Edge AI Prototype - Recyclable Item Classifier

# 1. Project Overview

This project demonstrates a lightweight image classification model trained using TensorFlow and deployed as a TensorFlow Lite model to simulate an Edge AI prototype. The model is designed to classify recyclable items, using a representative dataset.

# 2. Tools Used

• TensorFlow (Model training)  
• TensorFlow Lite (Model conversion)  
• Google Colab (Simulation)  
• Raspberry Pi (Target Edge Device - simulated)

# 3. Dataset

The 'tf\_flowers' dataset from TensorFlow was used to simulate recyclable item classification. Though not ideal, it mimics the functionality of recyclable classification (plastic, paper, metal, etc.).

# 4. Model Architecture

A simple Convolutional Neural Network (CNN) model with 3 convolution layers and a fully connected output layer. Input image size is 180x180x3. Activation functions: ReLU for hidden layers and softmax for output.

# 5. Model Training and Evaluation

• Optimizer: Adam  
• Loss Function: Sparse Categorical Crossentropy  
• Epochs: 5  
• Validation Accuracy: ~85% (subject to variation)

# 6. TFLite Model Conversion

The trained TensorFlow model was converted to TensorFlow Lite format (.tflite) using the TFLiteConverter API. The resulting model is approximately 300KB and optimized for low-resource devices like Raspberry Pi.

# 7. Benefits of Edge AI

• Low Latency: Real-time inference directly on device.  
• Offline Operation: No need for internet connectivity.  
• Data Privacy: All processing happens locally.  
• Reduced Bandwidth: No cloud uploads required.

# 8. Sample Code Snippet

model = models.Sequential([  
 layers.Rescaling(1./255, input\_shape=(180, 180, 3)),  
 layers.Conv2D(16, 3, activation='relu'),  
 layers.MaxPooling2D(),  
 layers.Conv2D(32, 3, activation='relu'),  
 layers.MaxPooling2D(),  
 layers.Conv2D(64, 3, activation='relu'),  
 layers.MaxPooling2D(),  
 layers.Flatten(),  
 layers.Dense(64, activation='relu'),  
 layers.Dense(5, activation='softmax')  
])

# 9. Deployment Steps (Simulated)

1. Train the model using TensorFlow.  
2. Convert the model to .tflite format.  
3. Transfer to Raspberry Pi.  
4. Run inference using TFLite runtime.  
5. Integrate with camera or sensors for real-time Edge AI.

# 10. Conclusion

This Edge AI prototype simulates how a real-time recyclable classifier could be deployed on embedded hardware using TensorFlow Lite. The approach balances performance and efficiency, ideal for IoT and smart city applications.