ML and AI at the edge
with Google Coral, Edge TPU and Cloud Platform

Markku Lepisto
IoT/ML Solutions Architect, JAPAC

Google Cloud
1. Be **socially beneficial**.

2. **Avoid** creating or reinforcing **unfair bias**.

3. Be built and tested for **safety**.

4. Be **accountable** to people.

5. Incorporate **privacy** design principles.

6. Uphold high standards of **scientific** excellence.

7. Be made **available** for uses that accord with these principles.
Introducing Coral
Edge AI

Privacy
Consumers & businesses
(privacy & compliance)

Latency
Self-driving car
@65 miles/hr
100 ms = 9.5 ft

Bandwidth
Always-ON,
4K frames,
1 frame = 24 MB

Offline
Offline, roaming,
poor connection

Cost
Streaming Cost,
perf/$$
Edge TPU

4 TOPS | 2 Watts

INT8 | INT16 | TFLite

Inference accelerator

- Optimized for vision applications and convolutional neural networks
- Runs concurrent state-of-the-art models on high-resolution video, at real-time (30+ FPS)
- Full support for quantized TensorFlow Lite models

Coral boards feature the Edge TPU. A purpose-built ASIC designed to bring inference to the edge
Coral Edge TPU
Coral product portfolio

Dev Board
A single-board computer with a removable system-on-module (SOM) featuring the Edge TPU.
Available now
Price $149.99

USB Accelerator
A USB accessory featuring the Edge TPU that brings ML inferencing to existing systems.
Available now
Price $74.99

PCI-E Accelerator
PCI-E device for easy integration of Edge TPU into existing systems.
Available now
Price $34.99

SOM
A fully integrated System on Module in a 40mm x 48mm pluggable module.
Available now
Price $114.99
Features

- Edge TPU Module (SOM)
  - NXP i.MX 8M SOC (Quad-core Cortex-A53, plus Cortex-M4F)
  - Google Edge TPU ML accelerator coprocessor
  - Cryptographic coprocessor
  - Wi-Fi 2x2 MIMO (802.11b/g/n/ac 2.4/5GHz), Bluetooth 4.1
  - Vivante GC7000L GPU
  - 8GB eMMC
  - 1GB LPDDR4
- USB connections
  - USB Type-C power port (5V DC)
  - USB 3.0 Type-C OTG port
  - USB 3.0 Type-A host port
  - USB 2.0 Micro-B serial console port
- Audio connections
  - 3.5mm audio jack (CTIA compliant)
  - Digital PDM microphone (x2)
  - 2.54mm 4-pin terminal for stereo speakers
- Video connections
  - HDMI 2.0a (full size)
  - 39-pin FFC connector for MIPI-DSI display (4-lane)
  - 24-pin FFC connector for MIPI-CSI2 camera (4-lane)
- MicroSD card slot
- Gigabit Ethernet port
- 40-pin GPIO expansion header (Raspberry Pi style)
- Supports Debian Linux (Mendel)
USB 3.0 accessory device featuring EdgeTPU

<table>
<thead>
<tr>
<th>Feature</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCU</td>
<td>Cortex-M0 MCU (power, thermal monitor)</td>
</tr>
<tr>
<td>TPU</td>
<td>Google ASIC</td>
</tr>
<tr>
<td>RAM Memory</td>
<td>11 MB</td>
</tr>
<tr>
<td>Power</td>
<td>USB 3.0, Type-C connector</td>
</tr>
<tr>
<td>Data</td>
<td>USB 3.0, Type-C connector</td>
</tr>
<tr>
<td>Supported OS</td>
<td>Debian/Raspbian Linux</td>
</tr>
<tr>
<td>Supported ML</td>
<td>TensorFlow, TF Lite</td>
</tr>
<tr>
<td>Supported Models</td>
<td>Inception v1/2/3, Mobile Net, Daredevil</td>
</tr>
</tbody>
</table>

Coral
Software Toolchain

The system software and toolchain to allow developers to run, manipulate, and create with our products.

**Mendel OS**
A fork of the Debian OS to power our Intelligence Boards, and a

**C++ & Python SDK**
APIs to low level connections

**Edge TPU Compiler**
Converts TF graphs to run on targeted chipsets

**Companion Software**
Abstracts away traditional board management/coding in a high-level program
Code Example with Object Detection

# Initialize engine.
engine = DetectionEngine(args.model)
labels = ReadLabelFile(args.label) if args.label else None

# Open image.
img = Image.open(args.input)
draw = ImageDraw.Draw(img)

# Run inference.
ans = engine.DetectWithImage(img, threshold=0.05, keep_aspect_ratio=True,
                             relative_coord=False, top_k=10)

if ans:
    for obj in ans:
        box = obj.bounding_box.flatten().tolist()
        # Draw a rectangle.
        draw.rectangle(box, outline='red')
Demo

Edge TPU inference
person detection and privacy protection
with pre-compiled posenet model
How about custom models?
AutoML
Learning to Learn
How it works with Cloud
Coral + Google Cloud ML solutions

- Define & analyze your data
  - Use Google’s Data Labeling Service if needed

- AutoML: Train your model
  - Feature engineering
  - Model selection
  - Hyperparameter tuning

- Cloud IoT Core

- (Re)Training

- Data Store

- Edge TPU
Demo

Edge TPU inference
with AutoML custom sushi model
Demos are cool..
but we live in the Real World™
Customer Case
Manufacturing Visual Inspection
The Problem

Multiple objects partially visible

Target object moves around, rotated at random angles

Defects are small blemishes on the object
Customer Initial Testing
Standard AutoML Usage

<table>
<thead>
<tr>
<th>True label</th>
<th>Predicted label</th>
<th>ng_xbp</th>
<th>ng_dp</th>
<th>ng_sp</th>
<th>ok</th>
</tr>
</thead>
<tbody>
<tr>
<td>ng_xbp</td>
<td>14.3%</td>
<td>71.4%</td>
<td>-</td>
<td>14.3%</td>
<td>14.3%</td>
</tr>
<tr>
<td>ng_dp</td>
<td>-</td>
<td>71.4%</td>
<td>14.3%</td>
<td>14.3%</td>
<td>14.3%</td>
</tr>
<tr>
<td>ng_sp</td>
<td>-</td>
<td>-</td>
<td>71.4%</td>
<td>28.6%</td>
<td>14.3%</td>
</tr>
<tr>
<td>ok</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>100.0%</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>True label</th>
<th>Predicted label</th>
<th>ng</th>
<th>ok</th>
</tr>
</thead>
<tbody>
<tr>
<td>ng</td>
<td>56.3%</td>
<td>43.8%</td>
<td></td>
</tr>
<tr>
<td>ok</td>
<td>-</td>
<td>100.0%</td>
<td></td>
</tr>
</tbody>
</table>
The Solution

Pre-processing
Pre-processing Steps

Rough crop of target object

A defect
Pre-processing Steps

Remove background
- Filter background pixels to zero
- Remaining non-zero pixels are the object
Pre-processing Steps

Detect edges
Pre-processing Steps

Detect lines with Hough transform
- Calculate rotation angle
- Remove outliers
Pre-processing Steps

(Counter) rotate image to zero degrees
Pre-processing Steps

Detect bounding box and crop tightly

→ Training image for AutoML
EdgeTPU Test Results using Coral DevBoard

Confusion Matrix

<table>
<thead>
<tr>
<th></th>
<th>Predicted NG</th>
<th>Predicted OK</th>
</tr>
</thead>
<tbody>
<tr>
<td>True NG</td>
<td>95.10%</td>
<td>4.90%</td>
</tr>
<tr>
<td>True OK</td>
<td>1.47%</td>
<td>98.53%</td>
</tr>
</tbody>
</table>

Performance

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average inference latency</td>
<td>8.98 ms</td>
</tr>
</tbody>
</table>
Demo

Data Pre-processing & Coral Edge TPU inference
CPU

Edge TPU

“Ok” / “Not ok”
The model is not static tho..
We need lifecycle management
Thank You!