RTR_Lite_MobileNetV2 is a lightweight and efficient deep learning model for plant disease detection, designed for deployment on resource-constrained devices like Raspberry Pi. It modifies the standard MobileNetV2 architecture by incorporating advanced attention mechanisms to improve accuracy while significantly reducing model size and computational cost.

Main Algorithm and Architecture

The **RTR_Lite_MobileNet** model is an enhanced variant of MobileNetV2.

- **Base Architecture**: The model uses **MobileNetV2** as its foundation due to its efficiency in mobile and embedded applications, which is achieved through depthwise separable convolutions.
- **Key Modifications**: The architecture modifies the standard MobileNetV2 in two main ways:
    1. **Initial Layer Enhancement**: After the first convolution layer, a **Triplet Attention** module and a custom **RES Block** are added to improve the initial capture of cross-dimensional features.

2. **Layer Replacement**: The last three bottleneck layers of the original MobileNetV2 are removed and replaced with a sequence o`RES Block -> Triplet Attention -> RES Block`, which is repeated twice.

- **Resulting Efficiency**: These changes reduce the total number of trainable parameters by **53.8%** (from 2.27 million to 1.05 million) and the model size from 8.69 MB to **4.01 MB**.

 # Key Attention Mechanisms and Formulas

The model's improved performance is largely due to the integration of three distinct attention mechanisms.

- **Triplet Attention**: This mechanism captures interactions across the channel, height, and width dimensions of the input tensor by using three parallel branches. The final output is a simple average of the three branches.

$$y = \frac{1}{3}(y_1 + y_2 + y_3)$$

- **Modified SENet (Squeeze-and-Excitation)**: This module models the interdependencies between channels to assign significance-based weights. A modified version uses adaptive average pooling and max pooling in parallel to create a richer feature representation.
- **ECA (Efficient Channel Attention)**: A lightweight module that uses a simple 1D convolution to capture local cross-channel interactions with minimal computational overhead, making it ideal for efficient models.
- **RES Block**: This is a custom block which combines the **Modified SENet** and **ECA** modules sequentially to enhance feature representation while maintaining gradient flow.

$$RES\_Block(x) = ECA\_Attention(SENet(x))$$

Datasets and Preprocessing

The model was trained and tested on seven diverse, publicly available datasets.

- **Datasets Used**: **Plant Disease Dataset**, **Coffee**, **Wheat**, **Soybean**, **Sugarcane**, **PlantDoc**, and **PaddyDoctor**. These datasets vary widely in size, class balance, crop types, and geographical regions.
- **Preprocessing**: All images were uniformly resized to **224x224 pixels**. The data was split into an **80:10:10 ratio** for training, validation, and testing.
- **Data Augmentation**: To address class imbalances and increase diversity, several augmentation techniques were applied, including random rotations, color jittering, horizontal flipping, and random affine transformations (scaling, shearing).

Key Findings and Performance

The RTR_Lite_MobileNet model consistently outperformed the original MobileNetV2 and other state-of-the-art models across all seven datasets.

- **Superior Accuracy**: The model achieved top accuracies across all datasets, most notably:
  - **99.92%** on the Plant Disease dataset (with augmentation).
  - **100%** on the Wheat dataset (with augmentation).
  - **82.00%** on the challenging, real-world PlantDoc dataset.

    - **97.11%** on the PaddyDoctor dataset.
    - **96.78%** on the Soybean dataset.
- **Computational Efficiency**: With a model size of just **4.01 MB** and a low computational cost of **0.526 GFLOPs**, the architecture is highly efficient and suitable for low-power devices.
- **Optimizer Performance**: The **Adam optimizer** with an initial learning rate of 0.001 was found to yield the best results, achieving a testing accuracy of 99.89% on the Plant Disease dataset.

 # Limitations and Misclassifications

Despite its high performance, the model has several limitations and areas for future improvement.

- **Misclassification Errors**: The model showed minor confusion between visually similar diseases. For example, it had difficulty distinguishing between "bacterial_leaf_blight" and "bacterial_panicle_blight" in the PaddyDoctor dataset, and between "Rust" and "Red Spider Mite" in the Coffee dataset.
- **Computational Trade-off**: While efficient, the model's FLOP count is still higher than ultra-lightweight architectures like MobileNetV3 and ShuffleNetV2, indicating a trade-off between the added accuracy from attention mechanisms and raw computational speed.
- **Real-World Variability**: The datasets, while diverse, may not fully capture the variability of real-world agricultural settings, such as different lighting conditions, crop maturity stages, and seasonality. Further field trials are necessary to validate robustness.

#Deployment on IoT Devices

The model was successfully deployed and evaluated on real-world IoT devices.

- **Deployment Workflow**: The model, initially built in PyTorch, was converted to **ONNX** and then to **TensorFlow Lite** format for deployment on edge devices.
- **Tested Devices**: The model was deployed and tested on **Raspberry Pi 4** and **Raspberry Pi 5**.
- **Performance Gains**: The RTR_Lite_MobileNet model demonstrated lower latency and memory usage compared to the original MobileNetV2. For example, on Raspberry Pi 5, it showed a **4.41% improvement in CPU latency** (136 ms vs. 142 ms) and a **9.09% improvement in GPU latency** (22 ms vs. 24 ms).