### **Performance Optimization Techniques**

To improve the model's accuracy and generalization, several optimization techniques were applied throughout the training process:

#### **1. Early Stopping**

* **What it does**: Monitors validation loss and stops training when it no longer improves, thus preventing overfitting.
* **Effect**: Helped avoid excessive training that could lead to overfitting. Validation accuracy stabilized around **87%**.

#### **2. Data Augmentation (via ImageDataGenerator)**

* **What it does**: Randomly alters images (e.g., flipping, zooming, shifting) during training to increase data diversity and robustness.
* **Effect**: Reduced overfitting and improved model's ability to generalize on unseen data.

#### **3. Image Rescaling**

* **What it does**: Normalizes pixel values to the range [0, 1] for faster and more stable training.
* **Implementation**:  
   rescale=1./255 in the data generators.
* **Effect**: Enabled efficient gradient updates and prevented exploding/vanishing gradients.

#### **4. Balanced Model Depth**

* **Design Decision**: Carefully chose a moderate number of layers — **2 convolutional + 2 dense layers** — to maintain a balance between model capacity and overfitting risk.
* **Effect**: Achieved solid validation accuracy without adding unnecessary complexity.

### 

### **Summary**

These combined optimization strategies significantly improved model performance and robustness. The use of **early stopping** and **data augmentation**, in particular, played key roles in achieving a strong **87% validation accuracy** while avoiding overfitting.