

# Enabling Neural Network Edge Computing on a Small Robot Vehicle

Presentation at IDC 2022

Jan Brederke




**HSB**

Hochschule Bremen  
City University of Applied Sciences

14 Sep 2022

## Legend for the Slide Handout

This handout comprises all slides shown. Additionally, it comprises notes for oral explanations. The notes are marked with “” in the head line (like this page).

# Overview

Enabling Neural Network Edge Computing on a Small Robot Vehicle

- 1 Motivation
- 2 The Sample Application
- 3 Measures for Performance Improvements Investigated
- 4 Implementation Experiments

# Edge Computing

## Motivation

### Cloud Computing

Using computing resources  
in data centers over the Internet.

### Edge Computing

computing close to sensors and actors,  
possibly even without permanent  
data and power supply connections



### Cloud Computing:

... for neural networks:  
powerful and power consuming  
special hardware

### Edge Computing:

... for neural networks:  
resources are restricted

- **autonomous vehicle**: often at the edge of the Cloud

# Our Particular Motivation for an Autonomous Vehicle: On-Board Computer in Spacecraft

## Motivation



photo: NASA

### special characteristics of an on-board computer:

- access to ground segment: intermittent only
- **lame** special processor, tolerating space radiation (processing resources about like a Raspberry Pi)
- extremely small production numbers:  
→ usually made with FPGA instead of ASIC

# On-Board Computers in Spacecraft

## Motivation

### space radiation:

injects electrical charges → malfunctions

destroys the chip → permanent failure

effect stronger, if structures on the chip are smaller

### special processor:

structures of  $\geq 65$  nm: still robust

current standard processor structures: 7 nm

special processor: much less data processing resources

### FPGA:

“field programmable gate array”

suitable for small production numbers,

no basic costs for mask production, . . .

### ASIC: “application specific integrated circuit”

# Increasing Demand for On-Board Computing Resources

## Motivation

for

- on-board image processing,  
for autonomous rovers on other celestial bodies
- constellations of nano-satellites  
each with narrow bandwidth to the ground segment
- ...

# Similar Challenges on Earth for the Internet of Things (IoT)

## Motivation

### scarce computing resources

- often due to cost restrictions for mass products
- when battery powered

# Our Research Question

## Motivation

How to make full use of neural networks on FPGAs  
as restricted as radiation tolerant FPGAs?

Or, more general:

... as restricted as in Edge Computing?

# Overview

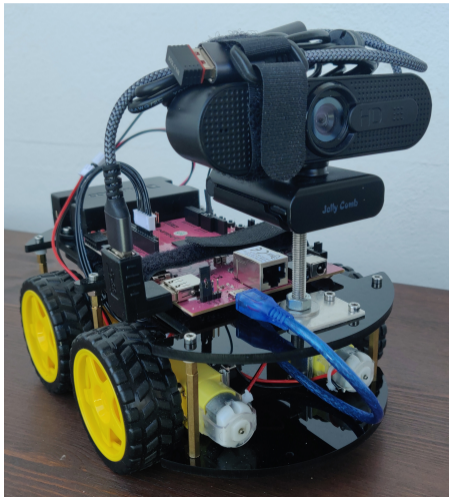
Enabling Neural Network Edge Computing on a Small Robot Vehicle

- 1 Motivation
- 2 The Sample Application
- 3 Measures for Performance Improvements Investigated
- 4 Implementation Experiments



# The Sample Application

## The Sample Application



- autonomous model car: proxy for an autonomous robotic space craft
- with camera and system-on-chip (SoC) Xilinx Zynq-7020
- SoC: with Arm CPU and **FPGA Artix-7**
- Artix-7: resources similar to an FPGA suitable for space

photo: Felix Müller

# The Task of the Vehicle

## The Sample Application

- visually recognize a person in view,
- follow it when it moves, and
- obey to driving commands by simple gestures;
- all of this autonomously and in real-time

# Overview

Enabling Neural Network Edge Computing on a Small Robot Vehicle

- 1 Motivation
- 2 The Sample Application
- 3 Measures for Performance Improvements Investigated
- 4 Implementation Experiments

# Measures for Performance Improvements Investigated

Measures for Performance Improvements Investigated

Measures for Performance Improvements Investigated:

... for neural network processing

# Application-Independent Measures for Improvements

Measures for Performance Improvements Investigated

## measures used on our vehicle

- use a pre-trained neural network for **inference only** on the vehicle
- use **hardware acceleration** for the neural network **by an FPGA**  
(with the FPGA choice obeying the restrictions of space engineering)
- employ a **quantized neural network**  
to reduce the hardware resources for the neural network

# Application-Independent Measures for Improvements

Measures for Performance Improvements Investigated

rationales for:

inference only:

inference is an order of magnitude cheaper than training

hardware acceleration by an FPGA:

a neural network is inherently concurrent, and  
an FPGA offers massively parallel execution

quantized neural network:

see next slide

# What is a Quantized Neural Network?

Measures for Performance Improvements Investigated

## standard neural network

values, weights, ...

represented as 32-bit floating point

## quantized neural network

values, weights, ...

represented as **few-bit integer**  
or even as **binary**

- less precision per node
- **much less chip area** per node,  
**much less power consumption** per node

# What is a Quantized Neural Network?

Measures for Performance Improvements Investigated

- more nodes per chip
  - same precision
  - and **in total:**
    - less chip area, less power consumption**
- quantized arithmetics:  
well suited for FPGAs



# Application-Specific Measures for Improvements

Measures for Performance Improvements Investigated

## measures used on our vehicle

- **split the input image into tiles** and process them sequentially to reduce hardware resources for the neural network
- **separate locating a person in a frame and determining the person's gesture** to reduce the number of tiles required, and thus the number of iterations

# Application-Specific Measures for Improvements (1)

Measures for Performance Improvements Investigated

rationales for:

splitting into tiles:

task doesn't fit the chip area by far  
→ partially sequentialize the task,  
thus making an area/speed trade-off

probably better:

to sequentialize processing the frame than  
to sequentialize computing the neural network,  
because of chip bandwidth

## Application-Specific Measures for Improvements (2)

Measures for Performance Improvements Investigated

rationales for:

separate locating a person and determining the gesture:

- recognizing a small but complex object requires having many tiles

- finding a simple object requires less tiles

- determining the gesture in a single “region of interest” requires one pass only

# Overview

Enabling Neural Network Edge Computing on a Small Robot Vehicle

- 1 Motivation
- 2 The Sample Application
- 3 Measures for Performance Improvements Investigated
- 4 Implementation Experiments

# Experiment 1:

## Tracking a Moving Road Sign

Implementation Experiments



video by:

R. Kaehn,  
N. Krekel,  
Pablo Navarro,  
Felix Müller

## Experiment 1:

# Tracking a Moving Road Sign

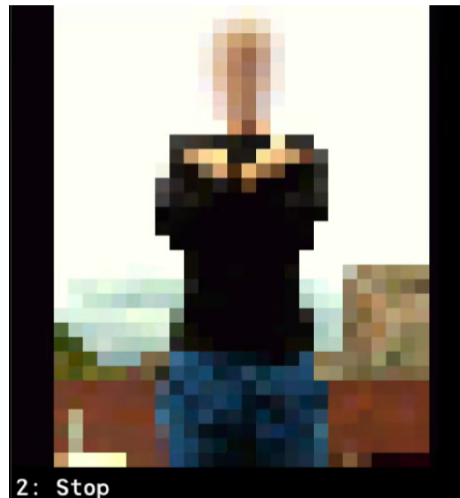
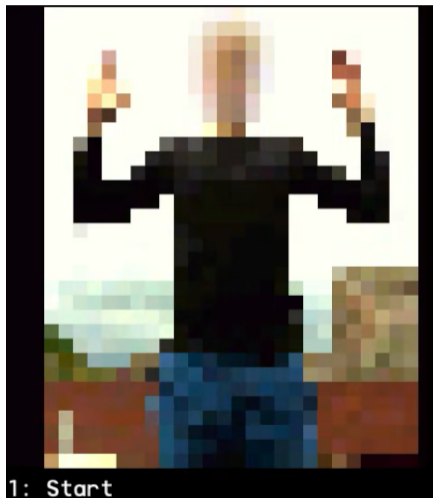
Implementation Experiments

- task of vehicle:  
track and follow a person carrying a road sign
- use a pre-trained network from other people, for simplicity
- only neural network on FPGA,  
input/output processing on CPU
- each video frame split into 1107 tiles of different sizes, fed into FPGA sequentially; resulting frame rate: 2 fps
- FINN framework for compiling the neural network into the FPGA
- FPGA resources used:  
block RAM      70 %      → still some room for  
look-up tables  46 %      more hardware acceleration

# Experiment 2:

## Tracking a Moving Person That Commands by Gestures

Implementation Experiments



## Experiment 2:

# Tracking a Moving Person That Commands by Gestures

## Implementation Experiments

- task of vehicle:  
track and follow a person,  
obey simple arm gestures (“start”, “stop”, “no specific pose”)
- now: training by us
- again: FINN framework for compiling the neural network into the FPGA
- network model: written in Brevitas, a PyTorch library
- unfortunately, no performance measurements available,  
due to some implementation bugs



# Experiment 3:

## Tiling in Hardware

Implementation Experiments



Results:  
0 (airplane)  
4 (deer)  
2 (bird)  
8 (ship)

## Experiment 3:

# Tiling in Hardware

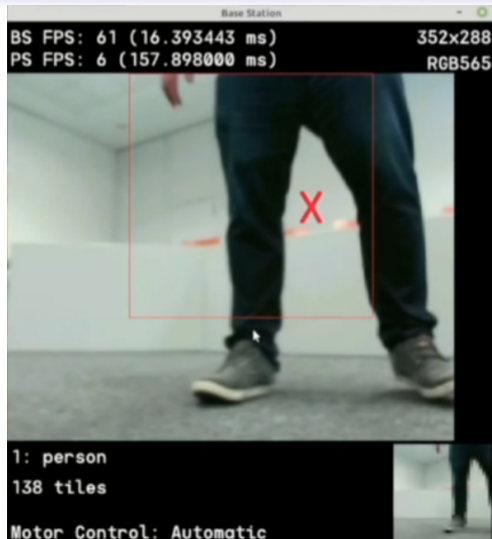
## Implementation Experiments

- do tiling on FPGA, too, instead of on CPU
- tested with:  
images from CIFAR10 data set in a video stream, differently scaled and placed
- measurements:  
FPGA @ 100 MHz was up to  $4\times$  faster than software solution on the CPU @ 650 MHz
- further optimizations are possible
- yet to be done: integration into our robot vehicle

# Experiment 4:

## Tracking a Moving Person and Only Then Looking for Gestures

Implementation Experiments



video by:

Ph. Altnickel,  
F. Cansu,  
I. Sastim,  
J. Steffen,  
M. Uhlenbrock

## Experiment 4:

# Tracking a Moving Person and Only Then Looking for Gestures

### Implementation Experiments

- separate locating a person and determining the person's gesture
- in screenshot: person is recognized even with parts missing
- optimum for number of tiles when following a person:  
138 tiles and 6 fps (instead of over 1000 tiles and 2 fps above)
- FPGA resources used:
  - block RAM      70 %
  - look-up tables 44 %
- therefore: tiling in hardware could be added easily
- gesture recognition still to be added  
could be done in software since it is not time critical

# Tool Chain Used (in Experiment 4)

## Implementation Experiments

- **FINN framework** for compiling neural networks into FPGA hardware, by FPGA manufacturer Xilinx
- **PyTorch** framework for training (in earlier experiments: Keras/Tensorflow)
- **Brevitas** software library for PyTorch for writing quantized neural networks in the FINN framework  
output is in the open format ONNX

# The FINN Stack

## Implementation Experiments

Customization  
of Algorithm



Customization  
of Hardware  
Architecture

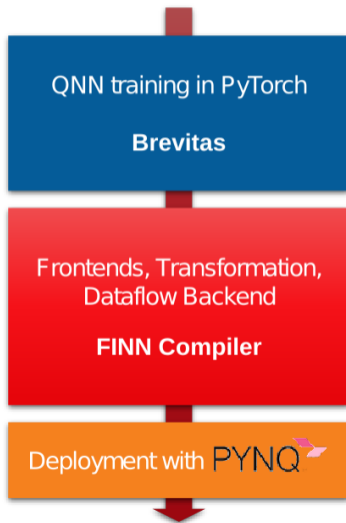





diagram: Xilinx

# Summary

## Enabling Neural Network Edge Computing on a Small Robot Vehicle

- computing resources at the edge: scarce  
(examples: on-board computer in spacecraft, IoT)
- investigated measures for improving performance of neural network processing there
  - inference only, FPGA, quantized neural network
  - sequentialize processing partially, determine region of interest roughly first
- experiments show promising results
- ongoing work

# References

-  Altnickel, Philipp, Tuncer Catalkaya, Jan Brederke, et al. (Sept. 1, 2021). *Gesten- und Objekterkennung durch schwache FPGAs in autonomen Fahrzeugen mittels neuronaler Netze*. Tech. rep. City Univ. of Applied Sciences Bremen, Germany. 153 pp.
-  Müller, Felix (June 23, 2021). “Dynamisches Tiling auf schwachen FPGAs zur Objekterkennung mithilfe kleiner neuronaler Netze”. *Bachelorthesis*. City Univ. of Applied Sciences Bremen, Germany.
-  Altnickel, Philipp, Ferhat Cansu, Jan Brederke, et al. (Mar. 1, 2022). *Personenerkennung durch schwache FPGAs in autonomen Fahrzeugen mittels Neuronaler Netze*. Tech. rep. City Univ. of Applied Sciences Bremen, Germany. 57 pp.