Smart Green System

Final Review

Group number:14

Guided by

Mr. Sherikh K K
Assistant Professor
ADS Department

Presented by

Fathimma Sana
Farah Fahmi
Ameen Ali

CONTENTS

INTRODUCTION

- Agriculture faces critical challenges in yield prediction, disease diagnosis, weed management and medicinal plant identification
- These issues directly affect productivity, profitability, and environmental sustainability.
- Inaccurate predictions and delayed detection lead to significant crop losses.
- Advanced machine learning and deep learning models offer transformative solutions.
- Techniques like Naïve Bayes, Random Forest, and CNNs enable precise analysis of field data.
- Early detection and targeted interventions reduce resource waste and environmental impact.

Objective

- Optimize crop management through data-driven, efficient decision-making.

LITERATURE SURVEY

Title	Authors	Methods	Results
Crop Prediction Model Using Machine Learning Algorithms	Ersin Elbasi, Chamseddine Zaki, Ahmet E. Topcu, Wiem Abdelbaki, Aymen I. Zreikat, Elda Cina, Ahmed Shdefat, and Louai Saker	Used Naïve Bayes, Decision Trees, and Random Forest for crop yield prediction based on soil, weather, and crop data.	Improved yield prediction accuracy but needs real-world validation.
Machine Learning Models for Predicting Crop Yield and Disease Detection	Andrew J, Jennifer Eunice , Daniela Elena Popescu , M. Kalpana Chowdary and Jude Hemanth	Applied CNN and LSTM for disease detection and yield prediction with image augmentation and environmental analysis.	Enhanced disease detection accuracy but requires adaptation for different regions.
Weed Species Identification in Different Crops Using Precision Weed Management: A Review	Anand Muni Mishra and Vinay Gautam	Trained deep learning models on labeled datasets using CNNs for weed identification.	Reduced herbicide use but demands high computational power and large datasets.
Deep Learning-Based Medicinal plant identification Using Images for Agricultural Applications	S. Kavitha, T. Satish Kumar,E.Naresh, Vijay H. Kalmani,Kalyan Devappa Bamane,Piyush Kumar Pareek	Used CNN and BPNN models with image preprocessing and augmentation for medicinal plant identification.	Achieved high accuracy and effective model for identification but requires extensive datasets and resources.

PROBLEM STATEMENT

- Crop diseases, inaccurate yield prediction, and weed infestations cause major agricultural losses.
- Traditional detection methods are slow, inefficient, and require expert intervention.
- Machine learning and deep learning improve accuracy but require large datasets and high computational power.
- Existing models struggle to generalize across different crops and environmental conditions.
- A real-time, automated disease detection and yield prediction and medicinal plant identification is essential.
- Integrating CNN and LSTM enhances accuracy and efficiency in crop monitoring.
- This solution helps farmers take timely action, improving productivity and sustainability.

METHODOLOGY

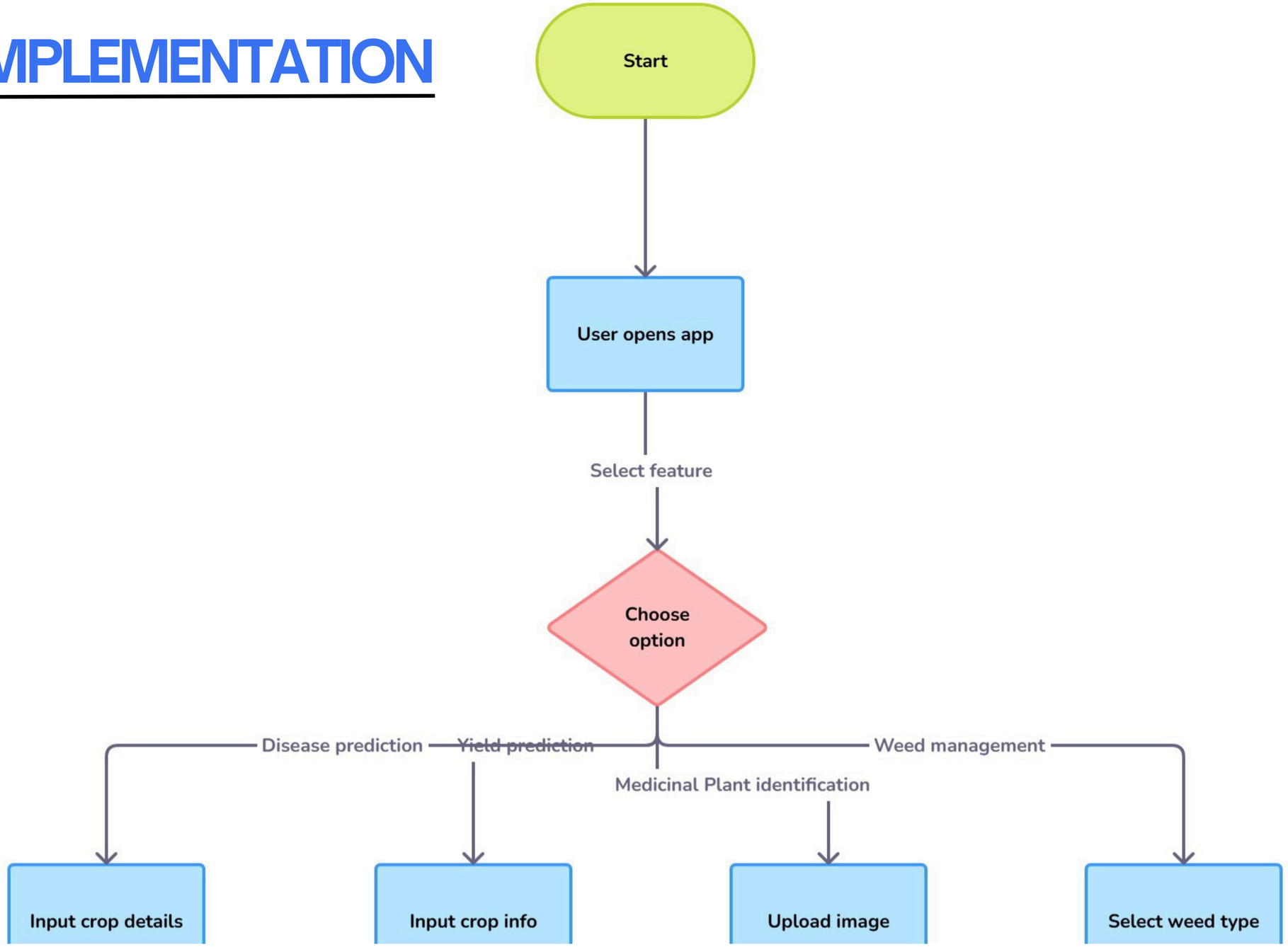
Approaches

- User uploads image and environmental data.
- OpenCV processes image, CNN detects disease.
- LSTM predicts yield, results are displayed.

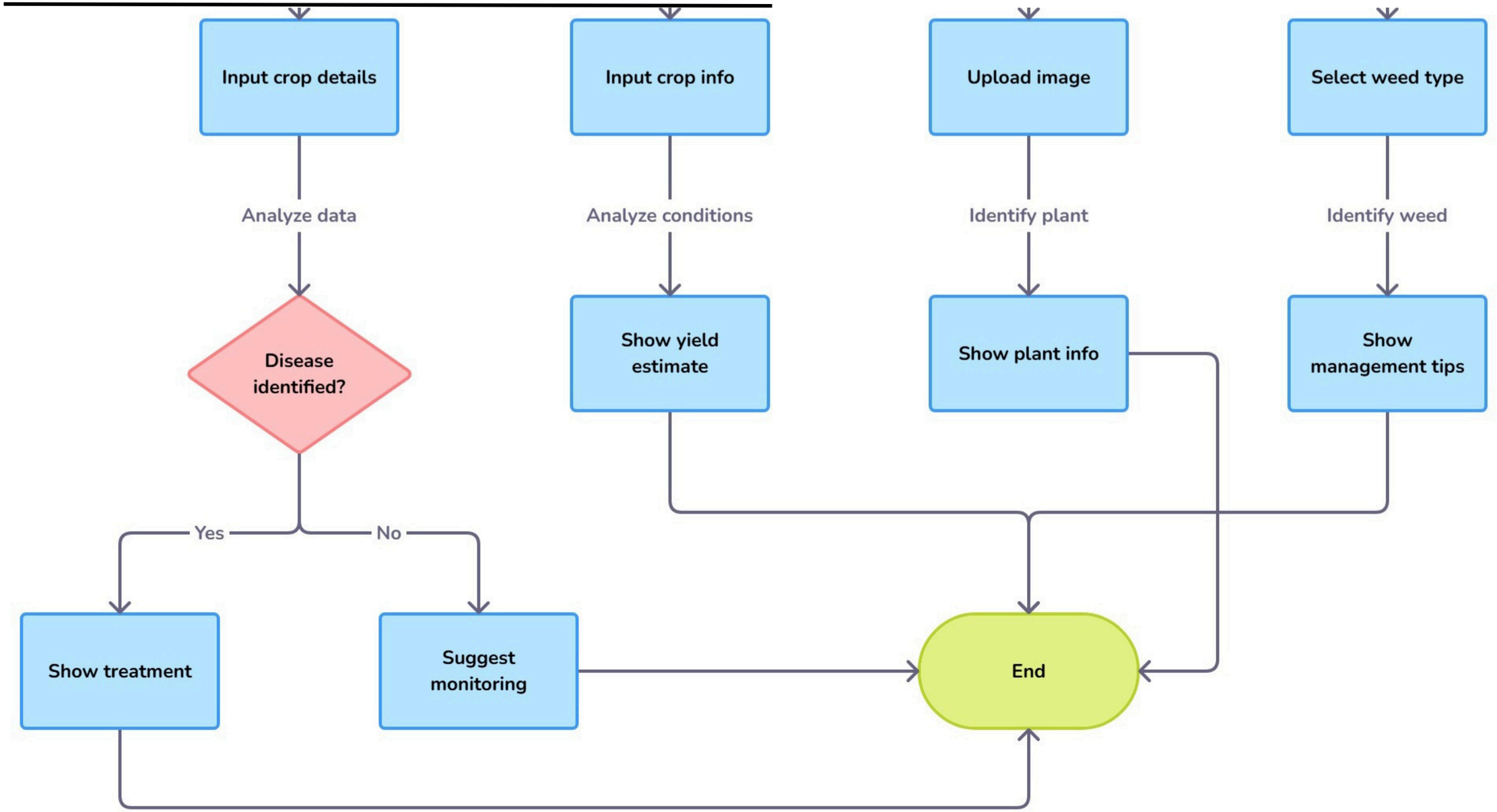
Tools and Software:

- Languages:Python,HTML/CSS, JavaScript.
- Frameworks: Flask, TensorFlow/Keras.
- Libraries: OpenCV, SQLite.

DESIGN AND IMPLEMENTATION



DESIGN AND IMPLEMENTATION



WORKING PRINCIPLE

Identification & Prediction Systems

1.Yield Prediction

- The model uses a neural network to learn patterns from environmental data to classify crops.
- The softmax activation function in the output layer predicts the most suitable crop by assigning probabilities to each class.

2.Crop Disease Detection

- The CNN model analyzes plant leaf images using convolutional layers to detect patterns associated with healthy and infected conditions.
- The final layer uses a sigmoid activation function to classify the image as either "Healthy" or "Infected" based on probability scores.

WORKING PRINCIPLE

Identification & Prediction Systems

3.Weed Identification

- The CNN model extracts features from images using convolutional layers to differentiate between "Weed" and "No Weed" based on learned patterns.
- The final layer uses a sigmoid activation function to classify the image, assigning a probability to determine whether it contains weeds.

4.Medicinal Plant Identification

- The CNN model extracts features from medicinal plant images using convolutional layers, learning patterns specific to each plant type.
- The model processes the input image and predicts the plant type by selecting the class with the highest probability.

EXPERIMENTAL SETUP

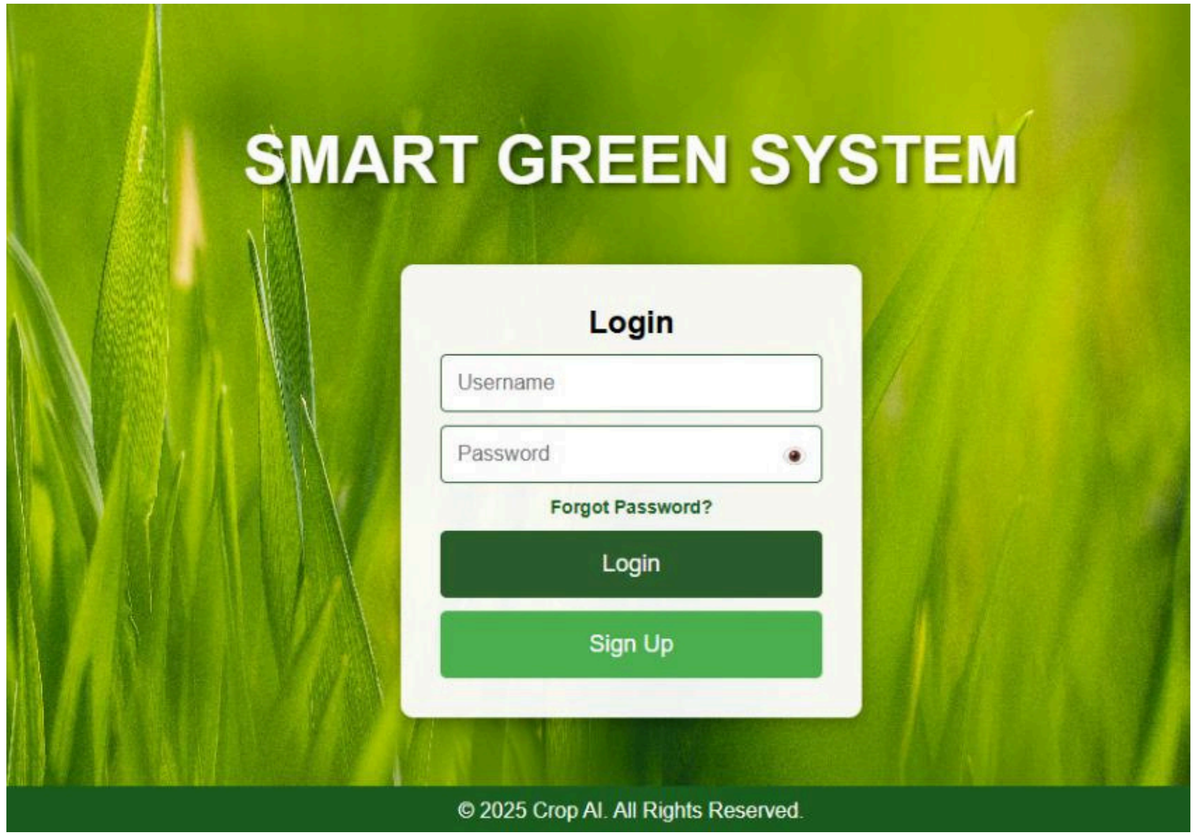
Developing Environment

- Operating System: Windows 10 / Linux (Ubuntu 20.04 recommended for TensorFlow performance)
- Libraries: TensorFlow/Keras, NumPy, Pandas, Matplotlib
- Database: SQLite (to store results and plant data)
- Browser: Google Chrome / Mozilla Firefox (for the web-based interface
- Development Tools: Python (3.9+), Jupyter Notebook, Flask for the front end

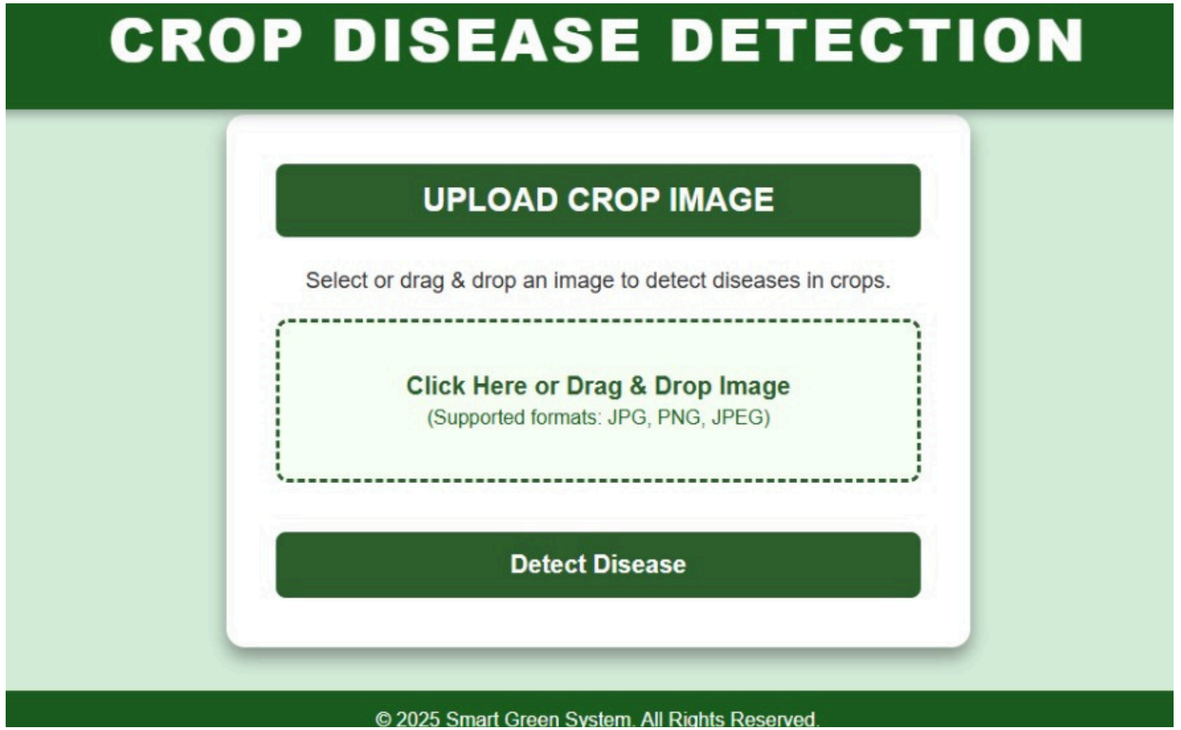
Hardware Setup

- Processor: Intel Core i5/i7 or AMD Ryzen 5/7 (Quad-core or higher)
- RAM: Minimum 8GB (16GB recommended for smoother model training)
- GPU: NVIDIA GTX 1650 or higher (for TensorFlow GPU acceleration)
- Storage: SSD with at least 256GB (faster data processing)
- Camera: 8MP or higher (for capturing plant images)

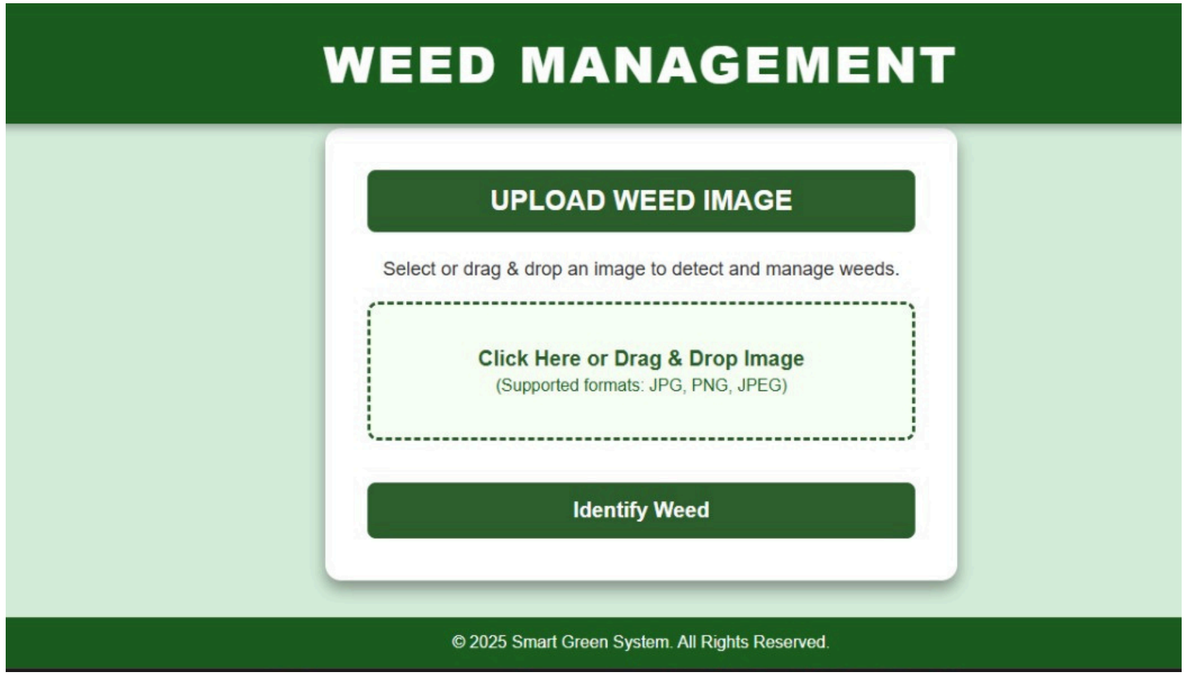
EXPERIMENTAL SETUP

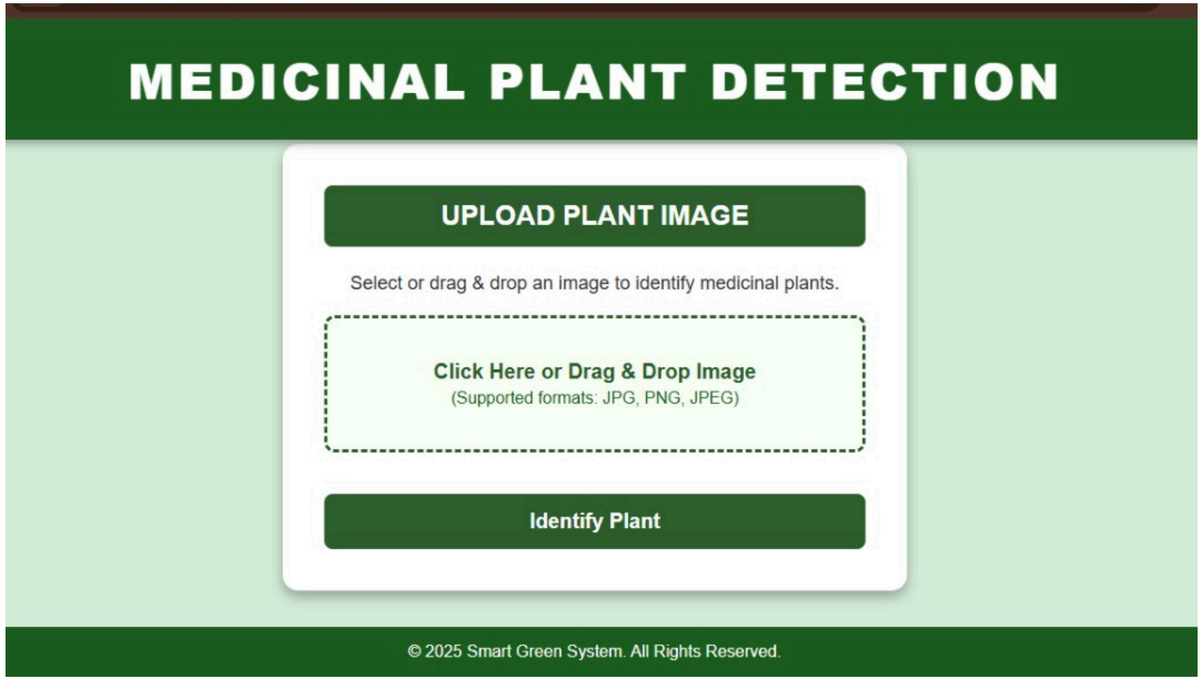


EXPERIMENTAL SETUP



EXPERIMENTAL SETUP





Smart

crop_prediction.py 2

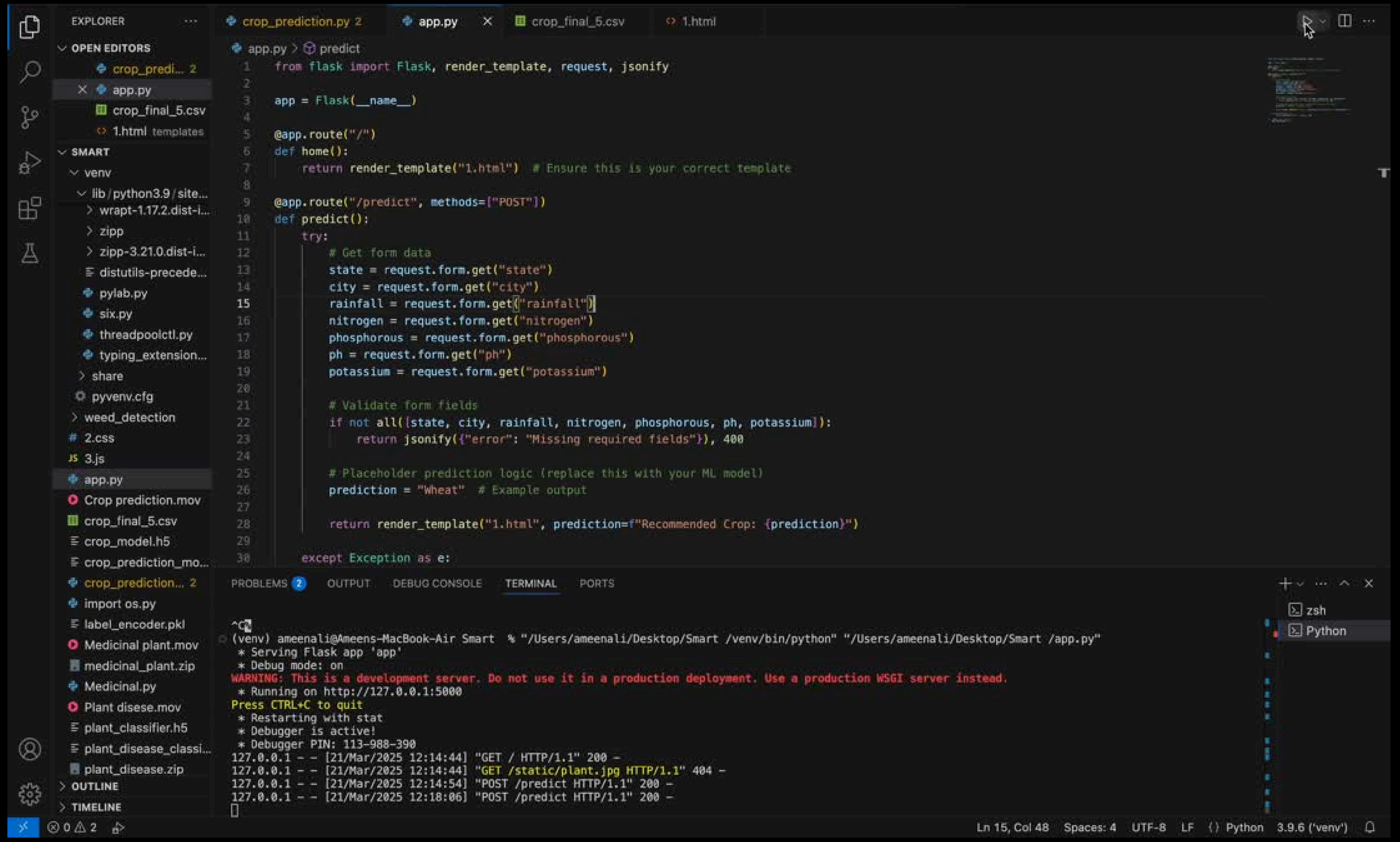
crop_prediction.py > ...

```
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 🔹 Split Dataset (80% Train, 20% Test)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_encoded, test_size=0.2, random_state=42)
print("✅ Data Preprocessing Complete.")

# 🔹 User Inputs Data FIRST
soil_pH = float(input("Enter Soil pH: "))
organic_matter = float(input("Enter Organic Matter (%): "))
nitrogen = float(input("Enter Nitrogen Content (mg/kg): "))
phosphorus = float(input("Enter Phosphorus Content (mg/kg): "))
potassium = float(input("Enter Potassium Content (mg/kg): "))
rainfall = float(input("Enter Rainfall (mm): "))
temperature = float(input("Enter Temperature (°C): "))
yield_tons = float(input("Enter Yield (Tons per Hectare): "))

# 🔹 Define Crop Prediction Model
model = Sequential([
    Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
    Dropout(0.3),
    Dense(32, activation='relu'),
    Dense(len(np.unique(y_encoded)), activation='softmax')  # Classification output
])

# 🔹 Compile and Train Model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=16, verbose=0)
print("✅ Crop Prediction Model Training Complete.")
```

PROBLEMS 2 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

```
(venv) ameenali@Ameens-MacBook-Air Smart  % /usr/bin/python3 /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/printEnvVariablesToFile.py /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/deactivate/zsh/envVars.txt
(venv) ameenali@Ameens-MacBook-Air Smart  % /usr/bin/python3 /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/printEnvVariablesToFile.py /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/deactivate/zsh/envVars.txt
(venv) ameenali@Ameens-MacBook-Air Smart  % python3 crop_prediction.py

/Users/ameenali/Desktop/Smart /venv/lib/python3.9/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
  warnings.warn(
✅ Data Preprocessing Complete.
Enter Soil pH: ^[2
Traceback (most recent call last):
  File "/Users/ameenali/Desktop/Smart /crop_prediction.py", line 35, in <module>
    soil_pH = float(input("Enter Soil pH: "))
ValueError: could not convert string to float: '\x1b2'
(venv) ameenali@Ameens-MacBook-Air Smart  %
```

Python
zsh
zsh


EXPLORER
OPEN EDITORS
crop_prediction.py 2
app.py
crop_final_5.csv
1.html
SMART
venv
lib / python3.9 / site...
wrapt-1.17.2.dist-i...
zipp
zipp-3.21.0.dist-i...
distutils-precede...
pylab.py
six.py
threadpoolctl.py
typing_extension...
share
pyvenv.cfg
weed_detection
2.css
3.js
app.py
Crop prediction.mov
crop_final_5.csv
crop_model.h5
crop_prediction_mo...
crop_prediction... 2
import os.py
label_encoder.pkl
Medicinal plant.mov
medicinal_plant.zip
Medicinal.py
Plant disese.mov
plant_classifier.h5
plant_disease_classi...
plant_disease.zip
OUTLINE
TIMELINE
app.py > predict
from flask import Flask, render_template, request, jsonify
app = Flask(__name__)
@app.route("/")
def home():
return render_template("1.html") # Ensure this is your correct template
@app.route("/predict", methods=["POST"])
def predict():
try:
# Get form data
state = request.form.get("state")
city = request.form.get("city")
rainfall = request.form.get("rainfall")
nitrogen = request.form.get("nitrogen")
phosphorous = request.form.get("phosphorous")
ph = request.form.get("ph")
potassium = request.form.get("potassium")
# Validate form fields
if not all([state, city, rainfall, nitrogen, phosphorous, ph, potassium]):
return jsonify({"error": "Missing required fields"}), 400
# Placeholder prediction logic (replace this with your ML model)
prediction = "Wheat" # Example output
return render_template("1.html", prediction=f"Recommended Crop: {prediction}")
except Exception as e:
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
zsh
Python
^C
(venv) ameenali@Ameens-MacBook-Air Smart % "/Users/ameenali/Desktop/Smart /venv/bin/python" "/Users/ameenali/Desktop/Smart /app.py"
* Serving Flask app 'app'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
* Restarting with stat
* Debugger is active!
* Debugger PIN: 113-988-390
127.0.0.1 - - [21/Mar/2025 12:14:44] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2025 12:14:44] "GET /static/plant.jpg HTTP/1.1" 404 -
127.0.0.1 - - [21/Mar/2025 12:14:54] "POST /predict HTTP/1.1" 200 -
127.0.0.1 - - [21/Mar/2025 12:18:06] "POST /predict HTTP/1.1" 200 -
Ln 15, Col 48
Spaces: 4
UTF-8
LF
Python
3.9.6 ('venv')

crop_prediction.py 2 | Medicinal.py 3

Medicinal.py > ...

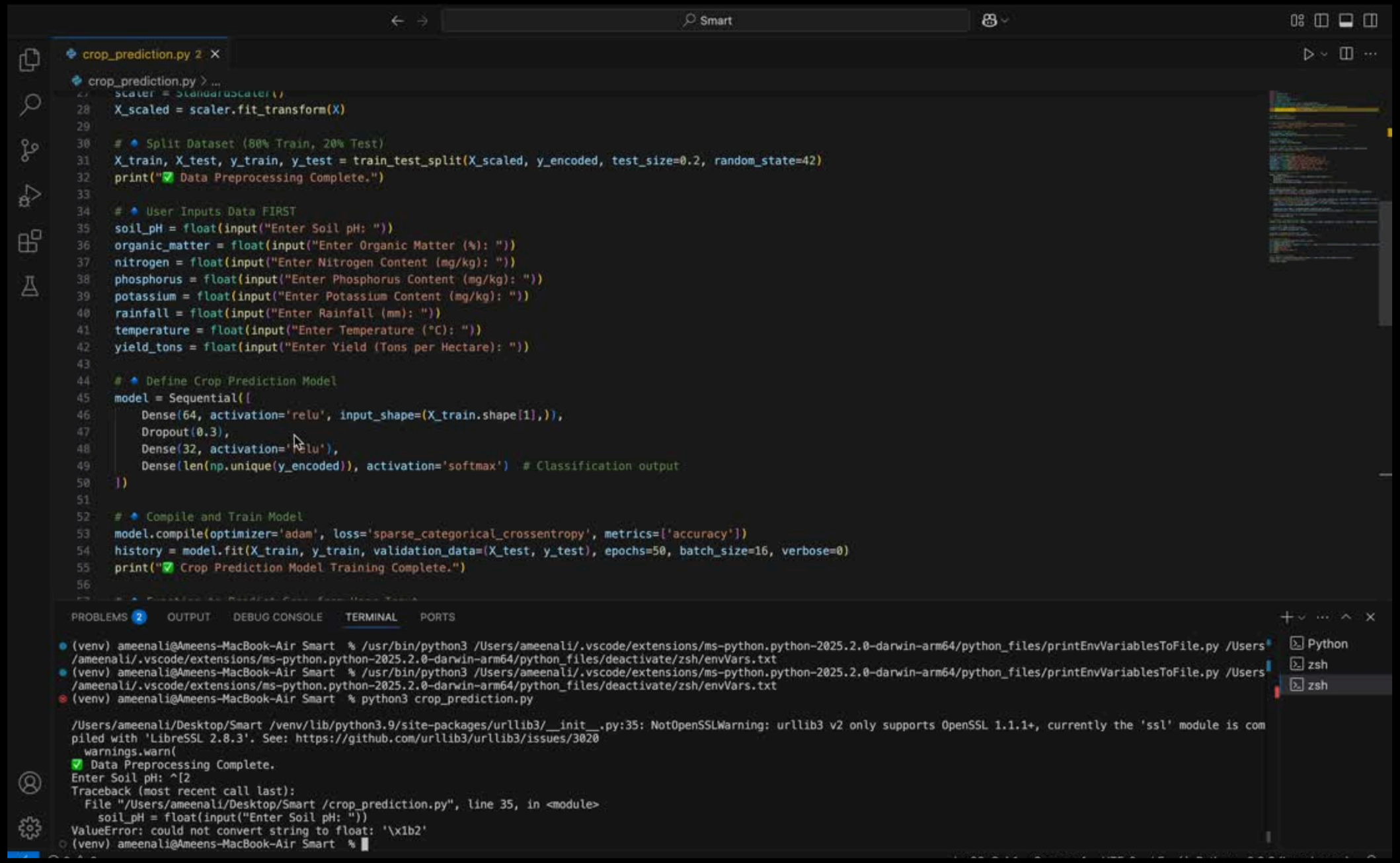
```python
from sklearn.metrics import accuracy_score, confusion_matrix, classification_report
import matplotlib.pyplot as plt

# 🔹 Define Dataset Path
dataset_path = "/Users/ameenali/Desktop/Smart /medicinal_plant1"  # Ensure this contains subfolders 'AloeVera', 'Tulsi', 'Neem'

# 🔹 Image Parameters
img_size = (150, 150)
batch_size = 32

# 🔹 Data Preprocessing
datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)

train_data = datagen.flow_from_directory(dataset_path, target_size=img_size, batch_size=batch_size, class_mode='categorical', subset='training')
val_data = datagen.flow_from_directory(dataset_path, target_size=img_size, batch_size=batch_size, class_mode='categorical', subset='validation')

# 🔹 CNN Model Architecture
model = Sequential([
    Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
    MaxPooling2D(2, 2),
    Conv2D(64, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Conv2D(128, (3, 3), activation='relu'),
    MaxPooling2D(2, 2),
    Flatten(),
    Dense(128, activation='relu'),
    Dropout(0.5),
    Dense(3, activation='softmax')  # 3 classes: Aloe Vera, Tulsi, Neem
])

```

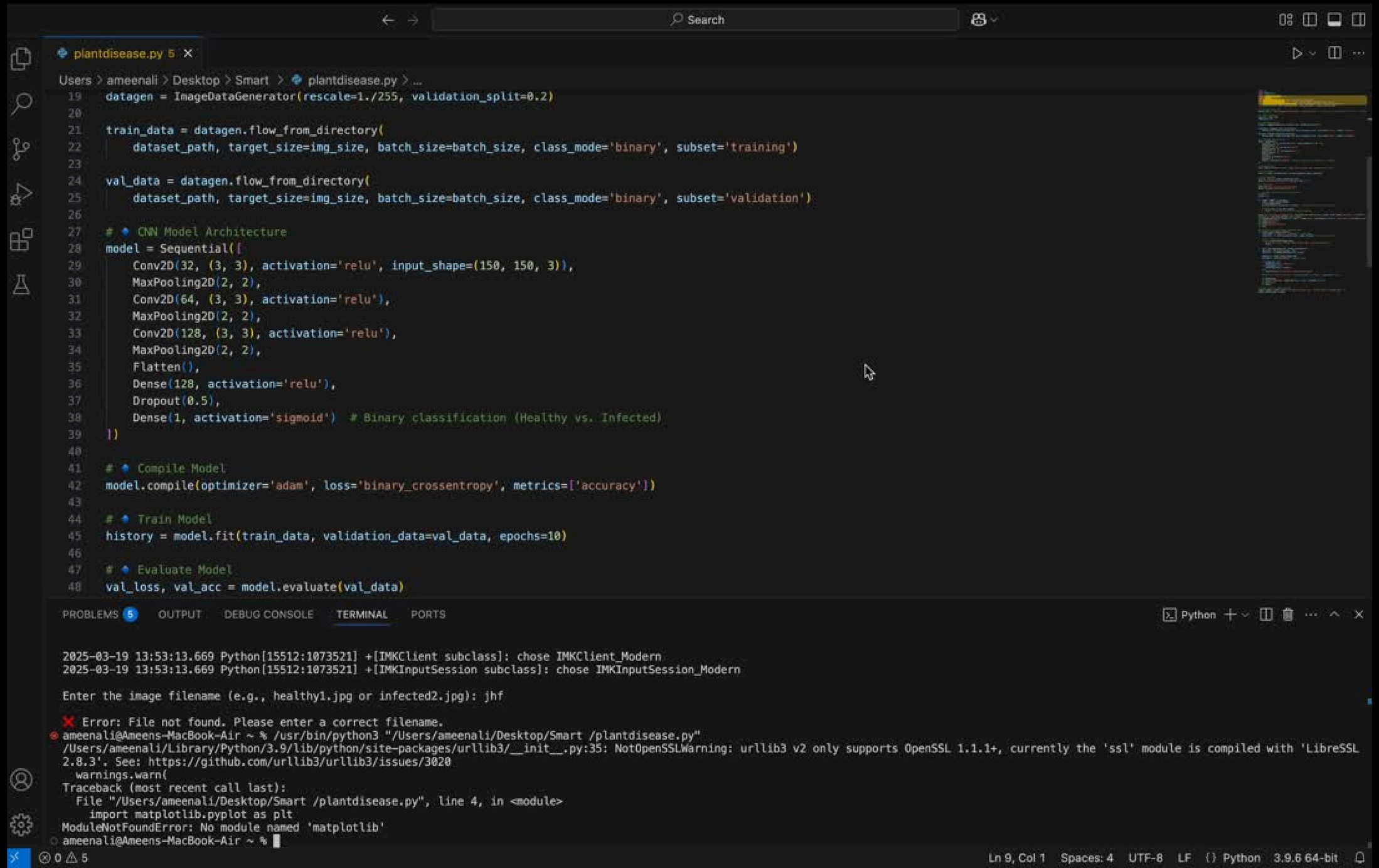
PROBLEMS 5 | OUTPUT | DEBUG CONSOLE | TERMINAL | PORTS

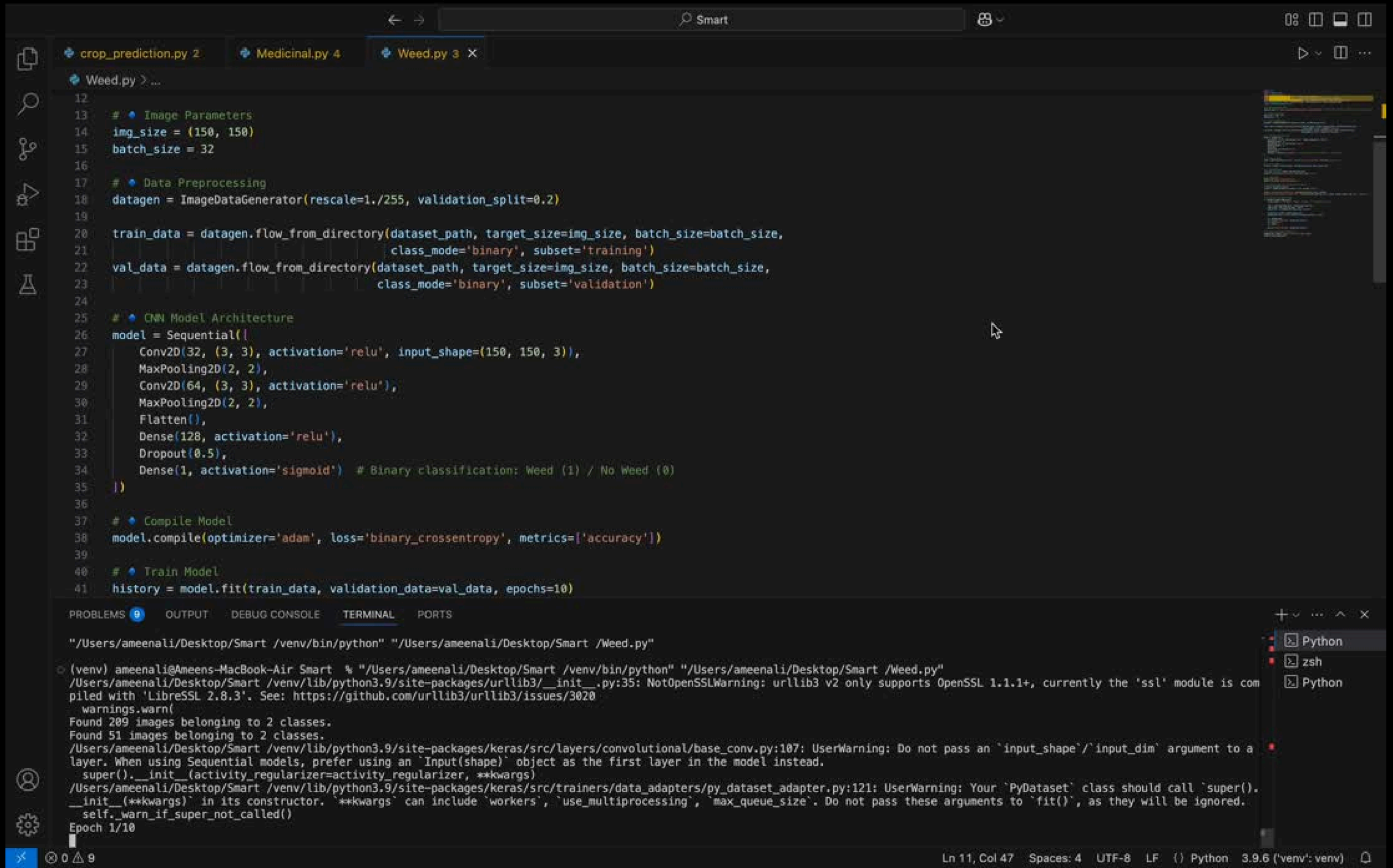
```
7/7 ━━━━━━━━━━━━━━━━━━━━ 1s 194ms/step - accuracy: 0.8328 - loss: 0.3762 - val_accuracy: 0.8077 - val_loss: 0.4907
2/2 ━━━━━━━━━━━━━━━━━━━━ 0s 36ms/step - accuracy: 0.7989 - loss: 0.5161

✅ Validation Accuracy: 80.77%
WARNING:absl:You are saving your model as an HDF5 file via `model.save()` or `keras.saving.save_model(model)`. This file format is considered legacy. We recommend using instead the
native Keras format, e.g. `model.save('my_model.keras')` or `keras.saving.save_model(model, 'my_model.keras')`.

✅ Model Saved Successfully.

Enter the path of the image: /Users/ameenali/Desktop/Smart /medicinal_plant1/Aloe Vera/859.jpg
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 50ms/step
2025-03-19 12:19:49.768 Python[13209:991376] +[IMKClient subclass]: chose IMKClient_Modern
2025-03-19 12:19:49.768 Python[13209:991376] +[IMKInputSession subclass]: chose IMKInputSession_Modern

✅ Predicted Plant: Aloe Vera
(venv) ameenali@Ameens-MacBook-Air Smart %
```

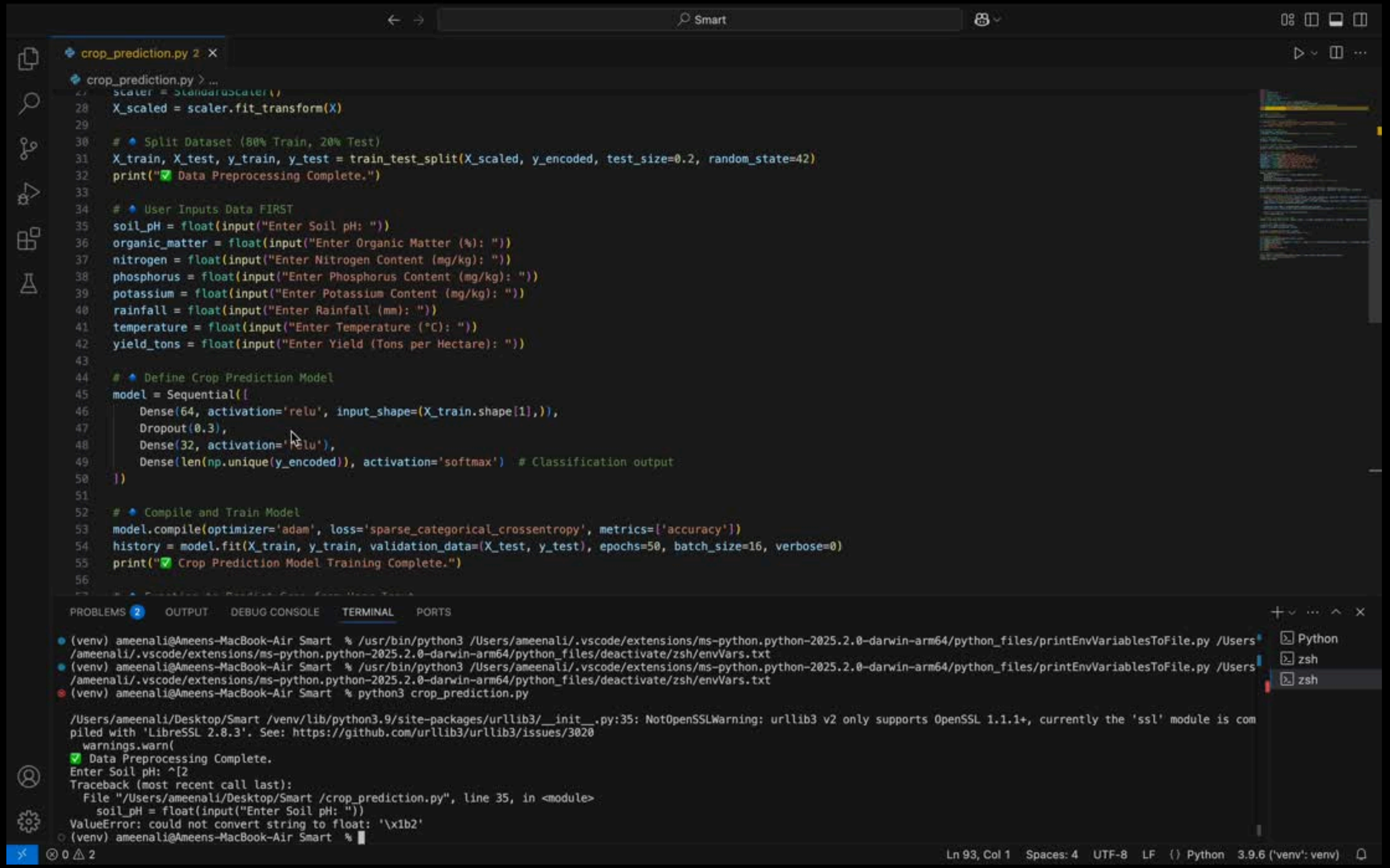
Python | zsh | Python

⊗ 0 ⚠ 5 | Ln 13, Col 64 | Spaces: 4 | UTF-8 | LF | {} Python | 3.9.6 ('venv': venv)


Smart
crop_prediction.py 2
crop_prediction.py > ...
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# 🔹 Split Dataset (80% Train, 20% Test)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_encoded, test_size=0.2, random_state=42)
print("✅ Data Preprocessing Complete.")
# 🔹 User Inputs Data FIRST
soil_pH = float(input("Enter Soil pH: "))
organic_matter = float(input("Enter Organic Matter (%): "))
nitrogen = float(input("Enter Nitrogen Content (mg/kg): "))
phosphorus = float(input("Enter Phosphorus Content (mg/kg): "))
potassium = float(input("Enter Potassium Content (mg/kg): "))
rainfall = float(input("Enter Rainfall (mm): "))
temperature = float(input("Enter Temperature (°C): "))
yield_tons = float(input("Enter Yield (Tons per Hectare): "))
# 🔹 Define Crop Prediction Model
model = Sequential([
Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
Dropout(0.3),
Dense(32, activation='relu'),
Dense(len(np.unique(y_encoded)), activation='softmax') # Classification output
])
# 🔹 Compile and Train Model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=16, verbose=0)
print("✅ Crop Prediction Model Training Complete.")
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
(venv) ameenali@Ameens-MacBook-Air Smart % /usr/bin/python3 /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/printEnvVariablesToFile.py /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/deactivate/zsh/envVars.txt
(venv) ameenali@Ameens-MacBook-Air Smart % /usr/bin/python3 /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/printEnvVariablesToFile.py /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/deactivate/zsh/envVars.txt
(venv) ameenali@Ameens-MacBook-Air Smart % python3 crop_prediction.py
/Users/ameenali/Desktop/Smart /venv/lib/python3.9/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
warnings.warn(
✅ Data Preprocessing Complete.
Enter Soil pH: ^[2
Traceback (most recent call last):
File "/Users/ameenali/Desktop/Smart /crop_prediction.py", line 35, in <module>
soil_pH = float(input("Enter Soil pH: "))
ValueError: could not convert string to float: '\x1b2'
(venv) ameenali@Ameens-MacBook-Air Smart %
Python
zsh
zsh


Search
plantdisease.py 5
Users > ameenali > Desktop > Smart > plantdisease.py > ...
datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
train_data = datagen.flow_from_directory(
dataset_path, target_size=img_size, batch_size=batch_size, class_mode='binary', subset='training')
val_data = datagen.flow_from_directory(
dataset_path, target_size=img_size, batch_size=batch_size, class_mode='binary', subset='validation')
# 🔹 CNN Model Architecture
model = Sequential([
Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
MaxPooling2D(2, 2),
Conv2D(64, (3, 3), activation='relu'),
MaxPooling2D(2, 2),
Conv2D(128, (3, 3), activation='relu'),
MaxPooling2D(2, 2),
Flatten(),
Dense(128, activation='relu'),
Dropout(0.5),
Dense(1, activation='sigmoid') # Binary classification (Healthy vs. Infected)
])
# 🔹 Compile Model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
# 🔹 Train Model
history = model.fit(train_data, validation_data=val_data, epochs=10)
# 🔹 Evaluate Model
val_loss, val_acc = model.evaluate(val_data)
PROBLEMS 5
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
Python
2025-03-19 13:53:13.669 Python[15512:1073521] +[IMKClient subclass]: chose IMKClient_Modern
2025-03-19 13:53:13.669 Python[15512:1073521] +[IMKInputSession subclass]: chose IMKInputSession_Modern
Enter the image filename (e.g., healthy1.jpg or infected2.jpg): jhf
❌ Error: File not found. Please enter a correct filename.
ameenali@Ameens-MacBook-Air ~ % /usr/bin/python3 "/Users/ameenali/Desktop/Smart /plantdisease.py"
/Users/ameenali/Library/Python/3.9/lib/python/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
warnings.warn(
Traceback (most recent call last):
File "/Users/ameenali/Desktop/Smart /plantdisease.py", line 4, in <module>
import matplotlib.pyplot as plt
ModuleNotFoundError: No module named 'matplotlib'
ameenali@Ameens-MacBook-Air ~ %
⊗ 0 ⚠ 5
Ln 9, Col 1
Spaces: 4
UTF-8
LF
{} Python
3.9.6 64-bit


Smart
crop_prediction.py 2
Medicinal.py 4
Weed.py 3
Weed.py > ...
# 🔹 Image Parameters
img_size = (150, 150)
batch_size = 32
# 🔹 Data Preprocessing
datagen = ImageDataGenerator(rescale=1./255, validation_split=0.2)
train_data = datagen.flow_from_directory(dataset_path, target_size=img_size, batch_size=batch_size,
class_mode='binary', subset='training')
val_data = datagen.flow_from_directory(dataset_path, target_size=img_size, batch_size=batch_size,
class_mode='binary', subset='validation')
# 🔹 CNN Model Architecture
model = Sequential([
Conv2D(32, (3, 3), activation='relu', input_shape=(150, 150, 3)),
MaxPooling2D(2, 2),
Conv2D(64, (3, 3), activation='relu'),
MaxPooling2D(2, 2),
Flatten(),
Dense(128, activation='relu'),
Dropout(0.5),
Dense(1, activation='sigmoid') # Binary classification: Weed (1) / No Weed (0)
])
# 🔹 Compile Model
model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
# 🔹 Train Model
history = model.fit(train_data, validation_data=val_data, epochs=10)
PROBLEMS 9
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
"/Users/ameenali/Desktop/Smart /venv/bin/python" "/Users/ameenali/Desktop/Smart /Weed.py"
(venv) ameenali@Ameens-MacBook-Air Smart % "/Users/ameenali/Desktop/Smart /venv/bin/python" "/Users/ameenali/Desktop/Smart /Weed.py"
/Users/ameenali/Desktop/Smart /venv/lib/python3.9/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is com
piled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
warnings.warn(
Found 209 images belonging to 2 classes.
Found 51 images belonging to 2 classes.
/Users/ameenali/Desktop/Smart /venv/lib/python3.9/site-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a
layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
super().__init__(activity_regularizer=activity_regularizer, **kwargs)
/Users/ameenali/Desktop/Smart /venv/lib/python3.9/site-packages/keras/src/trainers/data_adapters/py_dataset_adapter.py:121: UserWarning: Your `PyDataset` class should call `super().
__init__(**kwargs)` in its constructor. `**kwargs` can include `workers`, `use_multiprocessing`, `max_queue_size`. Do not pass these arguments to `fit()`, as they will be ignored.
self._warn_if_super_not_called()
Epoch 1/10
Python
zsh
Python
0 9
Ln 11, Col 47
Spaces: 4
UTF-8
LF
Python
3.9.6 ('venv': venv)

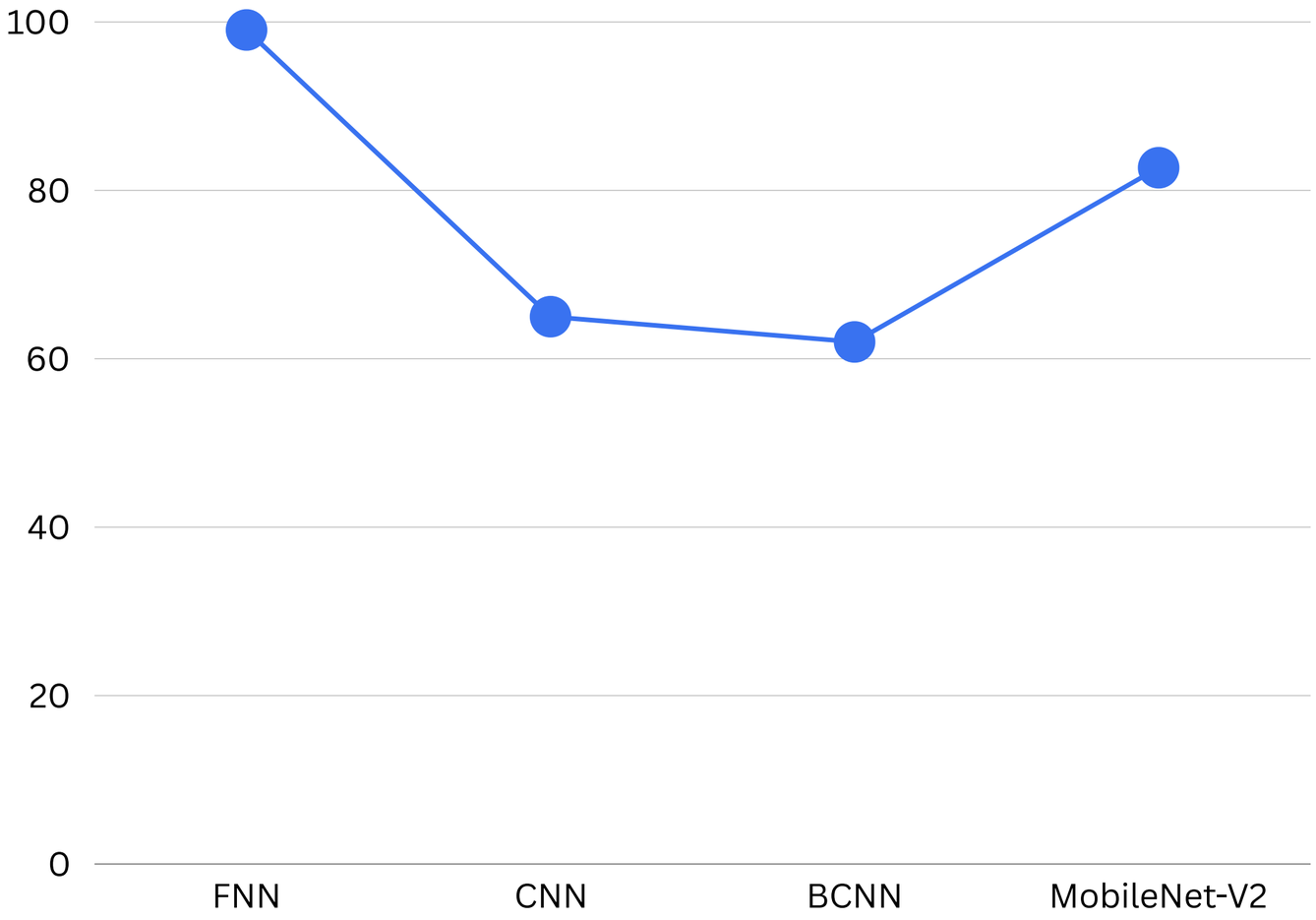

crop_prediction.py 2
crop_prediction.py > ...
X_scaled = scaler.fit_transform(X)
# 🔹 Split Dataset (80% Train, 20% Test)
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y_encoded, test_size=0.2, random_state=42)
print("✅ Data Preprocessing Complete.")
# 🔹 User Inputs Data FIRST
soil_pH = float(input("Enter Soil pH: "))
organic_matter = float(input("Enter Organic Matter (%): "))
nitrogen = float(input("Enter Nitrogen Content (mg/kg): "))
phosphorus = float(input("Enter Phosphorus Content (mg/kg): "))
potassium = float(input("Enter Potassium Content (mg/kg): "))
rainfall = float(input("Enter Rainfall (mm): "))
temperature = float(input("Enter Temperature (°C): "))
yield_tons = float(input("Enter Yield (Tons per Hectare): "))
# 🔹 Define Crop Prediction Model
model = Sequential([
Dense(64, activation='relu', input_shape=(X_train.shape[1],)),
Dropout(0.3),
Dense(32, activation='relu'),
Dense(len(np.unique(y_encoded)), activation='softmax') # Classification output
])
# 🔹 Compile and Train Model
model.compile(optimizer='adam', loss='sparse_categorical_crossentropy', metrics=['accuracy'])
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=16, verbose=0)
print("✅ Crop Prediction Model Training Complete.")
PROBLEMS 2
OUTPUT
DEBUG CONSOLE
TERMINAL
PORTS
(venv) ameenali@Ameens-MacBook-Air Smart % /usr/bin/python3 /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/printEnvVariablesToFile.py /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/deactivate/zsh/envVars.txt
(venv) ameenali@Ameens-MacBook-Air Smart % /usr/bin/python3 /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/printEnvVariablesToFile.py /Users/ameenali/.vscode/extensions/ms-python.python-2025.2.0-darwin-arm64/python_files/deactivate/zsh/envVars.txt
(venv) ameenali@Ameens-MacBook-Air Smart % python3 crop_prediction.py
/Users/ameenali/Desktop/Smart /venv/lib/python3.9/site-packages/urllib3/__init__.py:35: NotOpenSSLWarning: urllib3 v2 only supports OpenSSL 1.1.1+, currently the 'ssl' module is compiled with 'LibreSSL 2.8.3'. See: https://github.com/urllib3/urllib3/issues/3020
warnings.warn(
✅ Data Preprocessing Complete.
Enter Soil pH: ^[2
Traceback (most recent call last):
File "/Users/ameenali/Desktop/Smart /crop_prediction.py", line 35, in <module>
soil_pH = float(input("Enter Soil pH: "))
ValueError: could not convert string to float: '\x1b2'
(venv) ameenali@Ameens-MacBook-Air Smart %
Python
zsh
zsh
Ln 93, Col 1
Spaces: 4
UTF-8
LF
Python
3.9.6 ('venv': venv)

RESULTS & ANALYSIS

TOPIC	MODEL	ACCURACY(%)	PRECISION(%)	RECALL(%)	F1 SCORE (%)
Crop Prediction	fully connected feedforward neural network (FNN)	99.05	99.02	99.05	99.03
Medicinal Plant Identification	Convolutional Neural Network (CNN)	82.69	82	81	81
Disease Detection	Binary Classification CNN (Convolutional Neural Network)	62	60	63	60
Weed Detection	Transfer Learning CNN using MobileNetV2	65	64	61	63

RESULTS & ANALYSIS

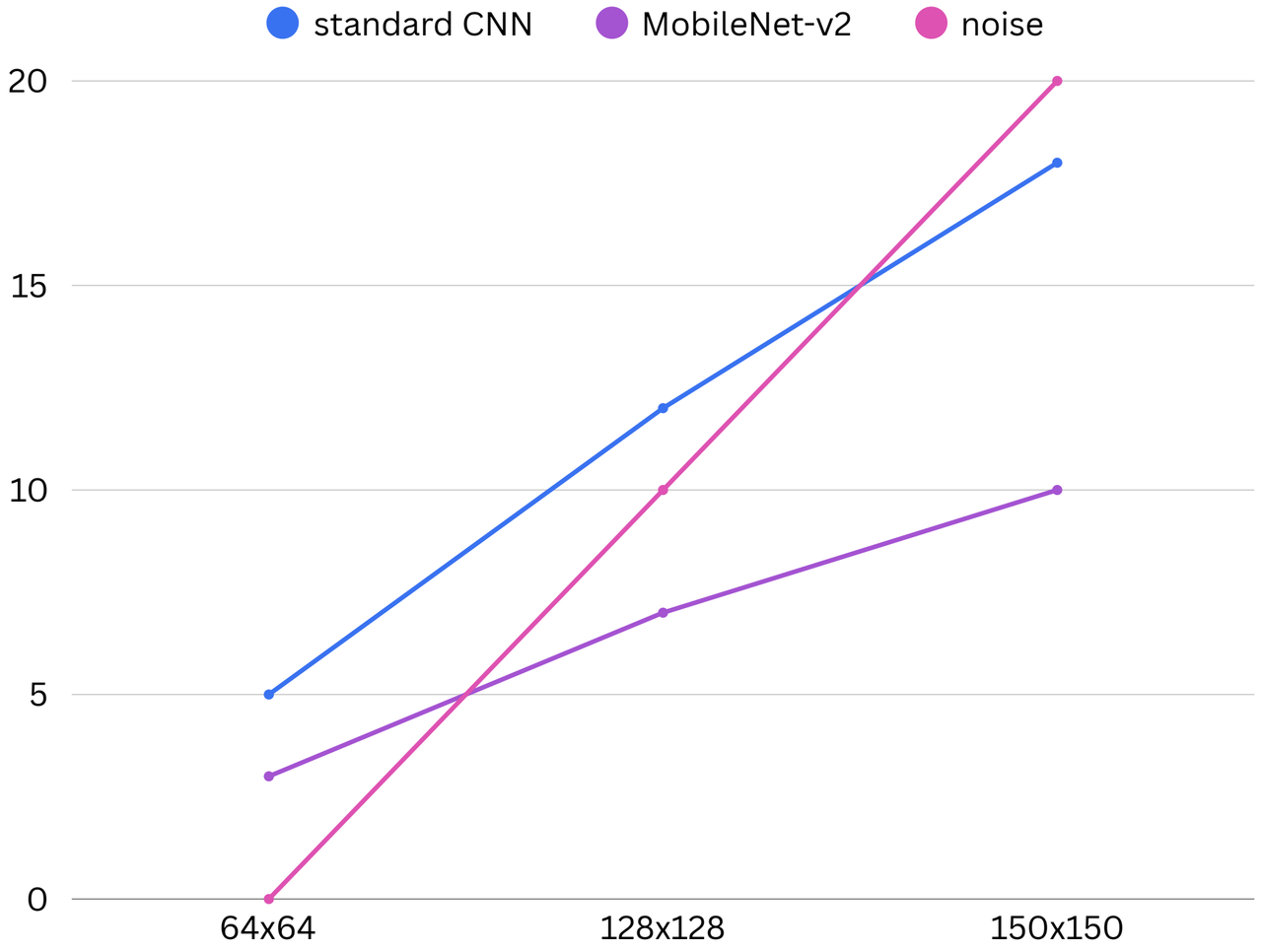
Accuracy comparision



- This graph shows the comparison of accuracy performance across different models. The FNN model achieves the highest accuracy, while CNN and BCNN have lower accuracy levels. MobileNetV2 performs better than CNN and BCNN, highlighting the advantage of transfer learning.

RESULTS & ANALYSIS

Processing Speed by Image Resolution & Noise (Image Resolution vs Processing Time Graph)



The graph compares Processing Speed by Image Resolution for Standard CNN (blue) vs. MobileNet-v2 (purple):

- X-axis: Image resolutions (64x64, 128x128, 150x150)
- Y-axis: Processing time (seconds)
- Noise increases processing time

RESULTS & ANALYSIS

Observation and Trends on the Result

1. Model Performance:

- The models achieve moderate to high accuracy (62%–99%) based on the dataset and architecture used.
- Transfer learning models (e.g., MobileNetV2) improve performance over traditional CNNs by extracting better features.

2. Potential Challenges:

- High-resolution images increase processing time, affecting real-time predictions.
- Class imbalance in datasets may lead to biased predictions, requiring data augmentation or rebalancing techniques.

RESULTS & ANALYSIS

Observation and Trends on the Result

3. User Interaction & Experience:

- The models provide real-time image classification with visual feedback through Matplotlib.
- Simple command-line input for image selection makes the system accessible but could be improved with a GUI for better usability.

4. Accuracy Trends:

- Accuracy generally improves with higher image resolution, but excessive resolution slows down processing.
- Transfer learning models consistently outperform basic CNNs, especially in low-data scenarios.

CONCLUSION

Key Findings

- CNN-based models effectively classify crops, plant diseases, and weeds with accuracy ranging from 62% to 99%.
- Higher image resolution enhances confidence but increases processing time.

Project Contributions

- Enhances user interaction through real-time image predictions and confidence scores.
- Implements automated weed detection and disease identification, aiding precision farming.

Limitations

- Processing time increases for high-resolution images, impacting real-time applications.
- Model accuracy depends on dataset quality, and imbalanced data can lead to biased predictions.

FUTURE SCOPE

1. Enhancing Dataset Diversity

- Expanding the training dataset with images from different regions, seasons, and plant growth stages can improve model generalization.

2. Integration with Augmented Reality (AR)

- AR-based plant identification can allow users to scan plants in real-time with detailed overlays of medicinal properties.

3. Mobile and Edge AI Optimization

- Implementing lightweight models (e.g., MobileNet, EfficientNet) for real-time identification on smartphones and low-power devices.

FUTURE SCOPE

4. Multilingual Support & Voice Assistance

- Adding multilingual support for plant information to make the system accessible to a broader audience.

5. IoT-Based Smart Farming Integration

- Using environmental sensors to enhance plant classification based on soil quality, humidity, and temperature.

6. Collaboration with Botanical Research Centers

- Partnering with research institutions to continuously improve the model and validate new medicinal plant species.

REFERENCES

1.Crop Prediction Model Using Machine Learning Algorithms

- [1] Elbasi E., [2] Zaki C., [3] Topcu A. E., [4] Abdelbaki W., [5] Zreikat A. I., [6] Cina E., [7] Shdefat A., [8] Saker L. (2023). Applied Sciences, 13, 9288. https://www.mdpi.com/2076-3417/13/16/9288

2.Deep Learning-Based Leaf Disease Detection in Crops Using Images for Agricultural Applications

- [1] Andrew J., [2] Eunice J., [3] Popescu D. E., [4] Chowdary M. K., [5] Hemanth J. (2022). Agronomy, 12, 2395. https://www.mdpi.com/2073-4395/12/10/2395

3.Weed Species Identification in Different Crops Using Precision Weed Management: A Review

- [1] Anand Muni Mishra, [2] Vinay Gautam. Weed Species Identification in Different Crops using Precision Weed Management: A Review.https://www.researchgate.net/publication/349463320_Weed_Species_Identification_in_Different_Crops_using_Precision_Weed_Management_A_Review

4.Medicinal Plant Identification in Real-Time Using Deep Learning Model

- [1] Kavitha S., [2] Kumar T. S., [3] Naresh E., [4] Kalmani V. H., [5] Bamane K. D., [6] Pareek P. K. (2024). SN Computer Science, 5, 73. https://doi.org/10.1007/s42979-023-02398-5

THANK YOU !

Any Queries?