### **Model Architecture Justification**

During the experimentation phase of this project, I tested multiple convolutional neural network (CNN) architectures to identify a structure that balances learning capacity and generalization performance. Below is a summary of the models explored, their results, and the rationale for choosing the final architecture.

#### **1. Initial Model (1 Conv + 1 Dense Layer)**

* **Structure**: A minimal architecture with a single convolutional layer followed by one dense layer.
* **Observation**: Due to the small number of trainable parameters and insufficient feature extraction, the model underperformed with only **78% validation accuracy**.
* **Conclusion**: This model lacked depth and capacity to learn complex patterns from color images.

#### **2. Deeper Dense Network (3 Dense Layers)**

* **Structure**: Increased the depth of the dense layers while keeping convolutional layers minimal.
* **Observation**: Validation accuracy increased to **88.5%**, but signs of **overfitting** began to emerge.
* **Conclusion**: The model started to memorize the training data, indicating that the architecture might be too dense or lacked regularization.

#### **3. Grayscale Image Model**

* **Structure**: Converted all color images to grayscale and trained with a deeper network (trained for 5 epochs).
* **Observation**: Achieved **97% training accuracy** but only **85% validation accuracy** — a clear case of overfitting.
* **Conclusion**: Reducing image color channels to 1 (grayscale) may have removed critical features, leading to poor generalization.

#### 

#### 

#### **4. Grayscale Model with Fewer Epochs**

* **Structure**: Same as above but trained for only 4 epochs to mitigate overfitting.
* **Observation**: Validation accuracy further dropped to **84%**.
* **Conclusion**: Reducing epochs was not enough to address the core issue — loss of important color information in grayscale images.

#### **5. Final Model (Selected Architecture)**

* **Structure**:  
   The final model consists of:  
  + **2 Convolutional Layers**:  
    - Conv2D(32, (3, 3), activation='relu')
    - Conv2D(64, (3, 3), activation='relu')
  + **2 Max Pooling Layers**: Applied after each convolutional layer to reduce spatial dimensions.
  + **1 Flatten Layer**: Converts 2D features to a 1D feature vector.
  + **2 Dense Layers**:  
    - Dense(256, activation='relu') — hidden layer
    - Dense(num\_classes, activation='softmax') — output layer for classification
* **Additional Strategy**: Implemented **EarlyStopping** to prevent overfitting.
* **Observation**: Achieved **87% validation accuracy** with strong generalization performance.
* **Conclusion**: This architecture offered a well-balanced trade-off between learning capacity and regularization. Working with **color images**, rather than grayscale, helped preserve vital features for accurate classification.