Food Freshness AI – End■to■End Build Log & Guide

Generated on: 2025-11-07 10:21

Overview

This document summarizes everything we set up to build a simple Food Freshness classifier end■to■end: dataset creation, training with TensorFlow/Keras, converting to TensorFlow Lite, and integrating the model into an Android app using Java + CameraX. It also lists the issues encountered and the fixes applied.

1) Dataset Preparation

1. Created a folder structure: **dataset/fresh/**, **dataset/rotten/**.
2. Initially only 7 images per class; target is 200–300 per class.
3. Captured images with varied backgrounds, lighting, and distances.
4. Planned data augmentation to compensate for small dataset.

2) Model Training (TensorFlow/Keras)

1. Base model: MobileNetV2 (ImageNet weights), input size 224×224.
2. Head training with frozen base, then fine■tuning of the top ~40 layers.
3. Strong on-the-fly augmentations: flip, rotation, zoom, translation, contrast.
4. Compiled with Adam; callbacks: EarlyStopping + ReduceLROnPlateau.
5. Artifacts produced: **model_freshness.h5**, **class_indices.json**.

```
        Export to TensorFlow Lite (dynamic range int8):
converter = tf.lite.TFLiteConverter.from_keras_model(model)
converter.optimizations = [tf.lite.Optimize.DEFAULT]
tflite_model = converter.convert()
open('model_freshness_int8.tflite','wb').write(tflite_model)
```

3) Android App (Java + CameraX + TFLite)

1. Created a new Android Studio project (Java, Empty Activity, minSdk 24).
2. Added dependencies: CameraX, TensorFlow Lite, Lite Support.
3. Added permissions in AndroidManifest: CAMERA.
4. UI: PreviewView, Button (Capture & Predict), TextView for result.
5. Assets: placed **model.tflite** and **class_indices.json** in *app/src/main/assets/*.
6. Inference pipeline: Bitmap → resize 224×224 → TensorImage → interpreter.run().

Camera Capture Path (Stable)

- Used ImageCapture to save JPEG to app-specific storage.
- Decoded JPEG to Bitmap (BitmapFactory).

- Optional rotation adjustment for certain devices.

4) Issues Encountered & Fixes

Issue	Cause	Fix
The PyDataset has length 0	Empty/incorrect dataset folders	Placed images into dataset/fresh and dataset/rotten; re-ra
JSONObject.keySet() compile error	Android's JSONObject lacks keySet()	Replaced with obj.names() iteration.
PlaneProxy.buffer not found	Wrong API property	Used planes[i].getBuffer() instead of .buffer.
ArrayIndexOutOfBounds in YUV→NV21	Ignored row/pixel strides in YUV_420_888	Switched to save-to-file JPEG → Bitmap path.
TFLite input mismatch (602112 vs 150528 bytes)	Model expected FLOAT32; app sent UINT8	Detected input type at runtime; normalized if FLOAT32.
Confidence 255.00	INT8 outputs scaled 0–255	Dequantized using scale/zeroPoint; applied softmax; disp

5) Current Status

- App builds and runs; predictions shown on device.
- Example: image predicted 'rotten' with confidence ~0.64.
- Small dataset (7 per class) limits confidence and generalization.

6) Recommended Next Steps

1. Collect 200–300 images per class with varied lighting/backgrounds.
2. Keep subject large in frame; avoid near-duplicates.
3. Consider more classes (fresh, ripe, overripe, moldy).
4. Tune training: 25–30 epochs, batch 16, stronger augmentation.
5. Optional: switch to EfficientNet-Lite or MobileNetV3-Small.
6. In app: threshold low-confidence (<0.80) as 'Unsure—retake photo'.

Appendix A – Key Gradle Dependencies

```
implementation "org.tensorflow:tensorflow-lite:2.14.0"
implementation "org.tensorflow:tensorflow-lite-support:0.4.4"
implementation "org.tensorflow:tensorflow-lite-task-vision:0.4.0" (optional)
implementation "androidx.camera:camera-camera2:1.2.3"
implementation "androidx.camera:camera-lifecycle:1.2.3"
implementation "androidx.camera:camera-view:1.2.3"
```

Appendix B – Dataset Structure

```
dataset/
  fresh/
    img_001.jpg, ...
  rotten/
    img_001.jpg, ...
```

Appendix C – Inference Notes

We detect the model input tensor shape and type at runtime. For FLOAT32 input we normalize images to [0,1]. For INT8/UINT8 outputs we dequantize with (value - zeroPoint) * scale and apply softmax before selecting the top class.