Boosting Autonomous Navigation solution based on Deep Learning using new rad-tol Kintex Ultrascale FPGA

European Workshop on On-Board Data Processing

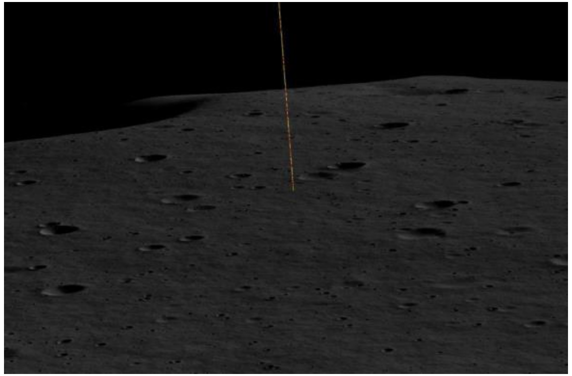
Teleconference 14th of June 2021

Dragos-Georgel Gogu
Avionics Engineer
GMV Innovating Solutions SRL
Contact: dgogu@gmv.com

INNOVATING SOLUTIONS

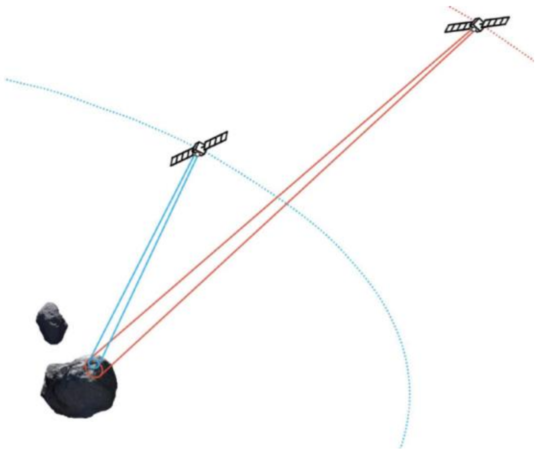
Contents

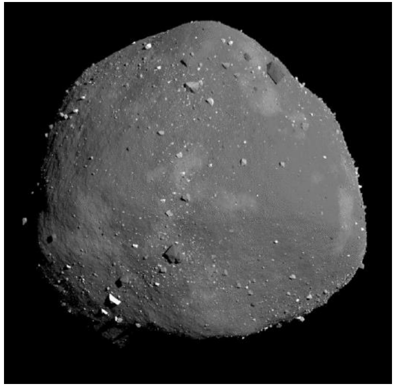
Context



Conventional Approach **VS** AI Based Approach

Moon Surface
Asteroid surface





Classical Design

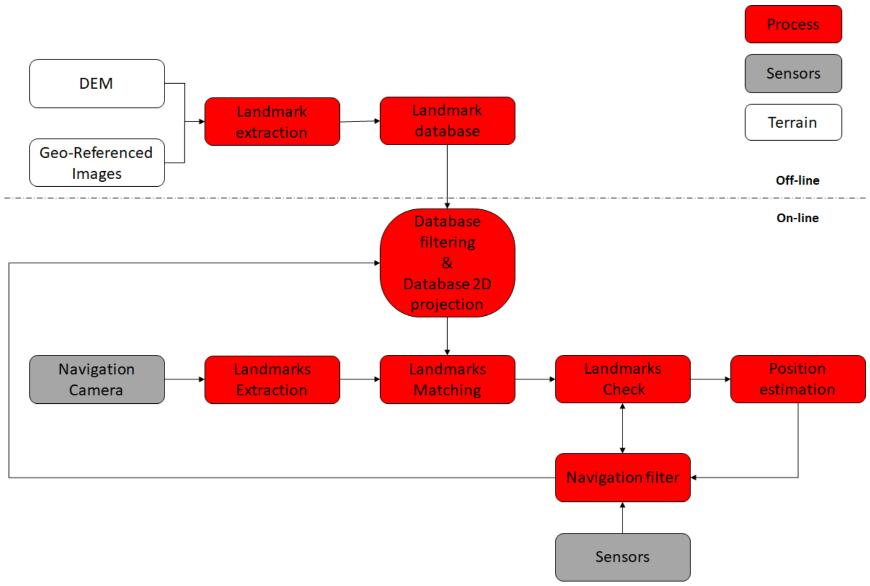
Conventional Absolute Navigation

Off-line:

- DEM & Geo Referenced Images

Landmark extraction

-Generation and Validation of the Landmark database



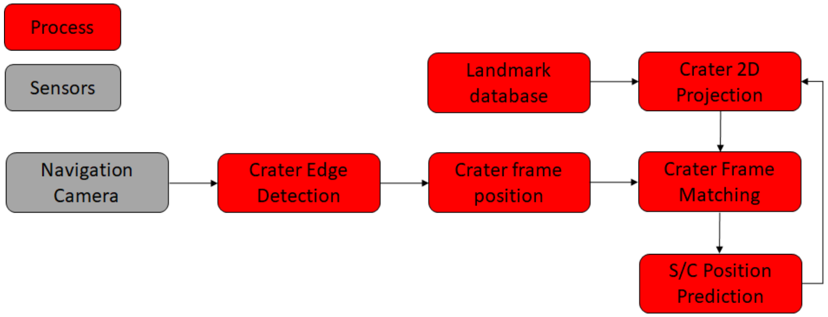
On-line:

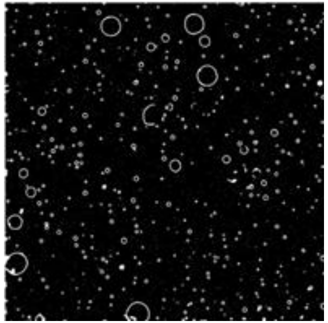
- Landmark extraction from navigation image
- Landmark matching with data base
- S/C states estimation using matched landmarks

AI Absolute Navigation Concept Design

NN-VISNAV

- Crater Edge Extraction based on Deep learning NN
- Deep Learning NN trained with different image datasets
- Trained NN use to extract the Landmarks on testing scenario
- NN implemented in dedicated FPGA will provide the edges detected in the image
- HW/SW co-design used to run Crater Frame Positon and Crater Frame Matching

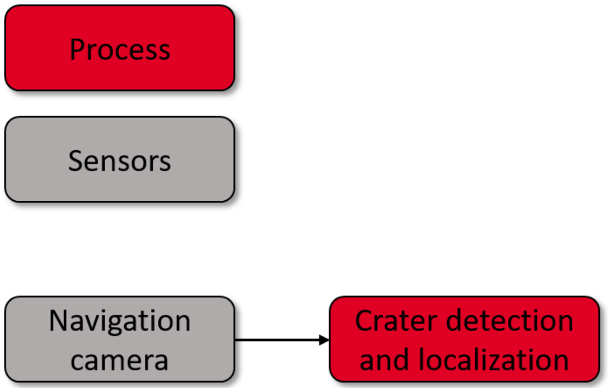




AI Absolute Navigation Concept Design

AITAG

- Crater detection and localization based on DNNs
- List of craters, their estimated size and their position
- Implemented in dedicated AI accelerator Myriad2



Patch matching on asteroid images concept design

AITAG observation asteroid scenario

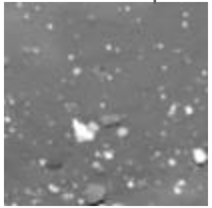
Matching reference patch within images from the camera

Potential improvements:

- Autonomous pinpointing of camera
- Filtering downlink of camera images, reducing throughput

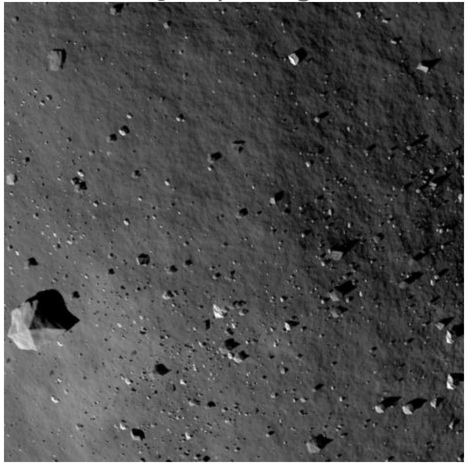
Based on using DNN that calculates similarity score between patch camera and subimage in query image

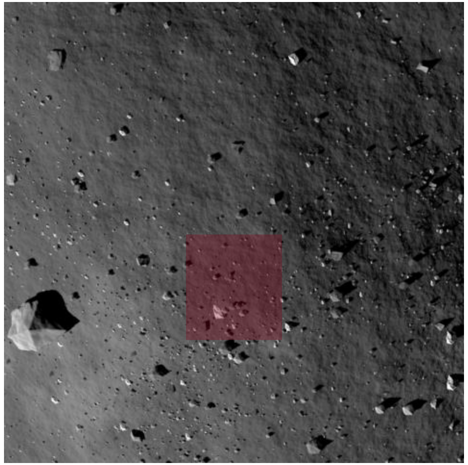
Reference patch



+

Query image

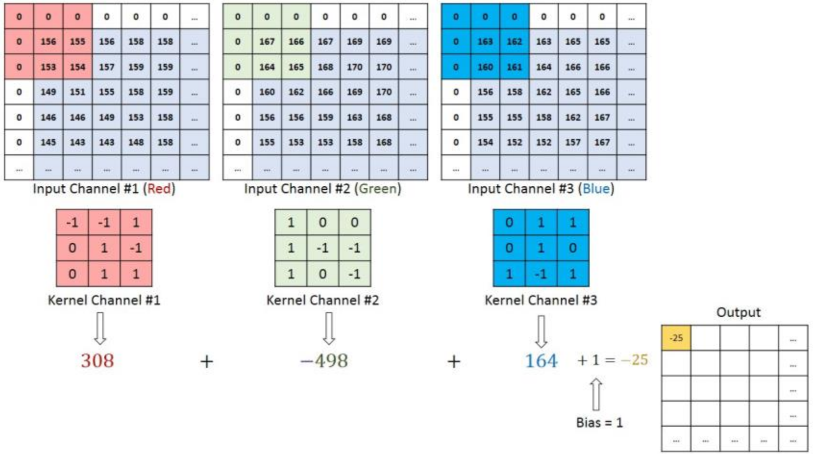




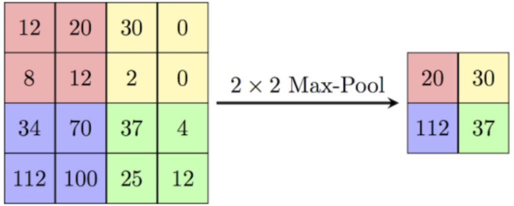
Detected patch

FPGA Firmware Development

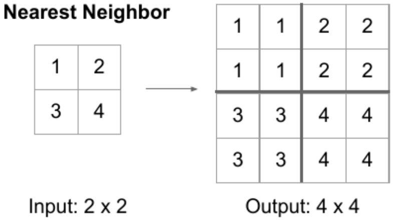
Convolution2D
- Network Parameters are called filters or kernels
- Filters are generated in the training process
- Applications in image and video recognition, recommender systems, image classification, medical image analysis
- Simple arithmetic: dot product between a portion of the image and a kernel. Then multiple convolutions are added together to create a single output channel



Max pooling2D module
- Keep the pixel with a higher value from a group a 4 adjacent pixels
- Resolution is reduced by a factor of 2 (4 times less pixels)
- It is a way of summarize information about an area of pixels



UpSampling 2D module
- Nearest Neighbor Scaling
- Duplicates input pixels vertically & horizontally



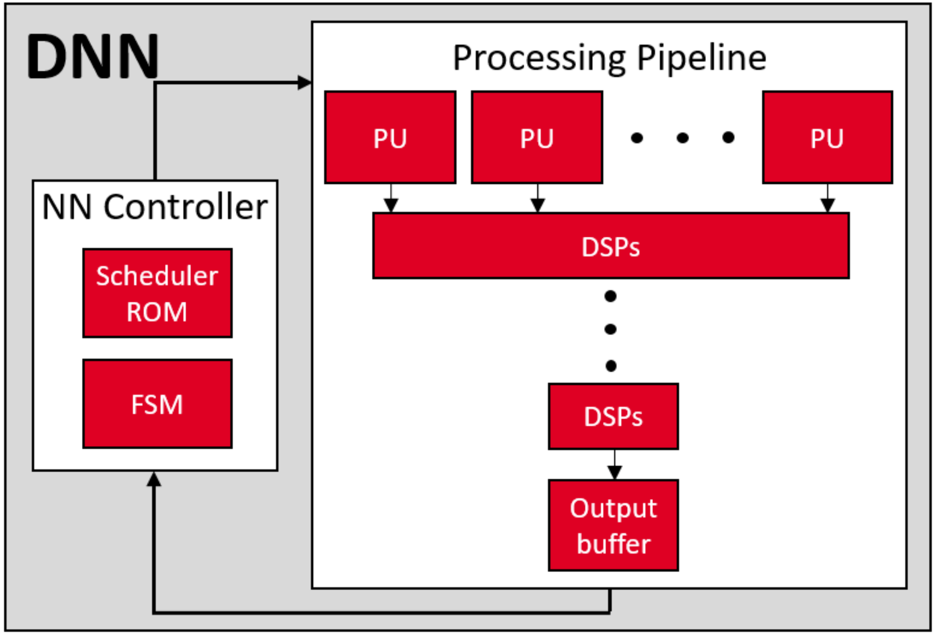
FPGA Design

Controller

- Finite State Machine
- Scheduler ROM
- Write/Read from DRAM
- Retrieval of parameters from DRAM
- Store and retrieval of intermediate layer results

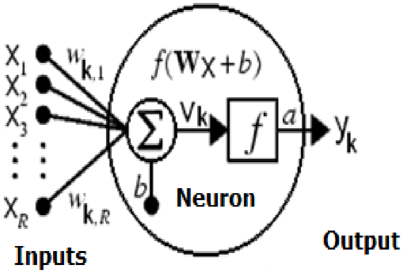
Processing pipeline

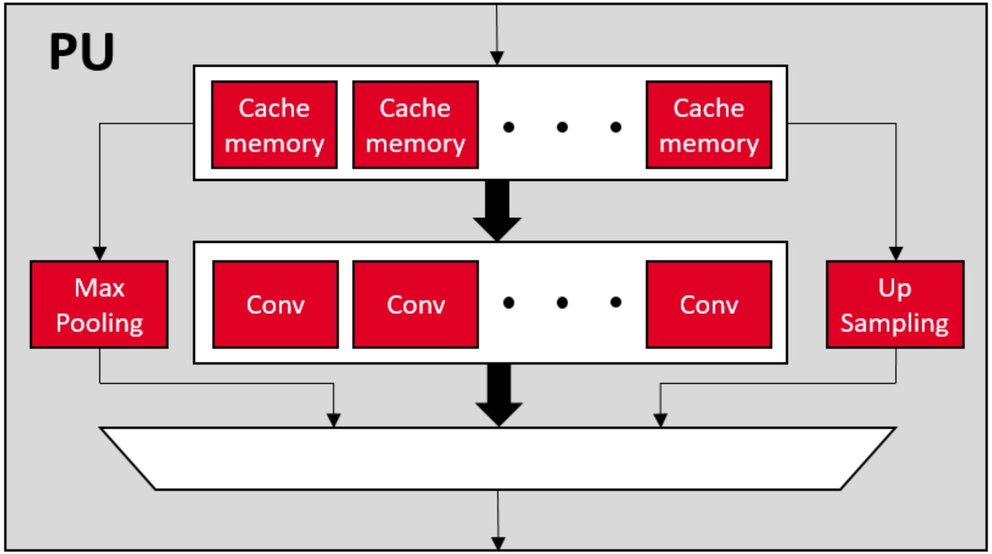
- Processing Units
- DSPs inverse pyramid
- Output buffer



FPGA Design

- Flexible design
- Possibility of trading off utilization of resources for speed
- More convolution blocks for improved speed
- Fewer convolution blocks for fewer utilization of resources
- More cache memory blocks for bigger feature maps capabilities





Avionics Architecture

FPGA-based acceleration of the DNN

OBC (client) + FPGA (inference server)

Ethernet connection

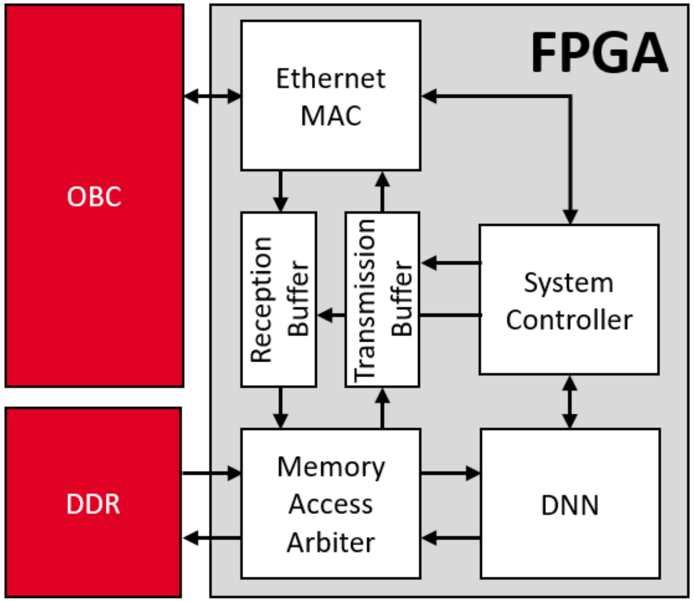
Custom protocol over raw Ethernet packets

Support for packet fragmentation and multi-packet acknowledge

Specific VHDL modules for each layer type

Parameterized

Offline generation of schedule instructions

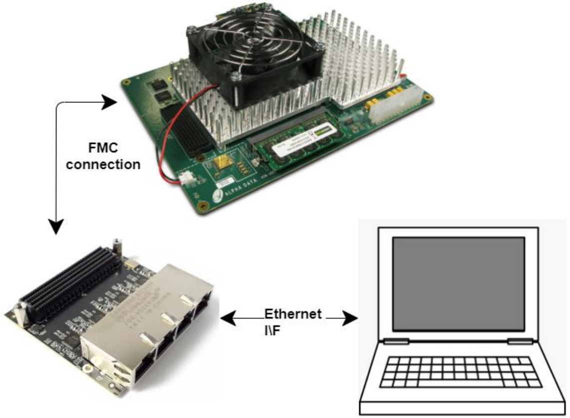


Avionics Architecture - Demonstrator

Alpha DEV Kit 2
Kintex UltraScale KCU060

Python command line interface

- loading the bitstream
- loading model parameters
- sending query images
- executing the inference
- retrieving results from any layer of the model
- monitoring the status of the system



Avionics Architecture

Moon Scenario

- To implement the pre trained neural network of the Moon Scenario in HW requires 30 different layers

Layer	Module Name	Processing units used
Layer 1	Convolution 2048x2048	1 PU
Layer 2	Max Pooling 2048x2048	1 PU
Layer 3	Convolution 1024x1024	9 PU
Layer 4	Max Pooling 1024x1024	1 PU
Layer 5	Convolution 512x512	9 PU
Layer 6	Convolution 512x512	10 PU
Layer 7	Max Pooling 512x512	1 PU
Layer 8	Convolution 256x256	10 PU
Layer 9	Convolution 256x256	7 PU
Layer 10	Max Pooling 256x256	1 PU
Layer 11	Convolution 128x128	7 PU
Layer 12	Convolution 128x128	14 PU
Layer 13	Max Pooling 128x128	1 PU
Layer 14	Convolution 64x64	14 PU
Layer 15	Convolution 64x64	14 PU
Layer 16	Up Sampling 64x64	1 PU
Layer 17	Convolution 128x128	14 PU
Layer 18	Convolution 128x128	14 PU
Layer 19	Up Sampling 128x128	1 PU
Layer 20	Convolution 256x256	11 PU
Layer 21	Convolution 256x256	7 PU
Layer 22	Up Sampling 256x256	1 PU
Layer 23	Convolution 512x512	13 PU
Layer 24	Convolution 512x512	10 PU
Layer 25	Up Sampling 512x512	1 PU
Layer 26	Convolution 1024x1024	14 PU
Layer 27	Up Sampling 1024x1024	1 PU
Layer 28	Convolution 2048x2048	14 PU
Layer 29	Convolution 2048x2048	9 PU
Layer 30	Convolution 2048x2048	9 PU

Test results

- Matched craters shows good uniformity displacement in FOV
- Using HW implementation it is possible to detected craters in high number also on low altitude
- Craters radius precision detection error is subpixel with respect to SW implementation
- Mean number of Craters matches between HW and SW is 171 for the entire trajectory

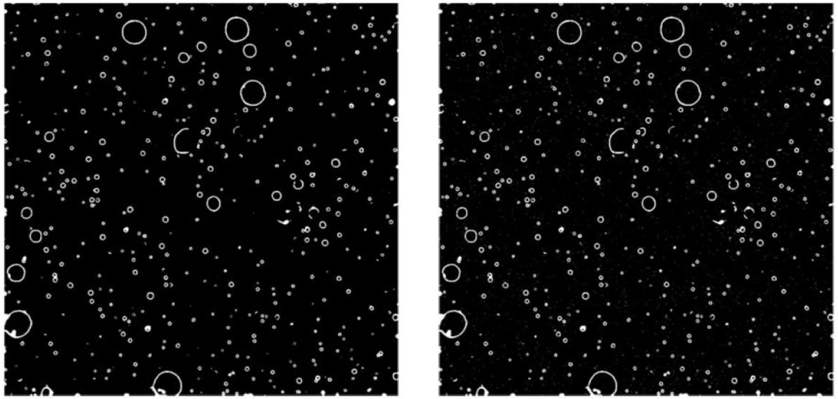
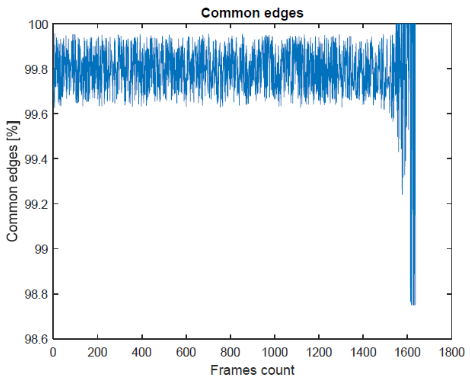
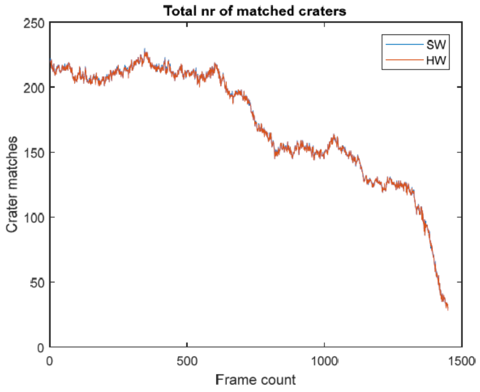


Figure 1: SW edges (left hand image), HW edges (right hand image)





Conclusions

- Crater detection and Crater matching have been implemented and validated by means of HW acceleration technique;

- The HW implementation of the Deep Learning Neural Network on FPGA was similar behavior with respect to the SW implementation;

- The HW implementation is done using fixed16, fixed point arithmetic, and the SW implementation have been developed on float32, floating point arithmetic;

- The HW implementation has subpixel accuracy on center and radius estimation in image frame;

- The AITAG scenario is under development

- The design of the deep learning implementation is done using the Processing Units which maximize the flexibility for future implementations.

Thank you

GMV:
Dragos Georgel Gogu
Florin Adrian Stancu
Antonio Pastor González
Daniel Fortún Sánchez
David Gonzalez-Arjona

Part of the work presented was carried under a program of, and funded by the European Space Agency

INNOVATING SOLUTIONS