# Edge Processing Unit

## The Way to Efficient Edge Computing for Industry 4.0

Silvan Zahno [1,2], Sharam Dadashnia [1,3,4], Remo Schnyder [5]

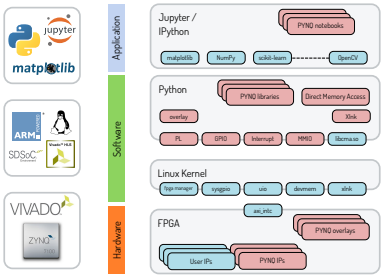
[1] Swiss Distance University of Applied Sciences, Brig, Switzerland; [2] University of Applied Sciences and Arts Western Switzerland Valais, Sion, Switzerland;
[3] Scheer GmBH, Saarbrücken, Germany; [4] Saarland University, Saarbrücken, Germany ; [5] Syrto AG, Raron, Switzerland

Industrie :: 2025

## Background

Edge computing has proven to be very valuable and is often needed in addition to cloud computing. Especially in areas such as image and video processing, where large amount of data is generated that cannot be transferred in their entirety to the cloud for processing, or in areas where security and real-time behavior are essential.

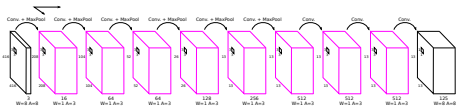
## Technologies




[1,3]

## Implementation

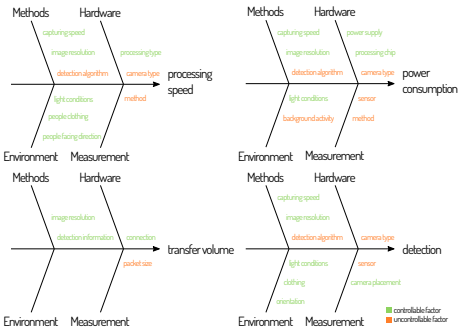
- Streamlit frontend dashboard
- Python backend application
- Tiny Yolo v3 CNN hardware accelerated
- MQTT broker for communication
- Datacollection and timestamp functionalities



[2]

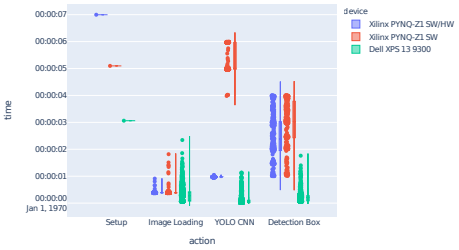
## Objectives

- Identify suitable usecase
- Developing an EPU capable of object detection with the help of an Hardware Accelerator
- Develop proof of concept (POC) system for running experiments and data collection.
- Evaluate the POC across four criteria
- Experiment parameters

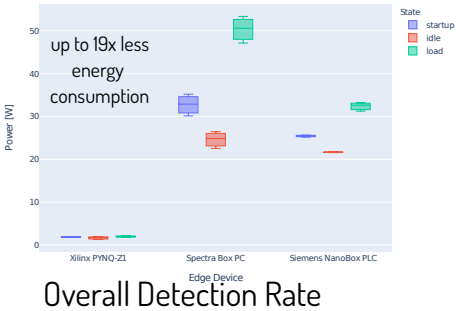



## Results

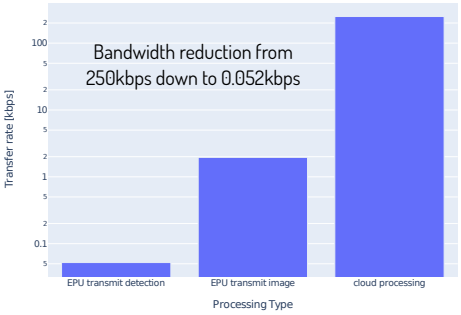
### Comparison FPGA vs. Software



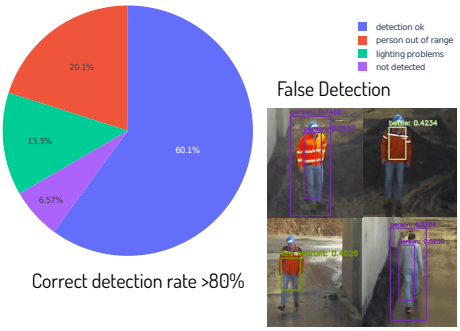

### Comparison Energy




### Comparison Data Transferrate



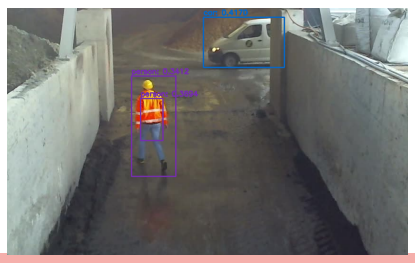

### Overall Detection Rate

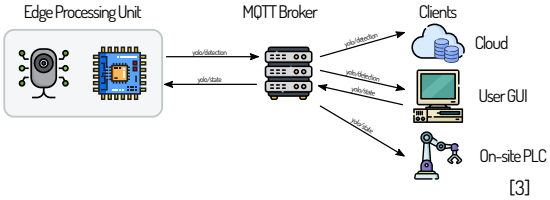



Correct detection rate >80%

## Use Case

- Highly automated slug treatment facility.
- Selected usecase contains an automatic crane, people verify the operation and wheelloader & truck deliver new to be processed material.
- 28 evaluated scenarios & >10'000 recorded images. 790 selected images processed.






[3]

## Discussion

- FPGA are well suited for machine learning especially neural net applications
- Toolchain has a high abstraction layer
- Additional development knowhow required
- Adaptive hardware for future expansion
- On-site processing reducing latency & bw

## References

[1] Xilinx. (2020, June). PYNQ - Python productivity for Zynq. http://www.pynq.io/
[2] Xilinx. (2020, December). QNN-MO-PYNQ. https://github.com/Xilinx/QNN-MO-PYNQ
[3] This poster has been designed using resources from Flaticon.com and Fontawesome.com

FFHS Fernfachhochschule Schweiz
Mitglied der SUPSI

syrto system robotic teleoperation

Hes·so VALAIS WALLIS School of Engineering

Get the source

Contact me