Convolutional Neural Networks on embedded automotive platforms: a qualitative comparison

Paolo Burgio, Gianluca Brilli, Antonio Marra, Marko Bertogna
University of Modena and Reggio Emilia

paolo.burgio@unimore.it
Neural network for tomorrow

› Extensively adopted in the embedded world

› Computer vision and image processing tasks,
  - object categorization and labeling

› Autonomous driving, industry 4.0
A lot of stuff to do...

- Multiple bidimensional layers
- Huge number of multiply-accumulate (MAC) operation
- on thousands of pixel of an input image
At low SWaP, training is "easy" with powerful servers for "big data"-sets.

Inference is an issue for on-vehicle ECUs, in-plant boards constrained in Size, Weight and Power.
...which architecture?

Accelerator

Core

Core

(Host) Memory

"pure" GP-many-cores

GP-GPUs

Reconfigurable logics

DL accelerators
This (ongoing) work

Profile open-source packages...
...of state-of-the-art (C)NNs...
...on automotive platforms

Three categories
› Present (Embedded GP-GPUs)
› (Next) future (Reconfigurable/FPGAs)
› (Next-next) future
Today

Nvidia Parker SoC
✓ Drive PX2 for autonomous driving
✓ 4 x ARM Cortex 57 + 2 x Denver
✓ Pascal GPU

Xilinx Zynq Ultrascale+
✓ 4 x ARM Cortex A53 + 2 x R5
✓ Mali GPU
✓ FPGA fabric
Yolo
Yolo on Tegra X2: FPS
Yolo on Tegra X2: power

Power (Board) @19V
Yolo on XU+

› Claimed 40FPS
› BNN (good for FPGA)
AlexNet
(PipeCNN)
AlexNet on TX2

- Classification, not detection (as Yolos)
  - Lighter

- 5 conv. layer, fully-connected last layer, 76% precision
  - [https://github.com/opencv/opencv_extra/blob/master/testdata/dnn/bvlc_alexnet.prototxt](https://github.com/opencv/opencv_extra/blob/master/testdata/dnn/bvlc_alexnet.prototxt)
AlexNet on TX2

› Classification, not detection (as Yolos)
  – Lighter

› 5 conv. layer, fully-connected last layer, 76% precision
AlexNet on TX2

- Classification, not detection (as Yolos)
  - Lighter

- 5 conv. layer, fully-connected last layer, 76% precision
  - [https://github.com/opencv/opencv_extra/blob/master/testdata/dnn/bvlc_alexnet.prototxt](https://github.com/opencv/opencv_extra/blob/master/testdata/dnn/bvlc_alexnet.prototxt)
AlexNet on XU+

› Clocked 2-3x than TX2
  - No clock scaling with xfDNN engine

<table>
<thead>
<tr>
<th>Avg E2E latency (ms)</th>
<th>Throughput (FPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4 Ghz 16,24</td>
<td>2.4 Ghz 61,554</td>
</tr>
<tr>
<td>300MHz</td>
<td>300MHz</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Board average power</th>
<th>SoC average power</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>2.4 Ghz 23,1165</td>
<td>2.4 Ghz 0,5097</td>
</tr>
<tr>
<td>300MHz</td>
<td>300MHz</td>
</tr>
</tbody>
</table>

Power comparison SoC XU+ and TX2 (max freq)[Watt]

<table>
<thead>
<tr>
<th></th>
<th>XU+ 300MHz</th>
<th>TX2 114,75MHz</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4 Ghz</td>
<td>0,5097</td>
<td>0,68751</td>
</tr>
<tr>
<td>1,11 Ghz</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
ZynqNet on Tegra X2

- Classification
- 28 layers, 83% precision
  - [https://dgschwend.github.io/netscope/#/preset/zynqnet](https://dgschwend.github.io/netscope/#/preset/zynqnet)
ZynqNet on Tegra X2

› Classification

› 28 layers, 83% precision
  
  - [https://dgschwend.github.io/netscope/#/preset/zynqnet](https://dgschwend.github.io/netscope/#/preset/zynqnet)
ZynqNet on UC+

> 78% accuracy (83% on TX2)

<table>
<thead>
<tr>
<th>Throughput (FPS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>300 Mhz</td>
</tr>
<tr>
<td>2.4 Ghz</td>
</tr>
<tr>
<td>77.47</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Board average power</th>
</tr>
</thead>
<tbody>
<tr>
<td>300 Mhz</td>
</tr>
<tr>
<td>2.4 Ghz</td>
</tr>
<tr>
<td>22.9953</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>SoC average power</th>
</tr>
</thead>
<tbody>
<tr>
<td>300 Mhz</td>
</tr>
<tr>
<td>2.4 Ghz</td>
</tr>
<tr>
<td>0.3885</td>
</tr>
</tbody>
</table>
Discussion

› GPU use Caffe engine, FPGA xFDNN engine
  – Highly-optimized for GPU (impressive performance for ZynqNet and AlexNet
  – Thanks to FP data
  – 16-bit int ops on FPGA => not optimal

› Caffè engine + demos from Xilinx not programmable, not flexible (only 1 image)

› XU+ carrier board - SoC as power efficient as TX2

Next

› (Consolidate XU+ results and) work on MPPA μcore

› PipeCNN on Altera's DE5Net board?
  – Claimed: AlexNet 67 FPS @27W, VGG16 0.7FPS @30W
State-of-the-art CNNs

› You-Only-Look-Once
  – Tiny YOLO

› AlexNet
  – PipeCNN

› ZynqNet
  – ETHZ

› ....
The present: NVIDIA Tegra X2

› 256 core Pascal GP-GPU
The (next) future: Xilinx Ultrascale+

- Xilinx Ultrascale+ EG/EV
  - With GPU Mali-400MP2
Yolo on Tegra X2: E2E latency

![E2E latency graph](image)

Legend: 0-0.5, 0.5-1, 1-1.5

Latency values: 0, 0.5, 1, 1.5, 2, 2.5, 3
Yolo tiny
Yolo tiny on Tegra X2

- 15 layer (half than Yolo)
- **Reported 57.1% mean avg Precision vs 78.6% Yolo**
Yolo tiny on Tegra X2

› 15 layer (half than Yolo)
› **Reported** 57.1% mean avg Precision vs 78.6% Yolo
Yolo tiny on Tegra X2: power

- 15 layer (half than Yolo)
- Reported 57.1% mean avg Precision vs 78.6% Yolo
Tinier yolo on UC+

- 7 layers vs. 15 (tiny yolo) vs. 30 (Yolo)

AlexNet on TX2

- Classification, not detection (as Yolos)
  - Lighter

- 5 conv. layer, fully-connected last layer, 76% precision
  - [https://github.com/opencv/opencv_extra/blob/master/testdata/dnn/bvlc_alexnet.prototxt](https://github.com/opencv/opencv_extra/blob/master/testdata/dnn/bvlc_alexnet.prototxt)
ZynqNet on Tegra X2

- Classification
- 28 layers, 83% precision
  - [https://dgschwend.github.io/netscope/#/preset/zynqnet](https://dgschwend.github.io/netscope/#/preset/zynqnet)