Model Architecture

What We Built

To classify crop diseases from RGB images, we used a **Convolutional Neural Network (CNN)** based on **MobileNetV2**, a lightweight and efficient deep learning model. Instead of training everything from scratch, we used a technique called **transfer learning**, which allows us to reuse a model that's already learned how to recognize general features in images (like edges, textures, and shapes).

How the Model Is Structured

Here's a breakdown of the architecture we used:

- **Base Model – MobileNetV2**
  We started with MobileNetV2, a model that's been trained on millions of images from the ImageNet dataset. This model is great for mobile or embedded applications because it's fast and doesn't require much memory.
  We **froze** the base model so it keeps the useful features it already knows.

- **Global Average Pooling**
  Instead of flattening the data, which can add too many parameters, we used **GlobalAveragePooling2D**. This layer reduces the size of the data in a smart way by taking the average of each feature map — it's simple and helps avoid overfitting.

- **Fully Connected Layer**
  We added a **Dense layer with 256 neurons** and ReLU activation. This acts as a bridge between the pretrained features and our custom output layer, helping the model learn patterns specific to crop diseases.

- **Output Layer**
  The final layer has as many neurons as there are disease classes in our dataset. We used **Softmax activation**, so the model gives probabilities for each class.

- **Compilation**
  We used the **Adam optimizer**, which adapts the learning rate during training, and a **categorical cross-entropy** loss function, which is standard for multi-class classification problems.

Why We Chose This Setup

- **Transfer learning** helps us get high accuracy without needing a huge dataset.
- **MobileNetV2** is ideal because it's lightweight and still powerful — perfect for real-time use in the field or on mobile devices.
- By freezing the base model and only training the top layers, we make training faster and avoid overfitting.
- The model is simple, efficient, and accurate — a good balance for practical deployment in agriculture.