NAME: JASWANT NAYAK

REG. NO.: 22BCE7180

ASSIGNMENT-2

Introduction

This document provides an overview of the architecture and functionalities of three models used for our project on image classification. Two models are well-established architectures, while the third is a novel model designed specifically for this project.

1. Existing Model 1: Convolutional Neural Network (CNN)

CNN is designed to automatically and adaptively learn spatial hierarchies of features from images. It involves three major types of layers:

- **Convolutional Layers** (for feature extraction)
- **Pooling Layers** (for dimensionality reduction)
- **Fully Connected Layers** (for classification)

Architecture:

- **Input Layer**: Takes an image of a weed (e.g., 128x128x3 for an RGB image).
- **Convolutional Layers**: Extracts features like edges, textures, and shapes.
- **Pooling Layers**: Reduces the spatial dimensions while retaining key features.
- **Fully Connected Layers**: Final layers responsible for classification.

Step-by-Step Breakdown:

1. **Input Layer**:
   a. Dimensions: 128x128x3
   b. Each input pixel has 3 channels (R, G, B).
2. **First Convolutional Layer (Conv1)**:
   a. Filters: 32 filters of size 3x3
   b. Output: The convolutional operation applies filters to extract low-level features such as edges.
   c. **Calculation**:

If we use **valid padding** (no padding), the output dimensions after applying a filter F of size (k x k) to an input image of size (H x W) are:
Output size = ((H-k)/s + 1) x ((W-k)/s + 1) x number of filters
Where s is the stride. For H=W=128, k=3, s=1, and 32 filters;
Output size = ((128-3)/1 + 1) x ((128-3)/1 + 1) x32 = 126x126x32

3. **Pooling Layer (MaxPooling)**:
    a. Pooling reduces the spatial dimensions (typically by a factor of 2).
    b. Output after pooling:

    126/2 x126/2x32 = 63x63x32
4. **Subsequent Convolutional Layers**:
    a. Conv2 applies 64 filters of size 3*3, and Conv3 applies 128 filters.

    The same convolution and pooling operations are performed, reducing the dimensions to:

    After Conv2 = 61x61x64, After Pooling = 30x30x64
    After Conv3 = 28x28x128, After Pooling = 14x14x128
5. **Flatten and Fully Connected Layer**:
    a. The output from the final convolutional layer is flattened to a vector of size:

    14*14*128 = 25,088
    b. A fully connected (FC) layer with 512 neurons transforms this flattened vector into class probabilities.
6. **Output Layer (Softmax)**:
    a. A softmax layer outputs a probability distribution over 8 classes (for 8 weed types).
    b. The predicted class is the one with the highest probability.

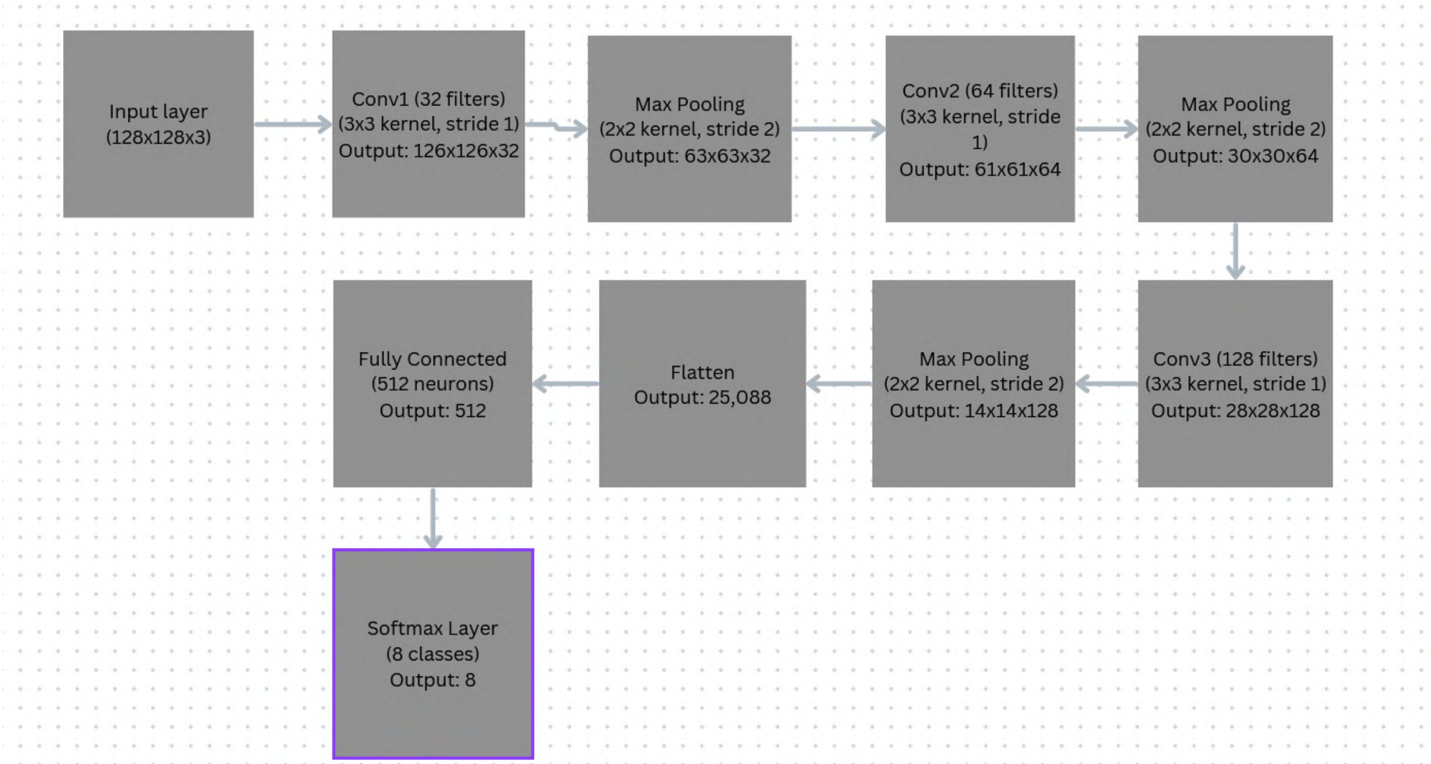
Computational Complexity:

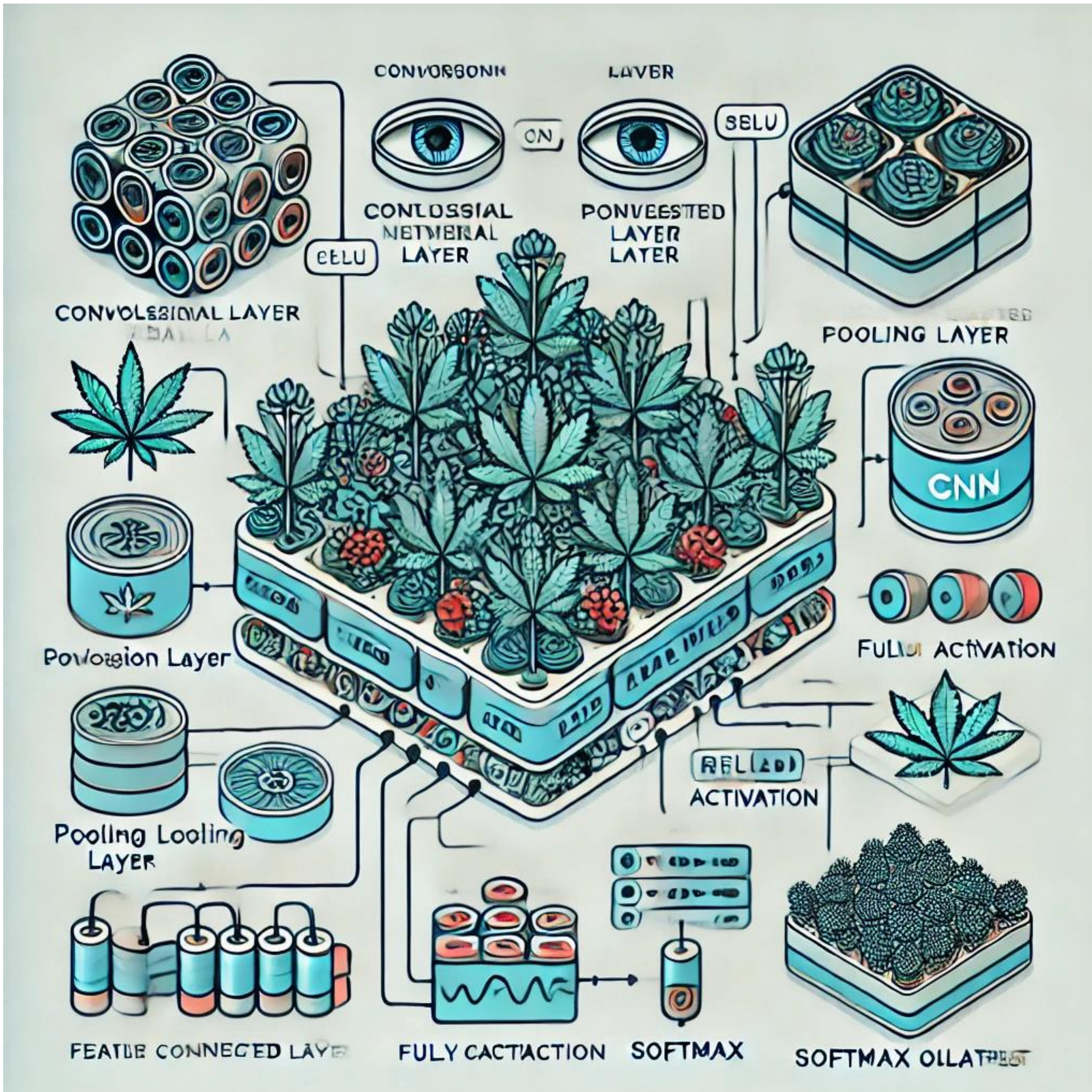
- **Number of parameters**:
    - Convolutional layers have parameters:

      K^2*number of filters*input channels
    - For Conv1, the number of parameters is 3*3*3*32 = 864

- As the layers grow deeper, the number of filters increases, leading to more parameters.

Architecture Diagram





2. Existing Model 2: ResNet (Residual Networks)

ResNet introduces **residual connections** (skip connections) that allow the network to train deeper layers effectively by bypassing some of the layers during the forward pass. The key idea is to avoid the **vanishing gradient problem** in deep networks.

Architecture:

- **Input Layer**: Takes a weed image.
- **Convolutional and Residual Blocks**: A series of convolutional layers combined with identity mappings (skip connections).
- **Fully Connected Layers**: Final layers for classification.

Step-by-Step Breakdown:

1. **Initial Convolution Layer**:
   a. A single convolution operation like CNN is performed.

   For example, applying 64 filters of size 7x7 reduces the dimensions to 62x62x64
2. **Residual Blocks**:
   a. **Residual Block**: Contains two convolutional layers, where the input to the block is added back to the output of the second convolution layer via a skip connection.
   b. For example, in the first block:

   $$\text{Output} = F(x) + x$$

   Where F(x) is the transformation from two convolutional layers, and x is the input to the block.
3. **Global Average Pooling**:
   a. Instead of flattening the feature maps, ResNet uses **Global Average Pooling**, which reduces the spatial dimensions to a 1x1 feature map, maintaining the depth.
   b. For example, if the input is 7*7*512, global average pooling reduces it to 1*1*512.
4. **Fully Connected Layer**:
   a. The final fully connected layer maps the pooled features to the number of classes (8 classes for weed types).
5. **Output Layer**:
   a. A softmax layer outputs the class probabilities.

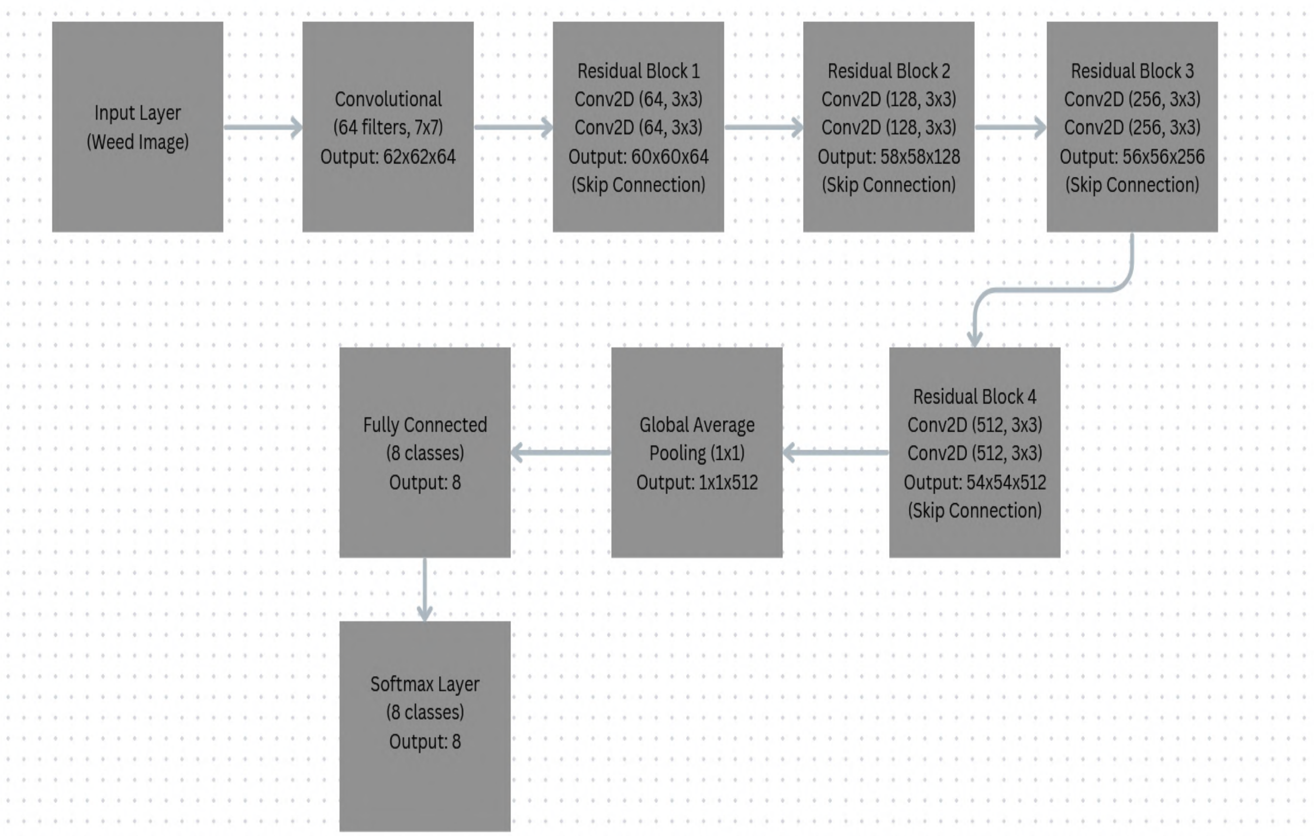
Computational Complexity:

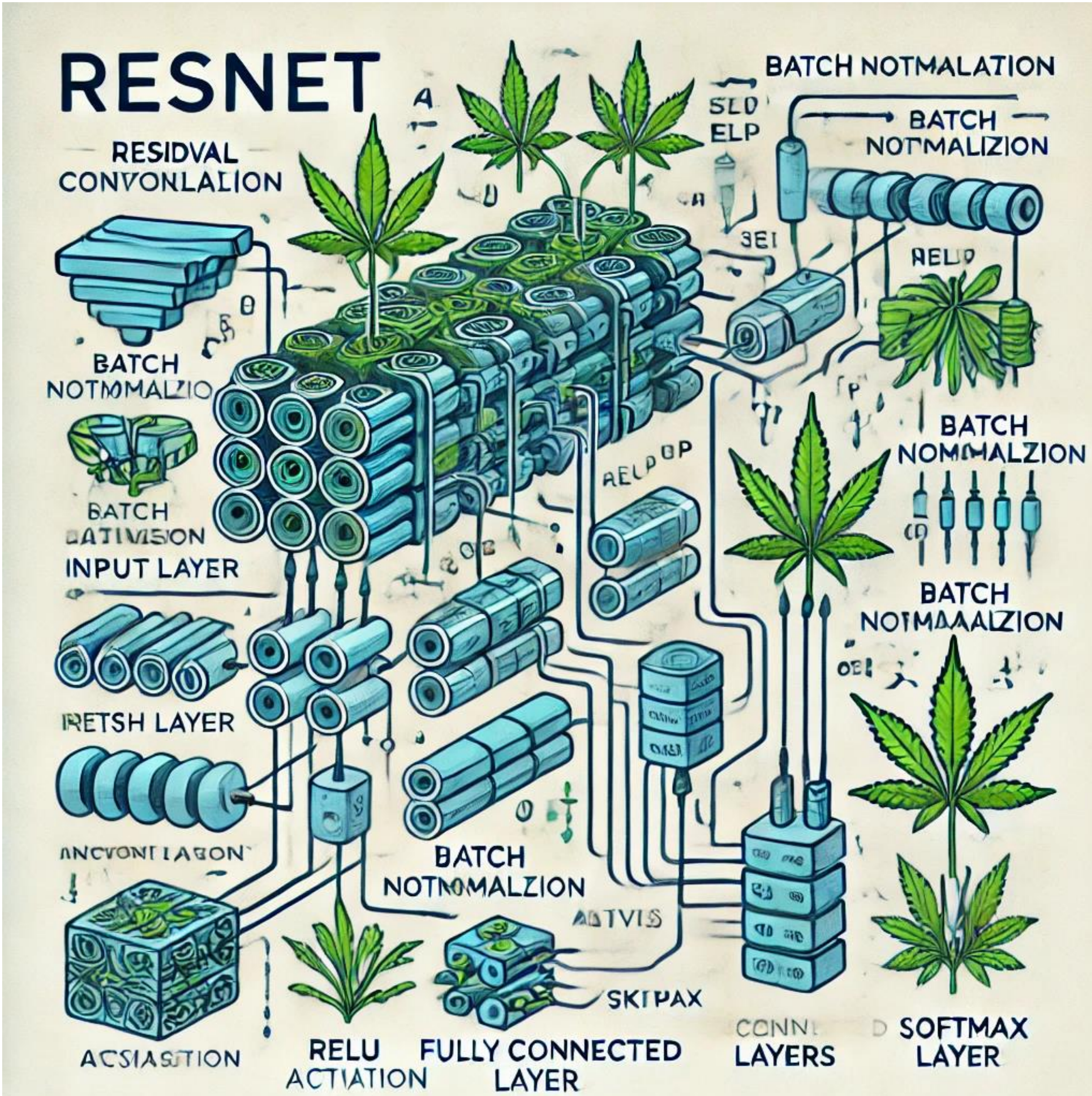
- **Residual Block**:

- The key computation in ResNet comes from the residual blocks. The skip connection allows the network to propagate gradients easily across layers, which helps in training deeper models (like ResNet34, ResNet50).
- Each residual block has a complexity like a typical CNN but with an added **element-wise addition** operation.

- **Number of Parameters**:
  - The number of parameters in a residual block depends on the number of filters and the input size.
  - For ResNet18, the number of parameters is around **11.7 million**, while ResNet50 has about **25 million**.

Architecture Diagram





3. Proposed Model: Custom Hybrid Model

Description

Purpose: This novel model integrates aspects of CNN and ResNet to combine the strength of feature extraction with deep residual learning. This hybrid model introduces a novel layer structure to capture more nuanced image details while maintaining computational efficiency.

Key Features:

- **Hybrid Residual-Convolutional Layers:** Combining convolutional and residual blocks to optimize feature extraction.
- **Adaptive Pooling Layers:** Dynamically adjust feature map size based on image input.
- **Custom Attention Mechanism:** Focuses on important parts of the image, improving classification accuracy for challenging datasets.

Components and Workflow:

- **Convolutional Blocks:** Initial feature extraction from input images.
- **Residual Blocks with Attention:** Enhance deeper feature learning while focusing on key parts of the image.
- **Adaptive Pooling and Dense Layers:** Final classification stage.

The **Custom Hybrid Model** for weed detection combines aspects of **CNN**, **ResNet**, and introduces a **custom attention mechanism** with **adaptive pooling**. The goal is to merge the advantages of both architectures and improve accuracy by focusing on the most important features in the image using attention.

Let’s break down and deeply analyze the **layer-by-layer operations** and **calculations** for the custom hybrid model, highlighting how each part contributes to the model's ability to detect weeds accurately.

1. Input Layer

- **Input Image**: 128x128x3 (RGB image).
- Each pixel has three color channels (R, G, B).

2. CNN Component: Convolutional Layers

This part of the architecture is like a standard CNN, where convolutional layers are used to extract features.

First Convolutional Layer (Conv1)

- **Input**: 128x128x3
- **Filters**: 32 filters of size 3x3
- **Stride**: 1 (moves by one pixel).
- **Padding**: 1 (so that the output maintains the same spatial dimensions).

Calculation:

Output size of the convolutional layer can be calculated by:

Output size = (Input size-Filter size + 2*Padding)/Stride + 1

For Conv1:

Output size = (128-3 +2*1)/1 + 1 = 128x128x32

- The result is 128x128x32, meaning there are 32 feature maps, each of size 128x128.

Pooling Layer

- **MaxPooling** layer with a 2*2 window is applied to reduce spatial dimensions.
- **Output** after pooling:

  128/2x128/2x32 = 64x64x64
- This reduces the size to 64x64x32.

Subsequent Convolutional Layers (Conv2 and Conv3)

- **Conv2**: 64 filters of size 3x3.
- After applying the same convolution and pooling operations:

  After Conv2: 64x64x64 and after pooling: 32x32x64
- **Conv3**: 128 filters of size 3x3.

  After Conv3: 32x32x128   and after pooling: 16x16x128

3. ResNet Component: Residual Blocks

The custom model incorporates **ResNet-style residual blocks** after the convolutional layers to capture deeper features. The residual connections allow the model to bypass certain layers, ensuring the gradients flow efficiently through deeper layers.

Residual Block

- A **residual block** consists of two convolutional layers. The input to the block is added directly to the output of the second convolutional layer via a **skip connection**.

Let's assume the first residual block operates on the output from Conv3, which is 16*16*128.

- **Conv layer 1**: 3x3 filters, 128 filters.
  Output size=16×16×128
- **Conv layer 2**: 3x3, 128 filters.
  Output size = 16 x 16 x 128

Skip Connection (Identity Mapping)

- **Skip connection**: The input 16x16x128 is added elementwise to the output of the residual block, resulting in:
  Final block output = 16x16x128

The identity mapping is added elementwise as:

$$F(x) + x$$

Where F(x) is the result of the convolutional layers, and x is the original input.

4. Custom Attention Mechanism

This layer allows the model to focus on important parts of the image, improving accuracy by weighing certain features more heavily than others.

Attention Weights

- The feature maps after the residual block (16x16x128) are **flattened** into a vector:
  16x16x128 = 32,768
- A **fully connected layer** is applied to learn the attention weights:
  FC Layer 1:32,768 -> 512
- The attention weights are then projected back to the original feature map size via a second fully connected layer:
  FC Layer 2:512->32,768
- The **attention weights** are reshaped back to the original feature map size (16x16x128) and multiplied elementwise with the feature maps.

Applying Attention

- Attention-adjusted feature map:
  Feature Map x Attention Weighs
  This produces a refined feature map that prioritizes certain areas of the image based on learned attention weights.

5. Adaptive Pooling

To ensure that the network is robust to different image sizes, **adaptive pooling** is applied. This layer dynamically adjusts the size of the feature maps to a fixed size, regardless of the input image size.

Adaptive Average Pooling

- The output from the attention mechanism (16x16x128) is passed through adaptive average pooling to reduce the size to a fixed dimension:
  Output: 1x1x128
- This reduces the dimensionality while preserving the depth of the feature maps.

6. Fully Connected Layers

After adaptive pooling, the feature maps are passed through fully connected (FC) layers for classification.

Fully Connected Layers

- **Flatten** the pooled output: 1x1x128 = 128.
- First fully connected layer:

  FC Layer 1: 128→512
- Second fully connected layer:

  FC Layer 2: 512→128
- Output layer (softmax):

  FC Layer 3: 128→8
- The softmax layer outputs a probability distribution over 8 classes (for 8 weed types).

7. Output Layer (Softmax)

- The final output is a **softmax** layer that produces a probability distribution over the 8 classes (the 8 weed types).
- The class with the highest probability is the predicted class.

Custom Hybrid Model: Overall Calculation and Complexity

1. **Convolutional Layers**: Perform standard convolution operations to extract basic features like edges, textures, and patterns.
2. **Residual Blocks**: Help the model learn deeper features without losing the gradients, crucial for deeper layers.
3. **Attention Mechanism**: Improves the model's focus on the most important parts of the image by applying attention weights.
4. **Adaptive Pooling**: Ensures flexibility in the model, making it robust to various input image sizes.
5. **Fully Connected Layers**: Finally, the fully connected layers map the learned features to the output classes, allowing the model to classify the weeds accurately.

Computational Complexity and Parameters

- **Convolutional Layers**: The number of parameters in each convolutional layer is:

  Params = (k*k*input channels) * number of filters

For Conv1 (3x3,32 filters):

$$3x3x3x32 = 864 \text{ parameters}$$

For deeper layers, this number increases significantly as the number of filters increases.

- **Residual Blocks**: Each residual block has two convolutional layers, with an identity mapping. The computational complexity increases with the number of filters and input size.
- **Attention Mechanism**: The attention mechanism adds extra computation in the form of two fully connected layers. However, it improves the model's ability to focus on relevant features, resulting in better accuracy.

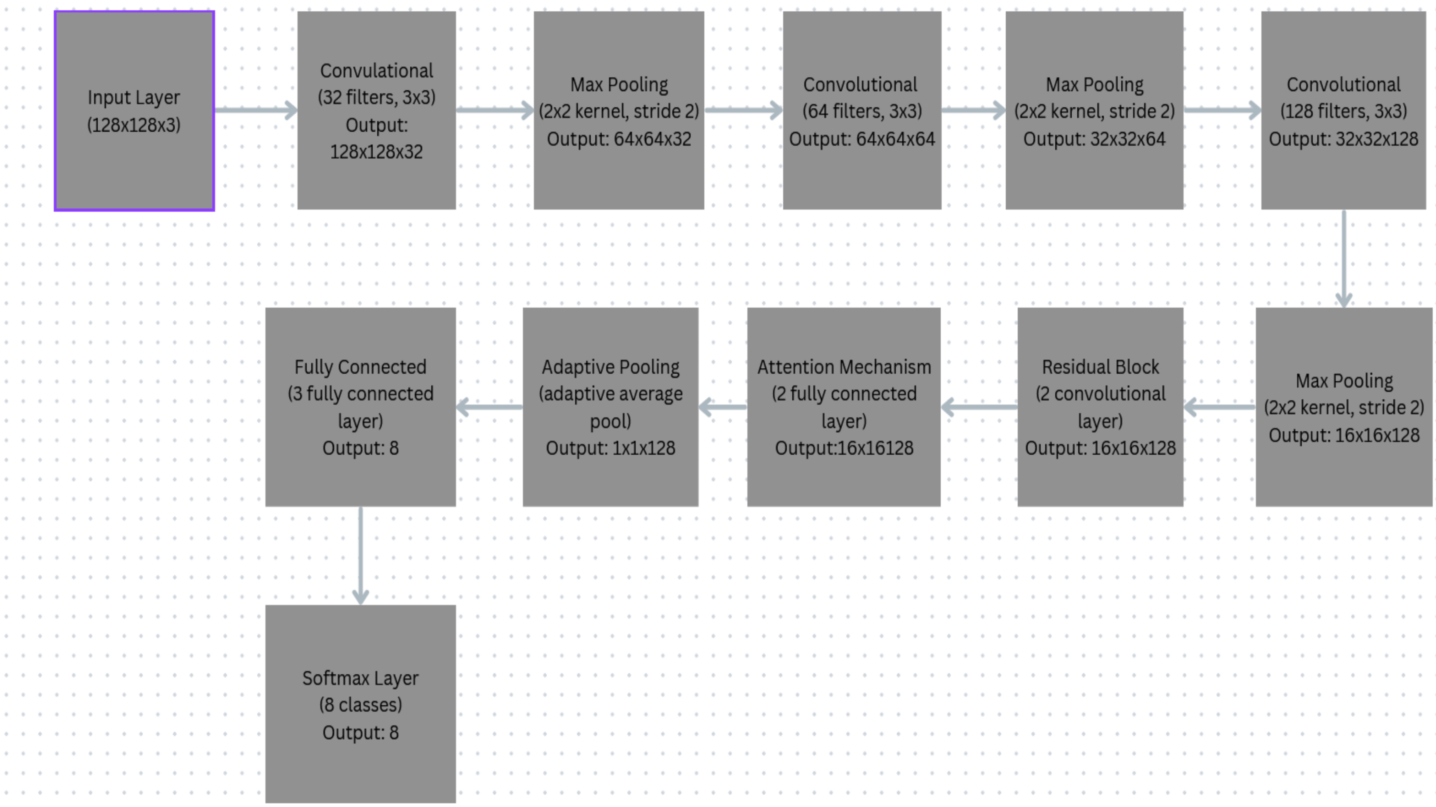
Conclusion: Advantages of the Custom Hybrid Model

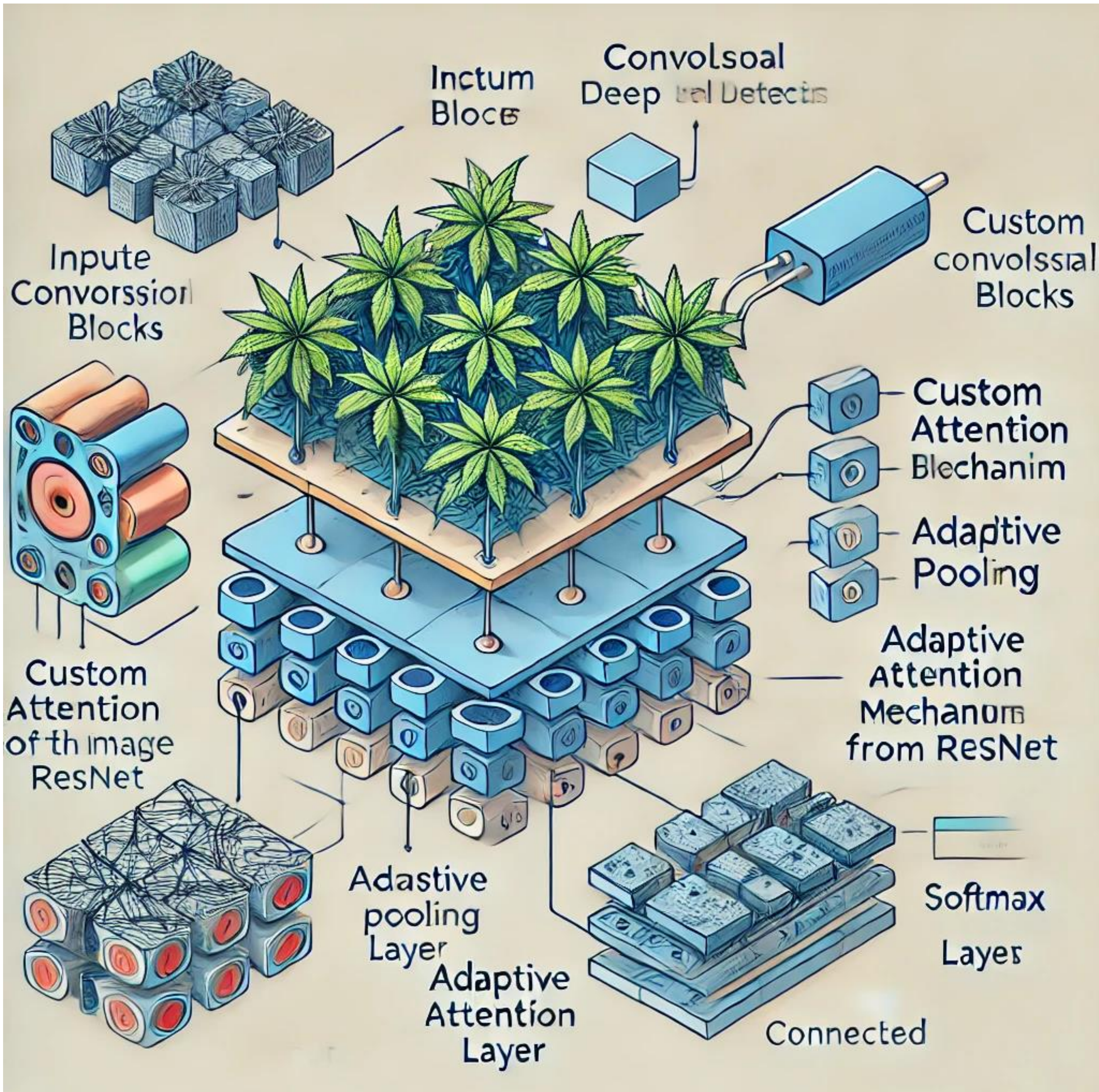
- **Better Feature Extraction**: Combining convolutional layers and residual blocks allows the model to capture both low-level and high-level features.
- **Efficient Gradient Flow**: Residual connections ensure that the gradients flow easily through deeper layers, making training more efficient.
- **Focused Learning**: The attention mechanism allows the model to focus on key parts of the image, improving classification accuracy.
- **Robustness**: Adaptive pooling makes the model robust to various input image sizes, which is essential in real-world applications where image sizes may vary.

This hybrid model, though computationally more complex than a standard CNN or ResNet, offers a significant improvement in accuracy and performance for weed detection.

Architecture of Custom Hybrid Model

Architecture Diagram





Conclusion for Custom Hybrid Model

The custom hybrid model introduced in this project integrates the strengths of both CNN and ResNet architectures while incorporating a novel attention mechanism to focus on critical image regions. By combining convolutional layers for feature extraction, residual blocks for deep learning optimization, and attention mechanisms to prioritize relevant data, this hybrid model offers several advantages:

1. **Improved Feature Extraction:** The combination of convolutional and residual blocks enables the model to extract deep hierarchical features from weed images, improving accuracy in differentiating weeds from crops.
2. **Robustness to Complex Data:** The residual connections help mitigate the vanishing gradient problem, allowing for deeper networks that perform better in complex scenarios, such as varying lighting conditions and diverse weed species.
3. **Enhanced Focus with Attention Mechanism:** The custom attention layer highlights key features of the image, increasing precision and minimizing misclassification, especially in cases where weeds and crops are visually similar.
4. **Adaptive Learning:** The use of adaptive pooling ensures the model can handle different image sizes and environmental conditions, making it robust for real-world applications in agriculture.

Overall, this hybrid architecture offers a significant improvement in weed detection performance, demonstrating versatility and scalability for various agricultural environments. Future steps include testing the model in real-world field conditions and refining it for deployment in autonomous weed detection systems.