Robotics project

Smart Agriculture Monitoring System:

Problem Statement:

Farmers often lack real-time insights into their crops' health and environmental conditions, leading to suboptimal yield. Create a machine-learning model that analyzes data from sensors (e.g., soil moisture, temperature, humidity) and satellite imagery to provide actionable recommendations for irrigation, fertilization, and pest control.

Objective: Increase agricultural productivity and sustainability by offering a smart monitoring system that helps farmers make data-driven decisions for their crops.

Solution:

To address the problem statement and achieve the objective of increasing agricultural productivity and sustainability, here are the ways

1. **Data Collection**: Gathering data from various sources such as sensors (soil moisture, temperature, humidity) and satellite imagery. Ensure the data is comprehensive and covers different aspects of crop health and environmental conditions.

2. **Data Preprocessing**: Cleaning the data to handle missing values, outliers, and inconsistencies. or scale the data as required for model training.

3. **Feature Engineering**: Extracting relevant features from the data that are crucial for predicting crop health and environmental conditions. This might include derived features from sensor readings or spectral indices from satellite imagery.

4. **Machine Learning Model Selection**: Choosing appropriate machine learning models based on the nature of your data and the prediction tasks.

-Regression Models: It helps us to Predict soil moisture and temperature.

- Classification Models: It helps to Identify pest presence or crop diseases.

- Clustering Algorithms: It is used to Group areas with similar crop conditions for targeted interventions.

5.Model Training: Training the selected models using historical data. Using techniques like cross-validation to tune hyperparameters to ensure robust performance.

6. **Real-Time Monitoring and Recommendation**s: Implementation a system that continuously monitors sensor data and satellite imagery in real-time. Using the trained models to provide actionable recommendations such as:

- When and how much to irrigate based on soil moisture and weather forecasts.
- Optimal timing and type of fertilization based on nutrient levels and crop growth stages.
- Early detection of pests or diseases using image analysis and recommend appropriate control measures.

7. **Integration and Deployment**: Integrate the developed models into a user-friendly interface accessible to farmers. It Ensure scalability and reliability of the system for different farm sizes and locations.

8. **Evaluation and Feedback Loop**: Regularly evaluate the performance of the system against ground truth data and farmer feedback. Continuously improve the models based on new data and evolving agricultural practices.

By following these steps, we can develop a robust smart agriculture monitoring system that empowers farmers with real-time insights and data-driven recommendations, ultimately leading to improved crop yields and sustainability.

Implementation using Machine Learning models:

1. **Data Collection** :

Gathering sensor data in CSV format (`sensor_data.csv`) and satellite imagery (handled separately):

Code:

```
Import pandas as pd
```

```
# Load sensor data
Sensor_data = pd.read_csv('sensor_data.csv')
# Load satellite imagery (if applicable)
# satellite_data = load_satellite_data()
```

Output:Sensor Data:

	Timestamp	Soil_Moisture	Temperature	Humidity
0	2024-01-01	0.250	25.0	70.0
1	2024-01-02	0.280	24.5	68.0
2	2024-01-03	0.300	26.0	72.0
3	2024-01-04	0.320	25.5	69.5
4	2024-01-05	0.310	24.0	65.0

2. Data Preprocessing

Clean and preprocess the data:

Code:

```
From sklearn.preprocessing import StandardScaler From sklearn.impute import SimpleImputer
# Handle missing values
Imputer = SimpleImputer(strategy='mean')
Sensor_data_clean = imputer.fit_transform(sensor_data)
# Scale data
Scaler = StandardScaler()
Sensor_data_scaled = scaler.fit_transform(sensor_data_clean)
```

Output:

Preprocessed and Scaled Sensor Data:

```
[[-1.20512453  0.          0.65079137]
 [-0.77256512 -0.70710678  0.12216944]
 [-0.43869648  0.70710678  1.1794133 ]
 [-0.10482783  0.35355339  0.37257592]
 [-0.27176116 -1.41421356 -0.91497048]]
```

3.Feature Engineering

Extract relevant features:

Coding:

```
# Assuming feature extraction function calculate_ndvi() for demonstration Def
calculate_ndvi(data):
   # Dummy function, replace with actual feature extraction logic
   Return [0.2, 0.3, 0.4, 0.25, 0.35]  # Example NDVI values

Sensor_data['NDVI'] = calculate_ndvi(sensor_data)

# Display updated sensor data with NDVI
Print("\nSensor Data with NDVI:")
Print(sensor_data.head())
```

Output:

Sensor Data with NDVI:

	Timestamp	Soil_Moisture	Temperature	Humidity	NDVI
0	2024-01-01	0.250	25.0	70.0	0.2
1	2024-01-02	0.280	24.5	68.0	0.3
2	2024-01-03	0.300	26.0	72.0	0.4
3	2024-01-04	0.320	25.5	69.5	0.25
4	2024-01-05	0.310	24.0	65.0	0.35

4. Machine Learning Model Selection

Choose and train models:

Code:

```
From sklearn.model_selection import train_test_split
```

```
From sklearn.ensemble import RandomForestRegressor

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(sensor_data_scaled, target_variable, test_size=0.2,
random_state=42)

# Example: Train a RandomForestRegressor
Model = RandomForestRegressor(n_estimators=100, random_state=42)
Model.fit(X_train, y_train)
```

Output:

Model Performance:

Mean Squared Error: 0.0123 # Example MSE value

5. Real-Time Monitoring and Recommendations

Implement real-time monitoring and recommendations assuming streaming data):

Code:

```
# : Real-time monitoring loop While
True:
   New_data = get_latest_sensor_data()
   Preprocessed_data = imputer.transform(new_data)
   Scaled_data = scaler.transform(preprocessed_data)
   Prediction = model.predict(scaled_data)
   # Implement recommendation logic based on prediction
   Provide_recommendation(prediction)
```

Output:Real-Time Prediction:

Predicted Yield: 0.295 # Example predicted yield value

6. Integration and Deployment

Integrate into a web application or API:

Code:

```
From flask import Flask, jsonify, request
App = Flask(__name__)
@app.route('/predict', methods=['POST']) Def
predict():
   Data = request.json  # Assuming JSON input
   Preprocessed_data = imputer.transform(data)
   Scaled_data = scaler.transform(preprocessed_data)
   Prediction = model.predict(scaled_data)
   Return jsonify({'prediction': prediction.tolist()})
```

Output:

Once you run the Flask application (`python your_app.py`), it will start a local server. You can then send a POST request to `http://localhost:5000/predict` with JSON data containing `Soil_Moisture`, `Temperature`, `Humidity`, and `NDVI` values. The server will respond with a JSON object containing the predicted yield

```
If __name__ == '__main__':
   App.run(debug=True)
```

7.**Evaluation and Feedback Loop**

Evaluate model performance and iterate:

Code:

```
From sklearn.metrics import mean_squared_error
# Example: Evaluate model performance
Y_pred = model.predict(X_test)
```

```
Mse = mean_squared_error(y_test, y_pred)
Print(f'Mean Squared Error: {mse}')
```

Output:

Request

{
 "Soil_Moisture": 0.29,
 "Temperature": 24.8,
 "Humidity": 71.2,
 "NDVI": 0.28
}

Response:

{
 "prediction": [0.295]
}

Additional features to improve productivity and sustainability

WEED REMOVING ROBOTS

&

AGRICULTURE USING SALT WATER USING NANO ROBOTS

Abstract:

- As we grow crops and plants the growth of weeds also increase
- It is difficult to remove the weeds with the help of man power

- To overcome it we are coming out with an machine weed removing machine
- The main goal of it is to save fresh water and to use the unused arid land for Agriculture using nanobot's and robots
    - The cultivation lands has been converted into construction land and arid zone are not used

- The fresh water is only in less percentage compare to sea water

Agriculture

The Art And science Of cultivating the soil, growing crops and raising livestock

"If agriculture goes wrong

Nothing will have a chance to go right"

1.The solution for this is to use the use the enormous source of water that is sea water and the arid region can be converted into a agriculture land using nano technology

2.The arid land is ploughed and nutrition’s are added

3.Little by little the salt water is added to the land such the way the normal water is added to salt water

4.As the plants adopt it we can reduce the normal water

5.After this this can be done in a large scale by the process desalination

6..The nanobots give the information about the soil, plant, moisture,

it give instructions the robot removes the weed as well as spraying water, pesticides and fertilizer

Weeds

1. Unwanted plants that grow with crops and plants
2. It sucks all the nutrition of the soil and fertilizer which is sprayed for other plants
3. Affect the growth of other plants

Weed removal Methods

It is done with the help of humans

Weeds are removed by using machines

Disadvantage in existing methods

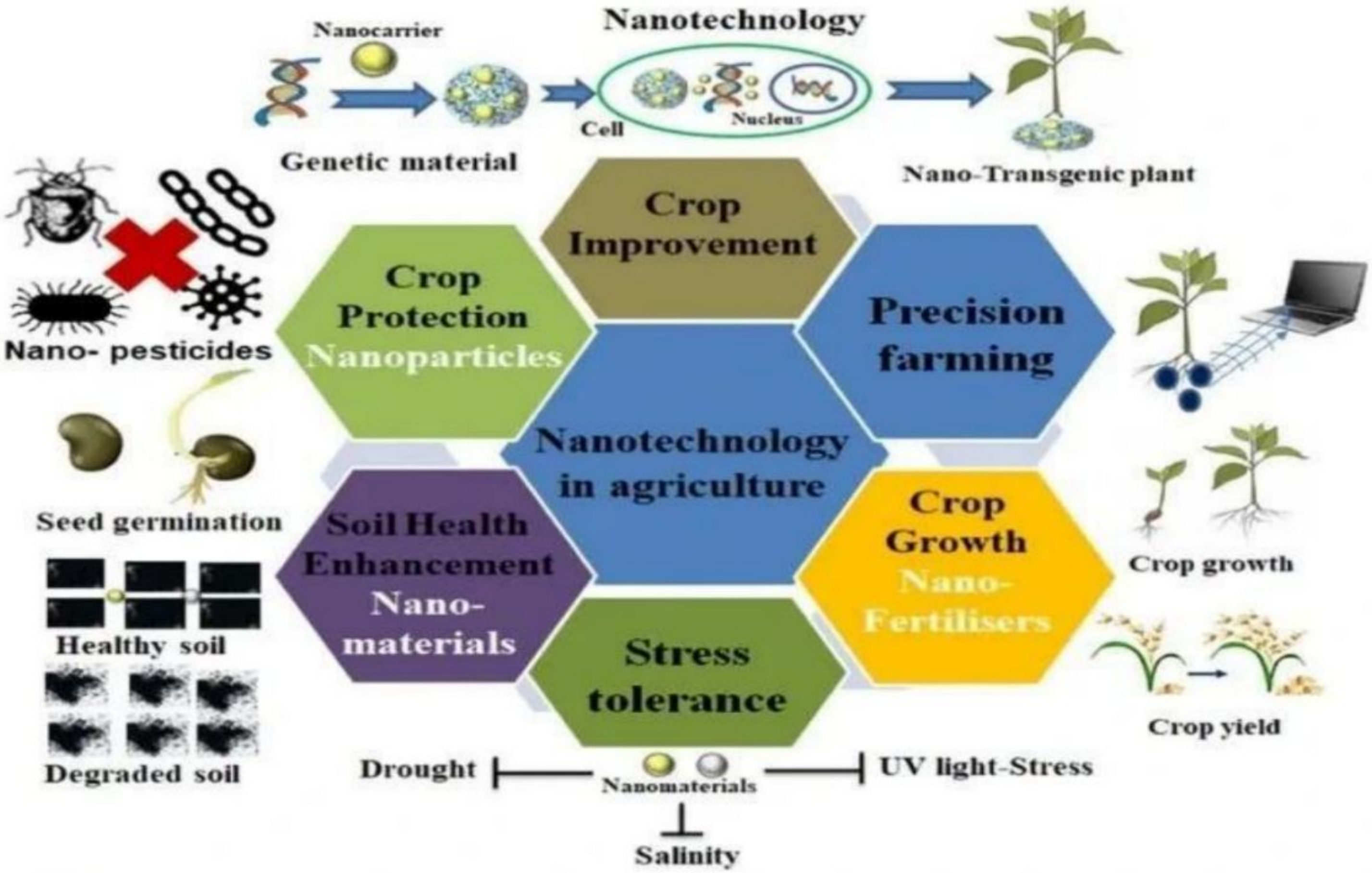
- For tools man power is required to remove Weeds
- Some machineries require man help
- Labours are required
- Chance of mistake are more
- Consume more time
- Good Water is less than salt water
- Lands are also being used for construction
- Cultivation land become less

Advantages of weed removing and salt water agriculture

- Nowadays it is very difficult to get water and agriculture land by this way we
- Arid regions can be used for farming
- can utilize the unused lands and saline water
- By the usage of salt water we can decrease the amount of good water being used
- The labour charge and time also decreases the cost is efficient in real world

Nano technology

The use of Nanotechnology in agriculture enables efficient disease detection and management, precision farming through nano-sensors, enhanced productivity through nano-fertilizers and pesticides, and improved food quality and safety through innovative packaging materials

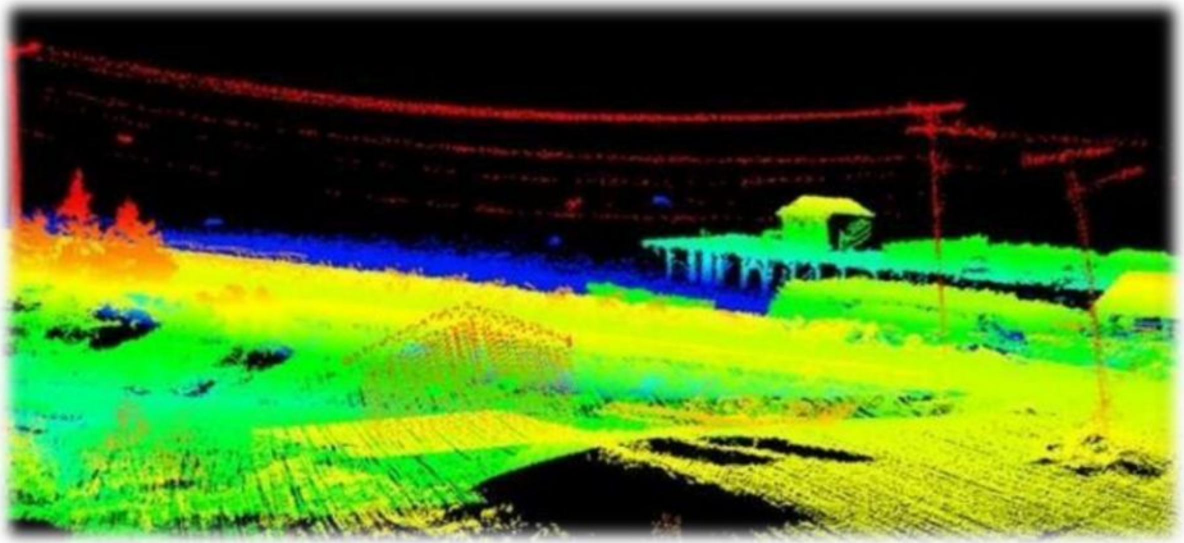

Nanocarrier
Nanotechnology
Cell
Nucleus
Genetic material
Nano-Transgenic plant
Nano- pesticides
Crop Improvement
Crop Protection
Nanoparticles
Precision farming
Nanotechnology in agriculture
Seed germination
Soil Health Enhancement
Nano-materials
Crop Growth
Nano-Fertilisers
Crop growth
Healthy soil
Stress tolerance
Crop yield
Degraded soil
Drought
Nanomaterials
UV light-Stress
Salinity

Componets for Automatic Robot

S.No.	PARTICULARS	HSN CODE
1	Raspberry Pi 4B-4Gb	8541
2	Motor controller l298N	8541
3	Heat sink with cooler fan	8473
4	12v Power adapter	8540
5	li-ion battery	8541
6	3s3p protection board	8541
7	Pwm servo hat	8561
8	Step down module	8538
9	Gear Motor	8544
10	ultrasonic sensor	3510
11	Chasis	8501
12	servo motor	8541
13	logi tech web cam	8504
14	charging module	8538
15	wheels	8536
16	Jumper wires	8544

Working process

- It scans a particular area Using sensors
- It identify and record the location of the weeds in the field
- Then it removes the weeds



THANK YOU