🌱 Project Title: Crop Stress Identification using CNN on Plant Village Dataset

Created By:

Name: MARREDDY GOWTHAM REDDY

Email: marreddygowthamreddy886@gmail.com

Phone No: 7416824867

📌 Objective:

The goal of this project is to **identify crop stress** using leaf images from the PlantVillage dataset by training a **Convolutional Neural Network (CNN)**. Early identification of crop stress can help in **precision irrigation** and **nutrient management** to increase agricultural yield and minimize losses.

📁 Dataset:

• **Source:** Local dataset located at `C:\Users\marreddyGowthamreddy\Downloads\PlantVillage`

• **Structure:** Images organized in folders by class (e.g., "Tomato___Leaf_mold", "Tomato___Healthy").

• **Classes:** Multiple crop conditions, both healthy and stressed.

• **Image Format:** RGB

• **Preprocessing:** Resized to 128x128 pixels and normalized using `rescale=1./255`.

1.Model Code and Working:

This project utilizes a Convolutional Neural Network (CNN) implemented using **TensorFlow/Keras**. The model reads images from the **PlantVillage dataset**, which is structured in folders by class (e.g., Tomato___Leaf_Mold, Tomato___Healthy, etc.).

Workflow:

- Images are resized to **128×128** and rescaled to the range [0, 1].
- Data is split into **training (80%)** and **validation (20%)** using ImageDataGenerator.
- A basic CNN architecture is trained over **10 epochs** to perform multi-class classification.
- Predictions are evaluated using standard metrics like **accuracy, precision, recall, MSE**, and **confusion matrix**.

2.Model Architecture:

Input: (128, 128, 3) RGB image

Conv2D (32 filters, 3x3 kernel, ReLU)

↓

MaxPooling2D (2x2)

↓

Conv2D (64 filters, 3x3 kernel, ReLU)

↓

MaxPooling2D (2x2)

↓

Flatten

↓

Dense (128 units, ReLU)

↓

Dropout (0.5)

↓

Dense (N classes, softmax)

Compilation:

- **Optimizer:** Adam
- **Loss Function:** categorical_crossentropy
- **Metric:** accuracy

3.Report on Why the Architecture Was Chosen:

• This architecture balances **accuracy and computational efficiency**.

• Conv2D + MaxPooling blocks allow hierarchical **feature extraction**.

• A **Dropout layer** helps reduce overfitting.

• The **softmax output layer** handles **multi-class classification**.

• It serves as a solid **baseline CNN model** before exploring deeper networks like ResNet, EfficientNet, etc.

4. Model Evaluation:

• **Validation Accuracy:** 88%

• **Precision:** 93%

• **Recall:** 94%

• **Strong classification ability** across multiple leaf conditions

• Confusion matrix showed strong diagonal dominance, indicating minimal misclassification

5. Optimization Report:

Techniques Applied:

- Image Rescaling via `ImageDataGenerator`
- Train/Validation Split (80:20)
- Dropout (0.5) to prevent overfitting
- Batch Size = 32
- Adam Optimizer for adaptive learning

Further Optimizations Suggested:

- Use **Data Augmentation**: Flip, Rotate, Shift
- Apply **EarlyStopping** and **ModelCheckpoint**
- Try **Transfer Learning** with pretrained models
- Tune learning rate & try **SGD with momentum**

🚀 Applications:

Precision Agriculture
Identify stressed crops early, enabling targeted intervention with irrigation, fertilizers, or pesticides.

Smart Irrigation Systems
Integrate with IoT sensors and automate watering schedules based on detected stress levels.

Agri-Drone Integration
Use aerial multispectral imaging and run this model on edge devices for large-scale monitoring.

Disease Management
Classify plant diseases before symptoms are visible to the naked eye, enabling preemptive treatment.

Farmer Advisory Platforms
Integrate with mobile apps that alert farmers when stress or disease is detected in their fields.

🔮 Future Scope

Multispectral and Hyperspectral Support
Extend the model to process additional spectral bands for more accurate stress detection (e.g., NDVI, NDRE, GNDVI).

Transfer Learning
Incorporate advanced pre-trained architectures (e.g., ResNet, EfficientNet) to improve accuracy and reduce training time.

□ **Explainable AI (XAI)**
Use Grad-CAM or SHAP to visualize which parts of the leaf image the model focuses on during prediction.

□ **Edge Deployment**
Optimize the model for deployment on devices like Raspberry Pi or NVIDIA Jetson for real-time field use.

□ **Real-Time Monitoring Dashboard**
Develop a web or mobile dashboard for real-time image uploads, visualization, and diagnosis reports.

🏁 Conclusion

This project successfully demonstrates how **Convolutional Neural Networks (CNNs)** can be applied to detect crop stress using plant leaf imagery. With high validation accuracy and precision, the model shows promise for real-world agricultural applications.

Its lightweight design ensures quick deployment and minimal training overhead, making it suitable for integration into **precision farming systems**, **mobile diagnostic tools**, and **drone-based monitoring platforms**.

With further enhancements like **transfer learning**, **spectral data fusion**, and **on-device inference**, this system can significantly contribute to **sustainable agriculture** and **global food security**.