Classification of Depth Image on AI Microcontroller

Jesse Santos

March 27, 2023
AI REVOLUTION
Gap Between Big Machines and Little Machines

Unconstrained

Constrained by cost, power, size
Time-of-Flight System

Sensor Module → Depth Processing

- Emitted Light
- Received Light
- Pixel Tap A
- Pixel Tap B

Phase Shift

Software / SDK

- Python
- Open3D
- VHDL
- OpenCV
- MATLAB
- ROS

User Applications
Hardware Accelerated Classification

Sensor Module

Illumination Module

Hardware Depth Engine

Emitted Light

Received Light

Pixel Tap A

Pixel Tap B

Phase Shift

Neural Network Accelerator

- Highly Optimized for Deep Convolutional Neural Networks
- 442k 8-Bit Weight Capacity with 1,2,4,8-Bit Weights
- Programmable Input Image Size up to 1024 x 1024 pixels
- Programmable Network Depth up to 64 Layers
- Programmable per Layer Network Channel Widths up to 1024 Channels
- 1 and 2 Dimensional Convolution Processing
- Streaming Mode
- Flexibility to Support Other Network Types, including MLP and Recurrent Neural Networks

Software / SDK

- Python
- Open3D
- VHDL
- OpenCV
- FPGA Chip
- Verilog

User Applications

- Python
- VHDL
- Cython
- FPGA Chip
- Verilog
- MATLAB
- ROS

INTEGRATED'S NEURAL NETWORK ACCELERATOR is Artificial Intelligence in Battery-Powered Devices

All inference at less than 1/100th the energy of other embedded solutions.
Hardware Accelerated Classification

ToF Module

ADTF3175

Depth Image

Contrast Enhance

Binarize

Background Removed

Neural Network Accelerator

MAX78000EVKIT

Depth Data, Sensor Data

Label

Depth Image Processing:
- Contrast Enhance
- Binarize
- Background Removed

Applications

Repetition
Max-Pooling
Batch Normalization
Down-sampling
ReLU Activation
Dropout

Depth Image
Training methodology using Ai8x Framework

1. Training
   ai8x-training
   - Dataset + Data Loader
   - Quant-Aware Training
   - Quantization
   - Checkpoint File (Weights)
   - Evaluation

2. Synthesis
   ai8x-synthesis
   - Model Description (YAML)
   - MAX7800X Synthesis
   - Input Data Sample

3. Deployment
   Embedded SDK
   - Live Data Acquisition
   - Embedded C Code
   - UI / Output Processing
   - MAX7800X

Software Tools
- PyTorch
- TensorFlow
- Keil Tools by ARM
- Eclipse
- IAR Systems

MAX78000EVKIT
MAX78000FTHR

https://github.com/MaximIntegratedAI
CNN MODEL

POINT CLOUD DATASET

DEPTH MAP DATASET

Princeton 3D Object Dataset

AUGMENTATION

FLIP

ROTATE

BLUR

SIMPLENET

Repetition 2x
Max-Pooling
Batch Normalization B
Down-sampling S
ReLU Activation R
Dropout D

Conv 1/128 3x3/1/1 S R
Conv 2/128 3x3/1/1 B S R
Conv 3/128 3x3/1/1 B S R
Conv 4/128 3x3/1/1 B S R
Conv 5/128 3x3/1/1 B S R
Conv 6/128 3x3/1/1 B S R
Conv 7/128 3x3/1/1 B S R
Conv 8/128 3x3/1/1 B S R
Conv 9/128 3x3/1/1 B S R
Conv 10/128 3x3/1/1 B S R
Conv 11/128 1x1/1/1 B S R
Conv 12/128 3x3/1/1 B S R
Conv 13/128 3x3/1/1 B S R

Image Count

Bathtub 156
Bed 615
Chair 989
Desk 286
Dresser 286
Monitor 565
Nightstand 286
Sofa 492
Table 444
Toilet 0

27 March 2023
CNN MODEL - EVALUATION

- Validation accuracy, ~\textbf{94\%}
- Training is quantization-aware

### Training Hyperparameters

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train/Validate (%)</td>
<td>80/20</td>
<td>Multi-step Learning Rate (MLR)</td>
<td>0.001</td>
</tr>
<tr>
<td>Epochs</td>
<td>50</td>
<td>Gamma</td>
<td>0.1</td>
</tr>
<tr>
<td>Batch Size</td>
<td>128</td>
<td>MLR milestones (Epoch)</td>
<td>10, 30, 40</td>
</tr>
<tr>
<td>Optimizer Algorithm</td>
<td>Adam</td>
<td>Loss Function</td>
<td>Categorical Cross Entropy</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Start of Quantization (Epoch)</td>
<td>30</td>
</tr>
</tbody>
</table>

### Accuracy

- **Validation accuracy**: \textbf{\approx 94\%}

### Loss

- **Loss**
  - Train
  - Validation
  - Best Checkpoint
### CNN MODEL - EVALUATION

#### CONFUSION MATRIX AFTER QUANTIZATION

<table>
<thead>
<tr>
<th></th>
<th>bathtub</th>
<th>bed</th>
<th>chair</th>
<th>desk</th>
<th>dresser</th>
<th>monitor</th>
<th>nightstand</th>
<th>sofa</th>
<th>table</th>
<th>toilet</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>True</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Prediction</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>bathtub</td>
<td>29</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>9</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>bed</td>
<td>1</td>
<td>97</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>chair</td>
<td>0</td>
<td>0</td>
<td>94</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>desk</td>
<td>5</td>
<td>0</td>
<td>1</td>
<td>48</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>20</td>
</tr>
<tr>
<td>dresser</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>64</td>
<td>6</td>
<td>14</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>monitor</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>94</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>nightstand</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>25</td>
<td>0</td>
<td>45</td>
<td>1</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>sofa</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>94</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>table</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>84</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>toilet</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>95</td>
</tr>
</tbody>
</table>

- **High correct classification**
  - \( \geq 94\% \)
  - Bed, Chair, Monitor, Sofa, Toilet

- **Worst performance**
  - Desk misclassified as table, vice versa
  - Dresser misclassified as nightstand, vice versa
Inference accuracy results:
- Chair, 89.4%
- Monitor, 89.8%
- Sofa, 90.7%
- Table, 79.5%

Inference time:
- 14ms ± 0.14ms → 71 images/second

Inference Energy Cost:
- 360 μJ per inference
- 35 mW on activity

Energy Measurements:

<table>
<thead>
<tr>
<th>Variable</th>
<th>Kernel</th>
<th>Input</th>
<th>Input + Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>E (J)</td>
<td>2.1×10^-4</td>
<td>2.7×10^-10</td>
<td>3.6×10^-4</td>
</tr>
<tr>
<td>T (s)</td>
<td>2.5×10^-2</td>
<td>3×10^-8</td>
<td>1.4×10^-2</td>
</tr>
<tr>
<td>I (W)</td>
<td>9×10^-3</td>
<td>9×10^-3</td>
<td>9×10^-3</td>
</tr>
<tr>
<td>A (W)</td>
<td>1.7×10^-2</td>
<td>1.8×10^-2</td>
<td>3.5×10^-2</td>
</tr>
</tbody>
</table>
Takeaway

- MAX78000 enables machine learning on the edge
- ADTF3175 compliments MAX78000 for hardware accelerated AI
- Future work, on expansion of dataset
  - E.g. multiple perspectives in the point-cloud space
  - E.g. more captured images by ToF
Thank You!

For more information, visit these links:

- **MAX78000**: https://www.analog.com/max78000
- **ADTF3175**: https://www.analog.com/adtf3175
- **AI8X**: https://github.com/MaximIntegratedAI

Jesse.santos@analog.com
Mariel.tinaco@analog.com
Carlos.bating@analog.com
Neon.dejesus@analog.com
Copyright Notice

This presentation in this publication was presented at the tinyML® Research Symposium (March 27, 2023). The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.

www.tinyml.org