 5

IMPLEMENTATION AND TESTING

5.1 Input and Output

5.1.1 Input Design

Nitrogen	Phosphoru	Potassium	Temperatu	Humidity	pH	Rainfall	Label
61	45	32	25.77469	45.03411	6.820868	68.53067	mango
72	76	31	33.53516	66.026	9.652323	4.687727	mango
67	92	40	21.71462	38.09572	3.572579	43.17333	mango
110	76	18	25.0668	83.64738	4.144573	14.21189	mango
117	56	54	35.31844	50.26296	5.163631	16.04196	mango
13	15	19	29.97291	20.41041	8.561546	44.08111	mango
98	26	23	35.42742	28.18574	3.798388	63.75351	mango
9	45	35	29.53193	60.32627	7.941134	62.25071	mango
96	63	40	24.35987	83.36566	6.275272	9.31173	mango
52	38	49	17.52171	52.3418	8.337549	109.6717	mango

Figure 5.1: **Input Design**

5.1.2 Output Design

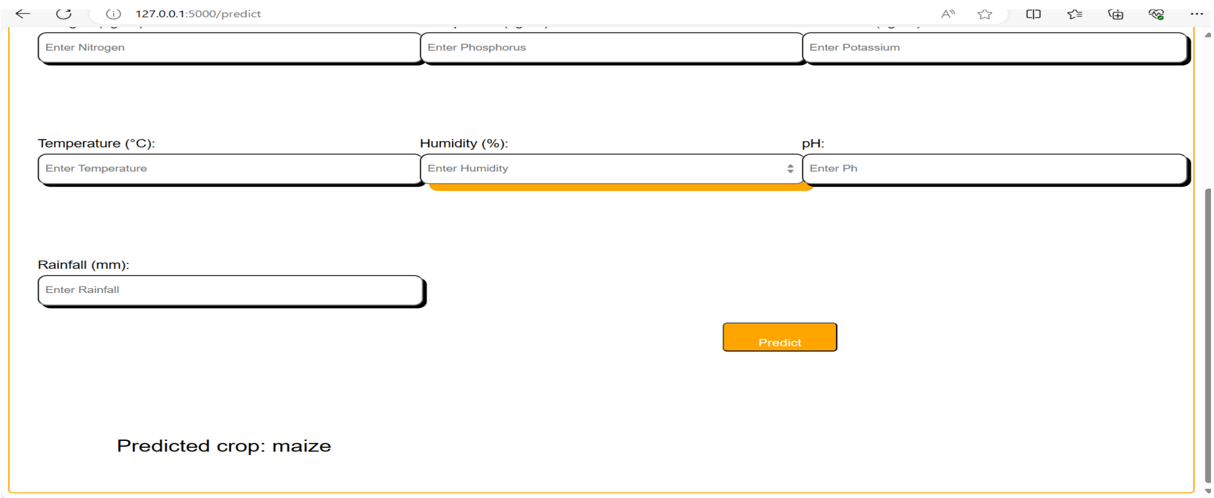

Figure 5.2: **Output Design**

5.2 Testing

5.3 Types of Testing

5.3.1 Unit testing

Input

```
model_accuracy = pd.DataFrame(columns=['Model','Accuracy'])
models = {
        "KNN" : KNeighborsClassifier(),
        "DT" : DecisionTreeClassifier(),
        'RFC' : RandomForestClassifier(),
        'GBC' : GradientBoostingClassifier(),
        'XGB' : XGBClassifier()
        }


for test, clf in models.items():
    clf.fit(x_train, y_train)
    y_pred = clf.predict(x_test)
    acc = accuracy_score(y_test,y_pred)
    train_pred = clf.predict(x_train)
    train_acc = accuracy_score(y_train, train_pred)
    print("\n", test + ' scores')
    print(acc)
    print(classification_report(y_test,y_pred))
    print(confusion_matrix(y_test,y_pred))
    print('*' * 100,"\n")
    model_accuracy = model_accuracy.append({'Model': test, 'Accuracy': acc, 'Train_acc': train_acc}, ignore_index=True)
```

Figure 5.3: **Convolutional Neural Network**

```
plt.scatter(x[y_her == 0, 0], x[y_her == 0, 1], s = 100, c = 'blue', label = 'Cluster 1')
plt.scatter(x[y_her == 1, 0], x[y_her == 1, 1], s = 100, c = 'green', label = 'Cluster 2')
plt.scatter(x[y_her== 2, 0], x[y_her == 2, 1], s = 100, c = 'red', label = 'Cluster 3')
plt.scatter(x[y_her == 3, 0], x[y_her == 3, 1], s = 100, c = 'cyan', label = 'Cluster 4')
plt.title('Clusters of crops')
plt.legend()
plt.show()
```

Figure 5.4: **Cnn Output**

Test result

5.3.2 Integration testing

Input

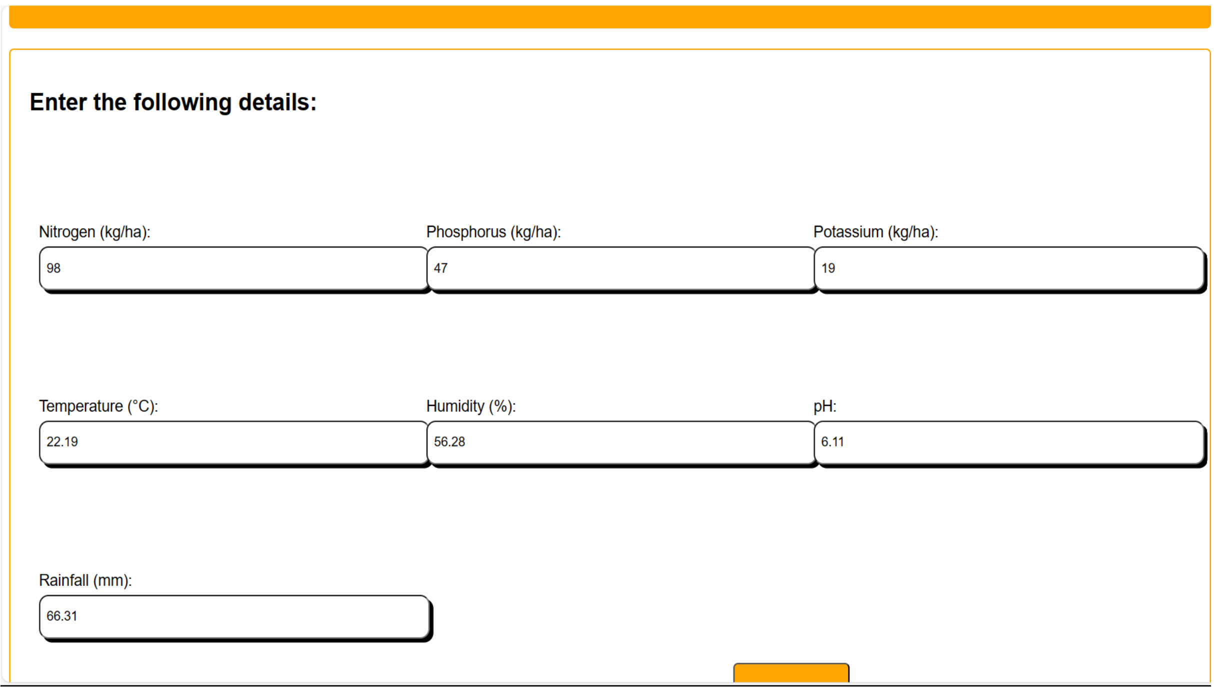

Figure 5.5: **Integration testing**

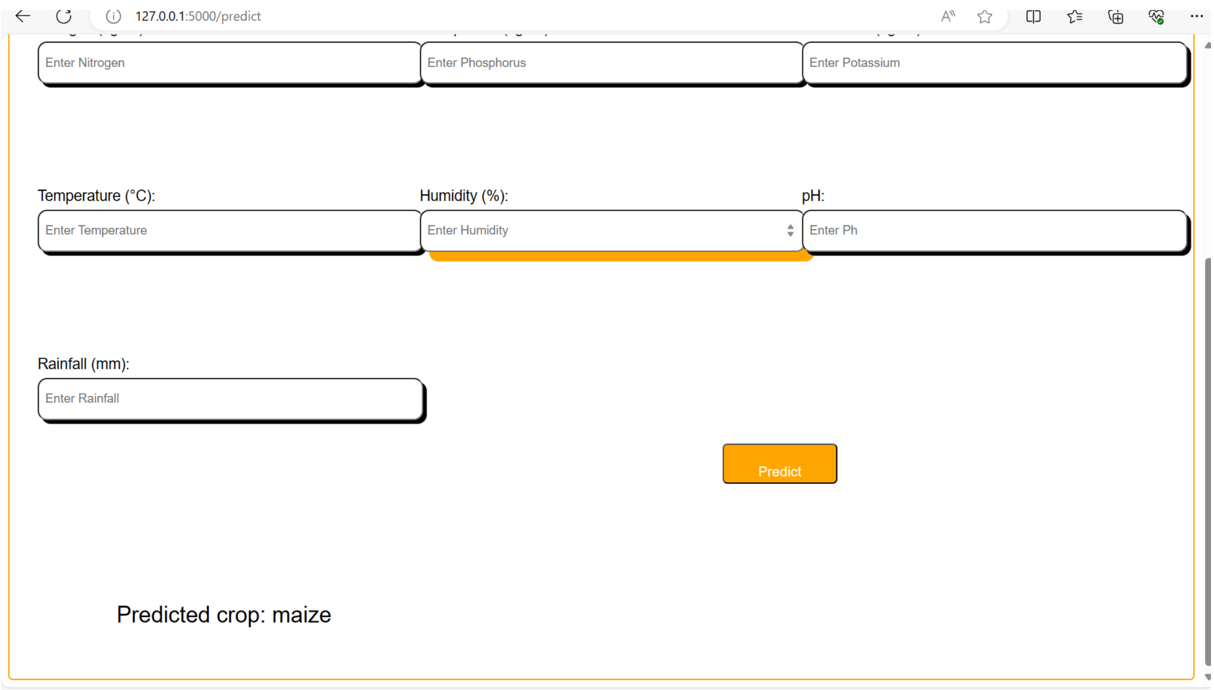

Figure 5.6: **Integration testing output**

Test result

5.3.3 System testing

Input

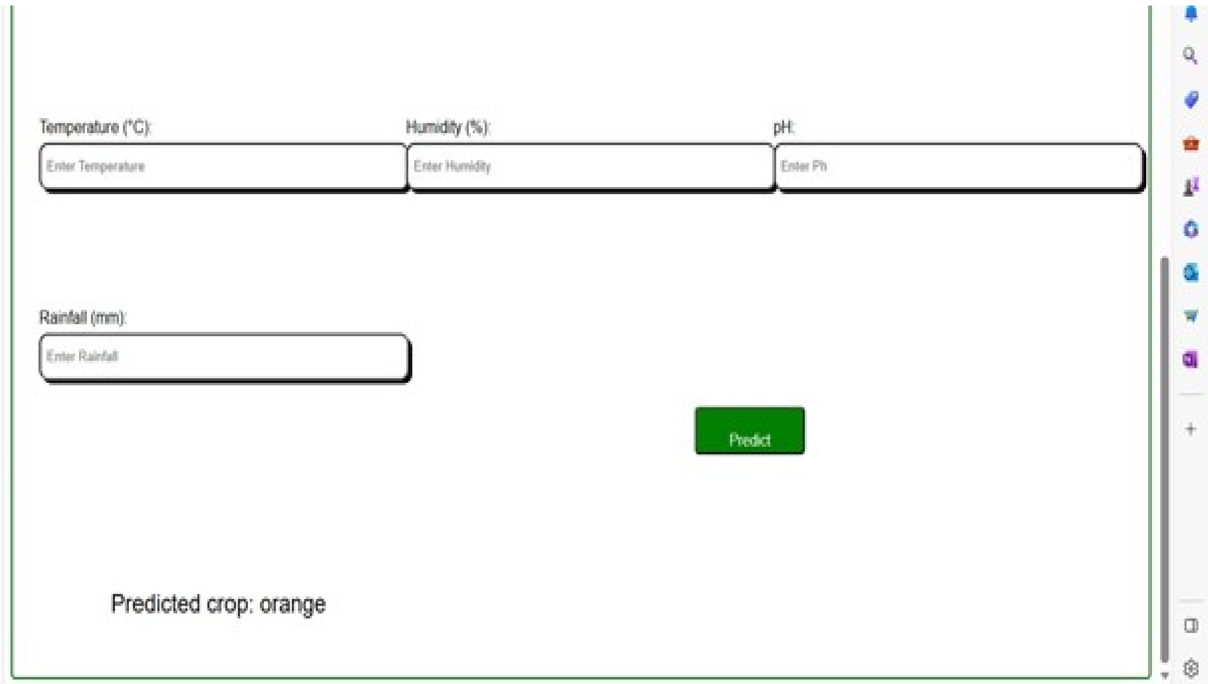

Figure 5.7: **System testing**

5.3.4 Test Result

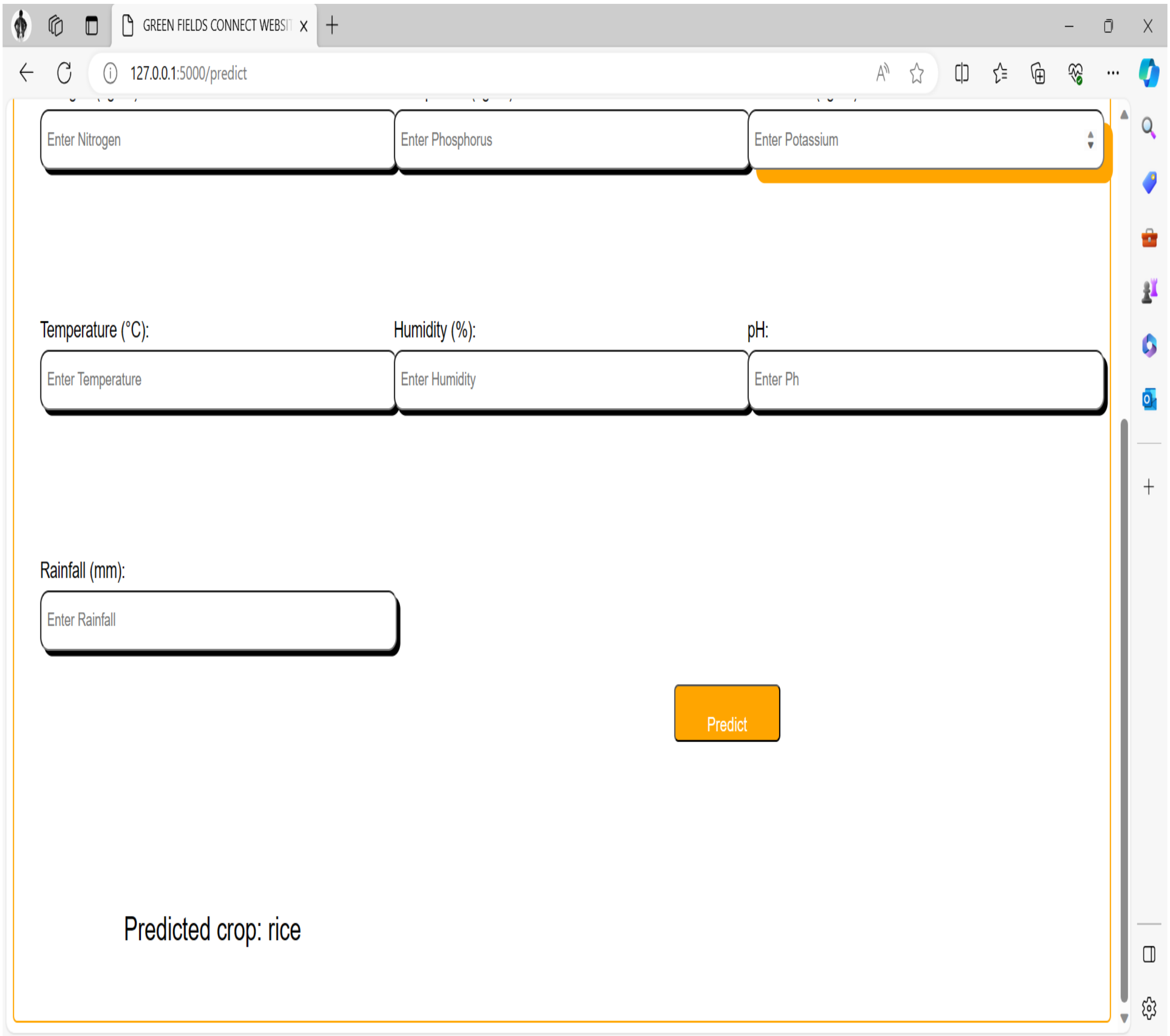

Figure 5.8: **Test Image**

Chapter 6

RESULTS AND DISCUSSIONS

6.1 Efficiency of the Proposed System

6.2 Comparison of Existing and Proposed System

Data Utilization:

Existing Model Relies on traditional statistical methods with limited data utilization. Proposed Model Leverages advanced machine learning techniques, including CNN, for more effective data analysis and crop recommendations.

Accuracy:

Existing Model May have lower accuracy due to simplistic algorithms. Proposed Model Aims for higher accuracy by utilizing ensemble models and deep learning.

Environmental Factors:

Existing Model May consider a subset of environmental factors. Proposed Model Considers a broader range of environmental factors, providing more comprehensive recommendations.

Adaptability:

Existing Model Might struggle to adapt to changing environmental conditions. Proposed Model Designed to be adaptable, incorporating machine learning for dynamic adjustments to varying conditions.

User Interface:

Existing Model Simple interfaces with basic input features. Proposed Model Incorporates a user-friendly interface, allowing users to input a wider array of environmental parameters for more accurate predictions.

Machine Learning Techniques:

Existing Model Primarily uses traditional algorithms like decision trees or rule-based systems. Proposed Model Integrates advanced machine learning techniques such as CNN for improved learning and prediction capabilities.

Scalability:

Existing Model May face challenges in scaling up to handle a large number of users or diverse datasets. Proposed Model Designed with scalability in mind, leveraging modern technologies for efficient scaling.

6.3 Sample Code

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

# Sample dataset (replace this with your actual dataset)
data = {
    'Temperature': [30, 25, 28, 35, 22, 18, 30, 28],
    'Humidity': [70, 80, 75, 65, 85, 90, 80, 75],
    'Rainfall': [10, 5, 8, 12, 3, 2, 9, 7],
    'Crop': ['Wheat', 'Rice', 'Maize', 'Barley', 'Oats', 'Potato', 'Soybean', 'Cotton']
}

df = pd.DataFrame(data)

# Separate features (X) and target variable (y)
X = df[['Temperature', 'Humidity', 'Rainfall']]
y = df['Crop']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create a Decision Tree classifier
model = DecisionTreeClassifier()
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")

# Example: Predict the crop for new environmental conditions
new_data = {
    'Temperature': [25],
    'Humidity': [80],
    'Rainfall': [6]
}

new_df = pd.DataFrame(new_data)

# Make prediction for new conditions
prediction = model.predict(new_df)
print(f"Predicted Crop: {prediction[0]}")
```

Output

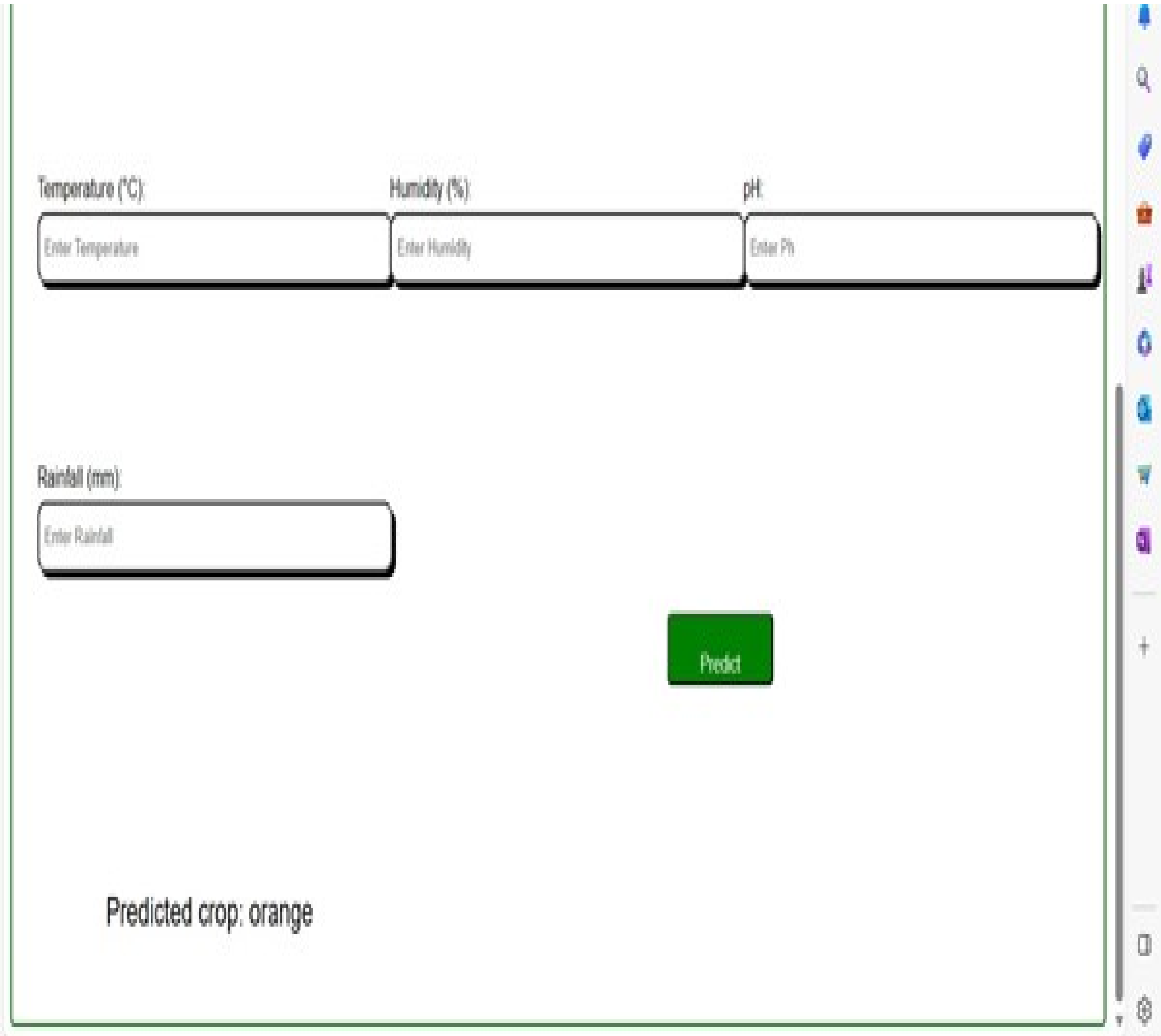

Figure 6.1: **Prediction Output 1**

The Figure 6.1 explains that the models are used to make predictions on new input data. For the web-based application, when a user inputs soil and climate parameters, the models process the data and predict the most suitable crop for cultivation.

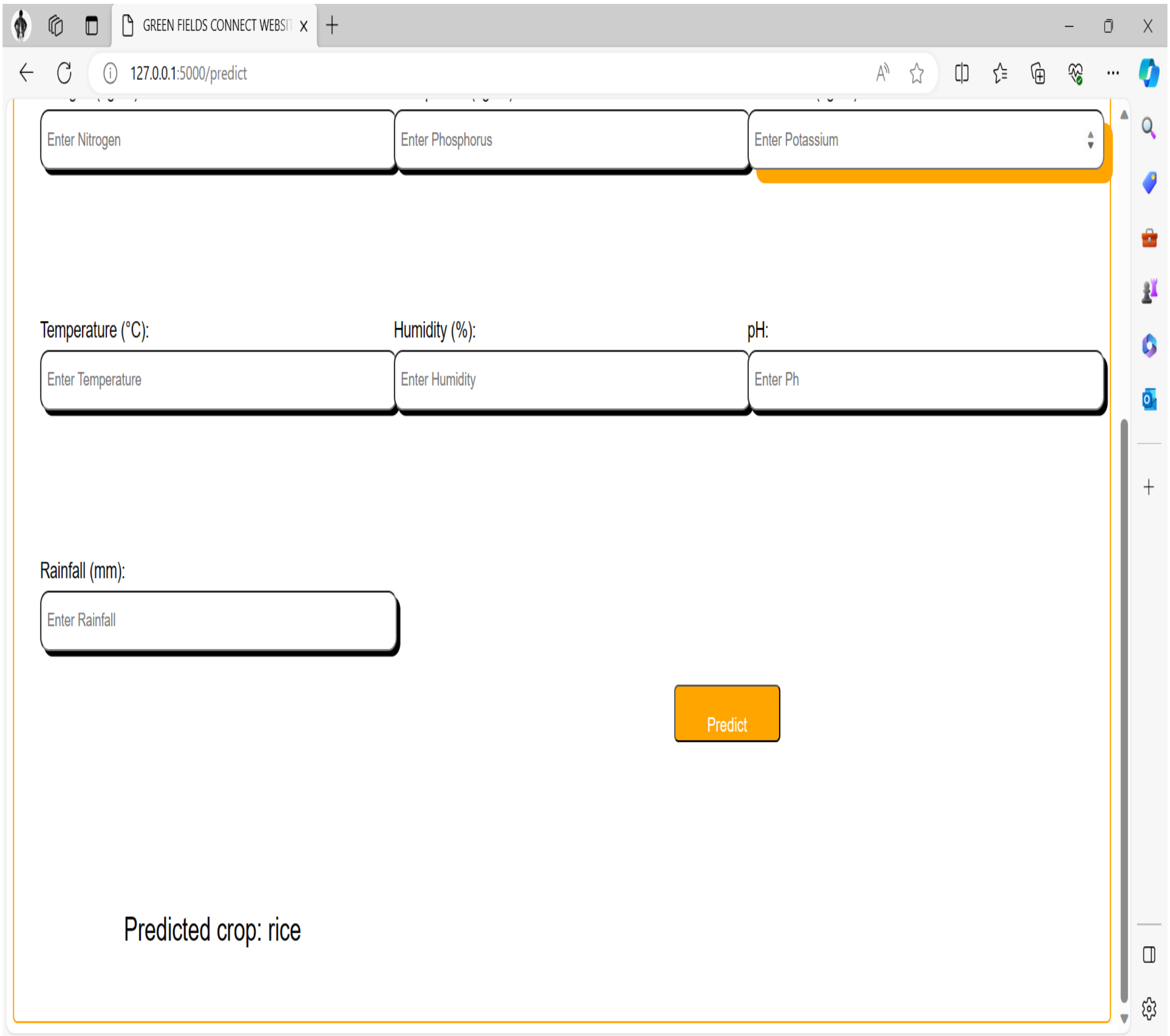

Figure 6.2: **Prediction Output 2**

The Figure 6.2 explains that the models are used to make predictions on new input data. For the web-based application, when a user inputs soil and climate parameters, the models process the data and predict the most suitable crop for cultivation.

Chapter 7

CONCLUSION AND FUTURE ENHANCEMENTS

7.1 Conclusion

In conclusion, this project aims to address the crucial challenges faced by farmers in crop selection by leveraging advanced technologies, specifically machine learning and precision agriculture. By incorporating a Convolutional Neural Network (CNN) and other machine learning techniques, the proposed system endeavors to provide accurate and efficient crop recommendations based on diverse environmental factors. The project's foundation lies in the recognition of the significance of agriculture in the economy and the need to empower farmers with intelligent decision-making tools. Through the integration of diverse technologies, including data mining, data preprocessing, and machine learning algorithms, the system seeks to enhance crop selection processes, ultimately leading to improved agricultural productivity and economic outcomes.

The utilization of a Flask-based web application further extends the project's impact by providing an accessible and user-friendly interface for farmers to input environmental parameters and receive real-time crop recommendations. The web application leverages a trained model, incorporating a CNN, to predict the most suitable crops for a given set of conditions. The system's feasibility is underscored by its ability to adapt to diverse geographical and climatic conditions, providing tailored recommendations for specific regions. The inclusion of a detailed methodology, algorithmic approach, and systematic testing procedures contributes to the robustness and reliability of the proposed crop recommendation system. In essence, this project represents a significant step towards modernizing agriculture, empowering farmers with data-driven insights, and contributing to the sustainable growth of the agricultural sector.

7.2 Future Enhancements

Integration of Remote Sensing Data:
Incorporating remote sensing data can enhance the precision of environmental factors considered in crop recommendations. Satellite imagery and other remote sensing technologies can provide real-time data on soil conditions, vegetation health, and climate, contributing to more accurate predictions.

Dynamic Learning Models:
Implementing dynamic learning models that continuously update and adapt based on new data and environmental changes can improve the system's accuracy over time. This involves incorporating techniques like online learning to accommodate evolving agricultural conditions.

User Feedback Mechanism:
Introducing a user feedback mechanism in the web application can allow farmers to provide insights into the actual outcomes of their crop selections. This feedback loop can be used to refine and improve the recommendation system, making it more responsive to local variations.

Crop Disease Prediction:
Extending the system to predict potential crop diseases based on environmental conditions can further assist farmers in preventive measures. Machine learning models can be trained on historical data to identify patterns associated with specific diseases.

Multi-Crop Recommendation:
Enhancing the system to recommend multiple crops in a rotational pattern can contribute to sustainable farming practices. Consideration of crop rotation can help maintain soil health and reduce the risk of pests and diseases.

Chapter 8

PLAGIARISM REPORT

Plagiarism Detector .net

Jan 07, 2024

Plagiarism Scan Report

6% Plagiarized

94% Unique

Characters:7228	Words:962
Sentences:63	Speak Time: 8 Min

Excluded URL	None

Content Checked for Plagiarism

GREEN FIELDS CONNECT WEBSITE USING DEEP LEARNING Minor project-1 report submitted in partial fulfillment of the requirement for award of the degree of Bachelor of Technology in Computer Science & Engineering By J.ESWAR SAI NITHESH (21UECM0101) (VTU 20324) I.SAI VAMSI (21UECS0234) (VTU 19738) S.TANUJ REDDY (21UECS0584) (VTU 19798) Under the guidance of Dr.P.Siva Prakash,M.E.,Ph.D., Associate Professor ABSTRACT The "Green Fields Connect Website Using Deep Learning" represents a compre- hensive solution for modern agriculture by integrating Convolutional Neural Net- works (CNNs) into a user-friendly web application. This innovative project

Chapter 9

SOURCE CODE & POSTER PRESENTATION

9.1 Source Code

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Green fields connect website using deep learning</title>
    <style>
        body {
            background-image: url('img/crop background image.jpg');
            background-size: cover;
            background-position: cover;
            font-family: sans-serif;
        }

        header {
            background-color: green;
            padding: 20px;
            margin-bottom: 20px;
            border-radius: 5px;
        }

        .title {
            color: white;
            font-size: 2em;
            text-align: center;
        }

        h2 {
            text-align: justify;
        }

        main {
            border: 2px solid green;
            padding: 20px;
            border-radius: 5px;
```

```css
    }

    .form-container {
        display: flex;
        flex-wrap: wrap;
        justify-content: space-between;
        background-image: url('img/crop background image.jpg');
        background-size: cover;
    }

    .form-container > div {
        width: calc(33.33% - 10px);
        margin-bottom: 10px;
        margin-left: 10px;
        margin-top: 90px;
    }

    label {
        display: block;
        margin-bottom: 5px;
    }

    input {
        width: 400px;
        margin-bottom: 10px;
        height: 4vh;
        border-radius: 10px;
        padding: 1vh;
        box-shadow: 4px 4px;
        font-weight: 200px;
    }

    input:hover {
        box-shadow: 10px 10px;
    }

    .recommend-crop {
        clear: both;
        display: block;
        width: 10%;
        margin-top: 5px;
        margin-left: 750px;
        text-align: center;
        padding: 20px 20px 20px 20px;
        border-radius: 5px;
        cursor: pointer;
        background-color: green;
        color: white;
        font-size: 15px;
    }
```

```
        .prediction {
            margin-top: 20px;
            font-size: 1.5em;
            text-align: center;
        }
    </style>
</head>
<body>
<header>
    <h1 class="title">GREEN FIELDS CONNECT WEBSITE USING DEEP LEARNING</h1>
</header>
<main>
    <section class="form-container">
        <h2>Enter the following details:</h2>
        <form action="/predict" method="post">
            <div class="form-container">
                <div>
                    <label for="nitrogen">Nitrogen (kg/ha):</label>
                    <input type="number" placeholder="Enter Nitrogen" name="nitrogen" id="nitrogen"
                        required>
                </div>
                <div>
                    <label for="phosphorus">Phosphorus (kg/ha):</label>
                    <input type="number" placeholder="Enter Phosphorus" name="phosphorus" id="
                        phosphorus" required>
                </div>
                <div>
                    <label for="potassium">Potassium (kg/ha):</label>
                    <input type="number" placeholder="Enter Potassium" name="potassium" id="
                        potassium" required>
                </div>
                <br>
                <div>
                    <label for="temperature">Temperature ( C ):</label>
                    <input type="number" step="0.01" placeholder="Enter Temperature" name="
                        temperature" id="temperature" required>
                </div>
                <div>
                    <label for="humidity">Humidity (%):</label>
                    <input type="number" step="0.01" placeholder="Enter Humidity" name="humidity" id
                        ="humidity" required>
                </div>
                <div>
                    <label for="ph">pH:</label>
                    <input type="number" step="0.01" placeholder="Enter Ph" name="ph" id="ph"
                        required>
                </div>
                <div>
                    <label for="rainfall">Rainfall (mm):</label>
```

```
                        <input type="number" step="0.01" placeholder="Enter Rainfall" name="rainfall" id="rainfall" required>
                    </div>
                </div>
                <input type="submit" class="recommend-crop" value="Predict">
            </form>
            <div class="prediction">
                {% if prediction_text %}
                    <p>{{ prediction_text }}</p>
                {% endif %}
            </div>
        </section>
</main>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>GREEN FIELDS CONNECT WEBSITE USING DEEP LEARNING</title>
    <style>
        body {
            background-image: url('img/crop background image.jpg');
            background-size: cover;
            background-position: cover;
            font-family: sans-serif;
        }

        header {
            background-color: Orange;
            padding: 20px;
            margin-bottom: 20px;
            border-radius: 5px;
        }

        .title {
            color: white;
            font-size: 2em;
            text-align: center;
        }

        h2 {
            text-align: justify;
        }

        main {
            border: 2px solid Orange;
            padding: 20px;
            border-radius: 5px;
```

```
        }

        .form-container {
            display: flex;
            flex-wrap: wrap;
            justify-content: space-between;
            background-image: url('img/crop background image.jpg');
            background-size: cover;
        }

        .form-container > div {
            width: calc(33.33% - 10px);
            margin-bottom: 10px;
            margin-left: 10px;
            margin-top: 90px;
        }

        label {
            display: block;
            margin-bottom: 5px;
        }

        input {
            width: 400px;
            margin-bottom: 10px;
            height: 4vh;
            border-radius: 10px;
            padding: 1vh;
            box-shadow: 4px 4px;
            font-weight: 200px;
        }

        input:hover {
            box-shadow: 10px 10px #ffa500;
        }

        .recommend-crop {
            clear: both;
            display: block;
            width: 10%;
            margin-top: 5px;
            margin-left: 750px;
            text-align: center;
            padding: 20px 20px 20px 20px;
            border-radius: 5px;
            cursor: pointer;
            background-color: Orange;
            color: white;
            font-size: 15px;
        }
```

```
        .prediction {
            margin-top: 20px;
            font-size: 1.5em;
            text-align: center;
        }
    </style>
</head>
<body>
<header>
    <h1 class="title">GREEN FIELDS CONNECT WEBSITE USING DEEP LEARNING</h1>
</header>
<main>
    <section class="form-container">
        <h2>Enter the following details:</h2>
        <form action="/predict" method="post">
            <div class="form-container">
                <div>
                    <label for="nitrogen">Nitrogen (kg/ha):</label>
                    <input type="number" placeholder="Enter Nitrogen" name="nitrogen" id="nitrogen" required>
                </div>
                <div>
                    <label for="phosphorus">Phosphorus (kg/ha):</label>
                    <input type="number" placeholder="Enter Phosphorus" name="phosphorus" id="phosphorus" required>
                </div>
                <div>
                    <label for="potassium">Potassium (kg/ha):</label>
                    <input type="number" placeholder="Enter Potassium" name="potassium" id="potassium" required>
                </div>
                <br>
                <div>
                    <label for="temperature">Temperature ( C ):</label>
                    <input type="number" step="0.01" placeholder="Enter Temperature" name="temperature" id="temperature" required>
                </div>
                <div>
                    <label for="humidity">Humidity (%):</label>
                    <input type="number" step="0.01" placeholder="Enter Humidity" name="humidity" id="humidity" required>
                </div>
                <div>
                    <label for="ph">pH:</label>
                    <input type="number" step="0.01" placeholder="Enter Ph" name="ph" id="ph" required>
                </div>
                <div>
                    <label for="rainfall">Rainfall (mm):</label>
```

```
                    <input type="number" step="0.01" placeholder="Enter Rainfall" name="rainfall" id
                        ="rainfall" required>
                </div>
            </div>
            <input type="submit" class="recommend-crop" value="Predict">
        </form>
        <div class="prediction">
            {% if prediction_text %}
                <p>{{ prediction_text }}</p>
            {% endif %}
        </div>
    </section>
</main>
</body>
</html>
```

9.2 Poster Presentation

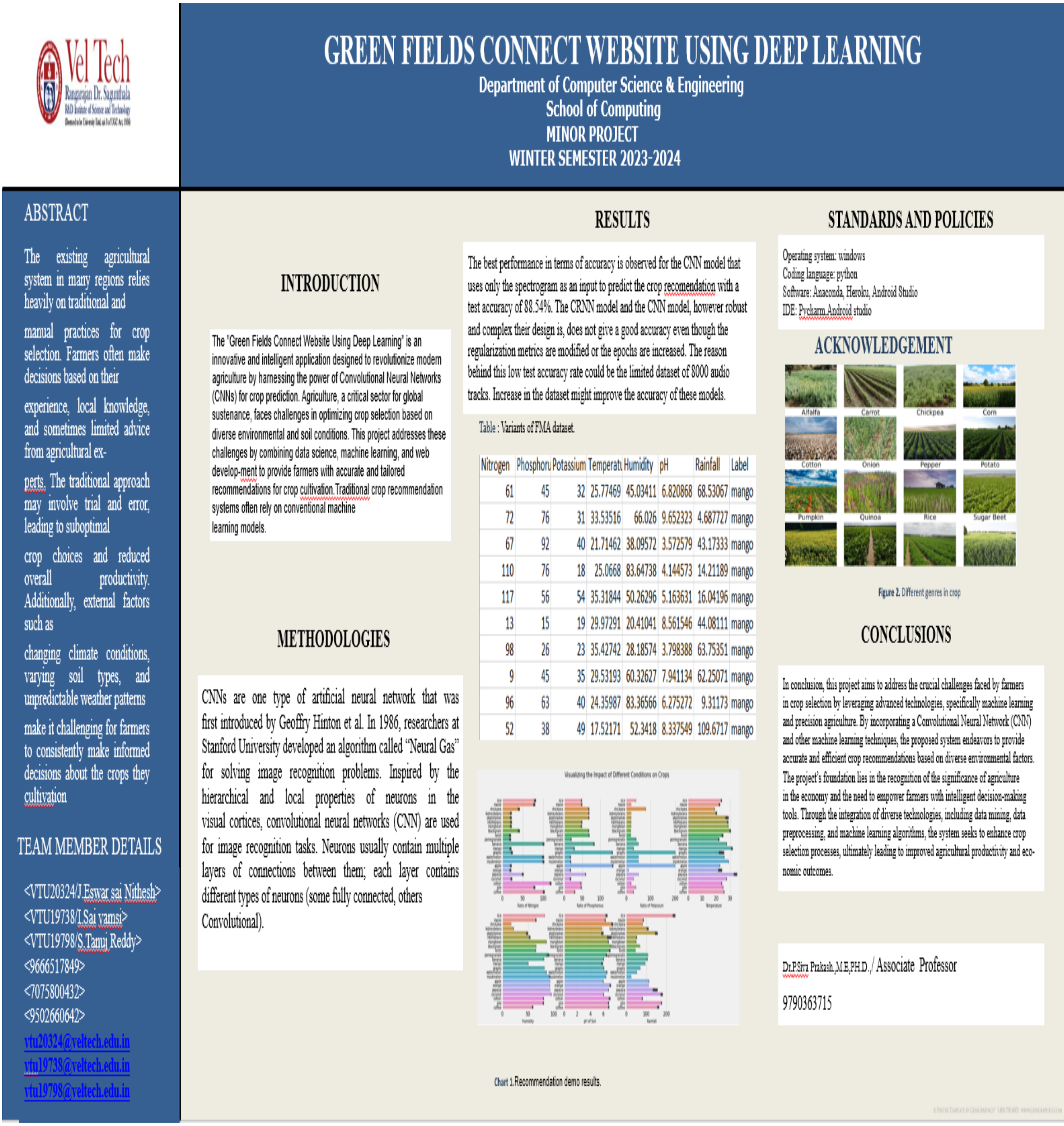

Figure 9.1: **Poster Presentation**

Chapter 10

REFERENCES

[1]A. Motwani, P. Patil, V. Nagaria, S. Verma and S. Ghane, ”Soil Analysis and Crop Recommendation using Machine Learning,” 2022 International Conference for Advancement in Technology (ICONAT), Goa, India, 2022, pp. 1-7, doi: 10.1109/ICONAT53423.2022.9725901.

[2] S. Babu, ”A software model for precision agriculture for small and marginal farmers,” 2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS), Trivandrum, India, 2013, pp. 352-355, doi: 10.1109/GHTC-SAS.2013.6629944.

[3] S. Pudumalar, E. Ramanujam, R. H. Rajashree, C. Kavya, T. Kiruthika and J. Nisha, ”Crop recommendation system for precision agriculture,” 2016 Eighth International Conference on Advanced Computing (ICoAC), Chennai, India, 2017, pp. 32-36, doi: 10.1109/ICoAC.2017.7951740.

[4] A. Savla, N. Israni, P. Dhawan, A. Mandholia, H. Bhadada and S. Bhardwaj, ”Survey of classification algorithms for formulating yield prediction accuracy in precision agriculture,” 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2015, pp. 1-7, doi: 10.1109/ICIIECS.2015.7193120.

[5] A. Manjula and G. Narsimha, ”XCYPF: A flexible and extensible framework for agricultural Crop Yield Prediction,” 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), Coimbatore, India, 2015, pp. 1-5, doi: 10.1109/ISCO.2015.7282311.

[6] R. Kumar, M. P. Singh, P. Kumar and J. P. Singh, ”Crop Selection Method to maximize crop yield rate using machine learning technique,” 2015 International Conference on Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), Avadi, India, 2015, pp. 138-145, doi: 10.1109/ICSTM.2015.7225403.

[7] A. T. M. S. Ahamed et al., ”Applying data mining techniques to predict annual yield of major crops and recommend planting different crops in different districts in Bangladesh,” 2015 IEEE/ACIS 16th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), Takamatsu, Japan, 2015, pp. 1-6, doi: 10.1109/SNPD.2015.7176185.

[8] M. Paul, S. K. Vishwakarma and A. Verma, ”Analysis of Soil Behaviour and Prediction of Crop Yield Using Data Mining Approach,” 2015 International Conference on Computational Intelligence and Communication Networks (CICN), Jabalpur, India, 2015, pp. 766-771, doi: 10.1109/CICN.2015.156.

[9] S. Babu, ”A software model for precision agriculture for small and marginal farmers,” 2013 IEEE Global Humanitarian Technology Conference: South Asia Satellite (GHTC-SAS), Trivandrum, India, 2013, pp. 352-355, doi: 10.1109/GHTC-SAS.2013.6629944.

[10] A. Savla, N. Israni, P. Dhawan, A. Mandholia, H. Bhadada and S. Bhardwaj, ”Survey of classification algorithms for formulating yield prediction accuracy in precision agriculture,” 2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS), Coimbatore, India, 2015, pp. 1-7, doi: 10.1109/ICIIECS.2015.7193120.