Edge AI Prototype Report: Waste Classification Using CNN & TensorFlow Lite

Project Title: Real-Time Waste Classification with Edge AI

Theme: *Pioneering Tomorrow's AI Innovations*

Author: Alois Maingi Mbithi

Course: Power Learn Project (PLP) – Edge AI Assignment

1. Introduction

This project implements an Edge AI solution to classify recyclable waste using a lightweight Convolutional Neural Network (CNN) trained on an image dataset. The model is optimized and deployed using TensorFlow Lite for real-time applications on edge devices such as Raspberry Pi and smartphones.

2. Problem Statement

Waste mismanagement is a growing environmental concern. Traditional waste sorting is manual and inefficient. This project proposes a smart waste classification system using computer vision deployed at the edge to automate sorting at the point of disposal.

3. Objectives

- Develop a CNN model to classify waste into two categories.
- Train and evaluate the model using TensorFlow.
- Convert the model into a lightweight TensorFlow Lite format.
- Demonstrate the benefits of Edge AI: low latency, enhanced privacy, and offline capability.

4. Tools and Frameworks

- **TensorFlow/Keras**: Model building and training

- **Google Colab**: Model development and testing environment
- **TensorFlow Lite**: Model optimization for edge deployment
- **Matplotlib**: Visualization of training performance
- **Dataset**: Custom image dataset with two classes – recyclable and non-recyclable waste

5. Dataset Summary

- **Total Images**: 22,564 (Train), 2,513 (Validation)
- **Image Size**: 128x128 RGB
- **Classes**: 2 (Recyclable, Non-Recyclable)
- **Source**: Provided zip dataset via Google Drive

6. Model Architecture

A simple CNN model was used:

- 3 Convolutional Layers
- MaxPooling after each convolution
- Flatten layer followed by Dense layers
- Softmax output layer

7. Training Configuration

- **Epochs**: 5
- **Optimizer**: Adam
- **Loss Function**: Categorical Crossentropy
- **Validation Split**: Predefined in dataset

8. Accuracy Metrics

Training & Validation Accuracy and Loss

Epoch	Training Accuracy	Validation Accuracy	Validation Loss
1	76.78%	86.15%	0.3580
2	84.72%	87.39%	0.3286
3	86.00%	90.81%	0.2522
4	87.91%	87.66%	0.3207
5	89.42%	91.44%	0.2521

The model showed excellent performance with over 91% validation accuracy after 5 epochs, demonstrating its capability to generalize on unseen data.

9. Model Deployment Steps

Step 1: Save the Keras Model

```
model.save("recyclenet_model.keras")
```

Step 2: Convert to TensorFlow Lite

```
converter = tf.lite.TFLiteConverter.from_keras_model(model)
tflite_model = converter.convert()

with open("recyclenet_model.tflite", "wb") as f:
    f.write(tflite_model)
```

Step 3: Deploy on Edge Device

- Transfer `recyclenet_model.tflite` to Raspberry Pi or Android device.
- Use `TensorFlow Lite Interpreter` in Python or Java to load and run inference.
- Provide image input and receive class prediction output in real-time.

10. Visualization

Training and validation performance were plotted using Matplotlib to monitor learning behavior:

- **Accuracy Plot**: Showed steady improvement across all epochs
- **Loss Plot**: Demonstrated decreasing training and validation loss

11. Benefits of Edge AI in This Project

- **Low Latency**: Inference runs instantly on-device without delay.
- **Privacy Preserving**: No need to send images to cloud servers.
- **Offline Operation**: Works even without internet access.
- **Energy Efficient**: Suitable for low-power devices like Raspberry Pi.
- **Real-Time Decision Making**: Ideal for smart bins or environmental monitoring systems.

12. Challenges Encountered

- Large model file size prevented direct GitHub upload. Solved by uploading model to Google Drive.
- Dataset upload in Colab took long; resolved by compressing and using `gdown` for faster retrieval.
- Errors during mounting and conversion were handled by isolating steps and debugging in stages.

13. Conclusion

This project demonstrates how Edge AI can bring machine learning to real-world sustainability problems. With a high-performing model and a lightweight `.tflite` deployment, this waste classification system can empower smart cities and environmental initiatives to automate waste sorting efficiently and privately.