Development methods and deployment of machine learning model inference for two Space Weather on-board analysis applications on several embedded systems

A cooperation from TEC-SWT and TEC-EDD

Hugo Marques[1], Kyra Foerster[2], Malte Bargholz[2], Maris Tali[2], Luis Mansilla[1], David Steenari[2]

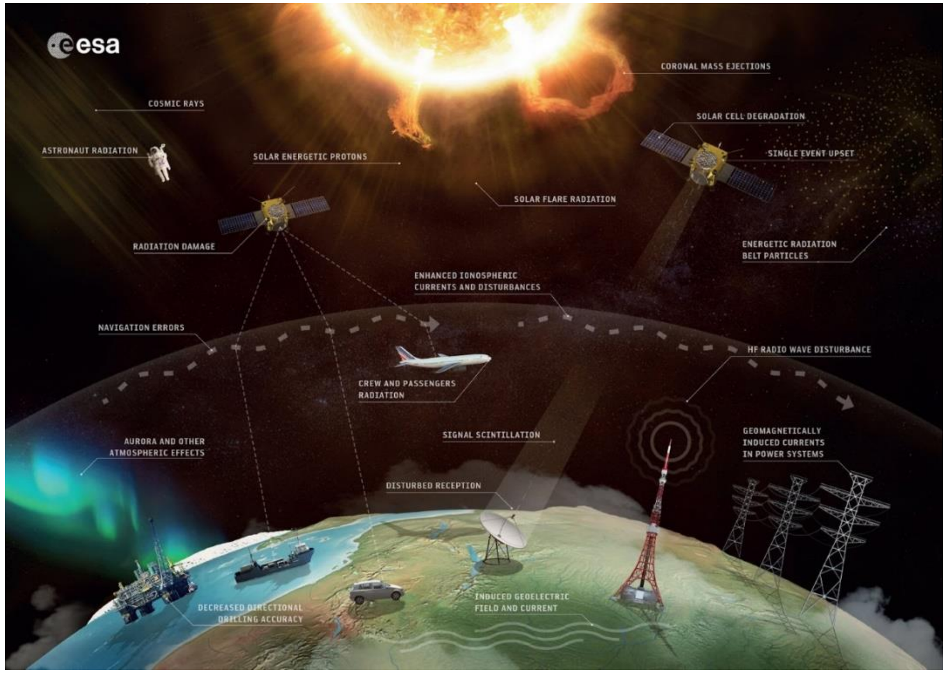
[1]: TEC-SWT, Software Technology Section
[2]: TEC-EDD, On-Board Computers and Data Handling Section

17/11/2021

Table of Content

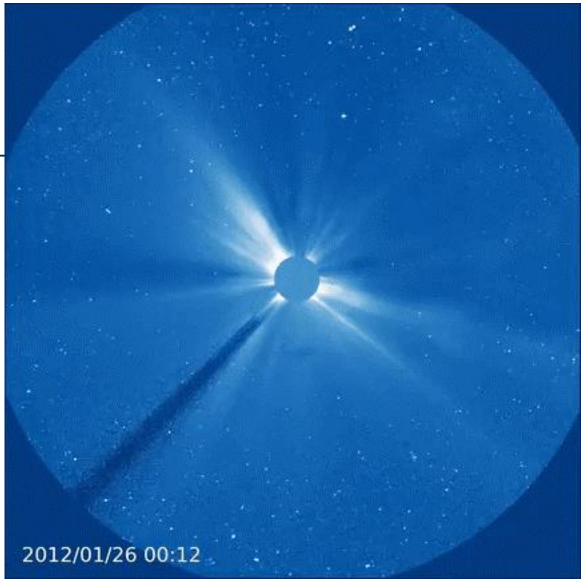
Space Weather On-Board Detection

- Background and motivation
  - Communication limitations may result in severe delays in the generation of alarms raised upon detection of critical events.
  - Image analysis tasks can be performed directly onboard, through the use of deep learning.
  - Alerts can be immediately communicated to the ground segment.
  - Internal activity to study the feasibility of deploying on-board analytics for Space Weather events detection.
- Possible interesting use case on the soon to be formally known as Lagrange Mission.

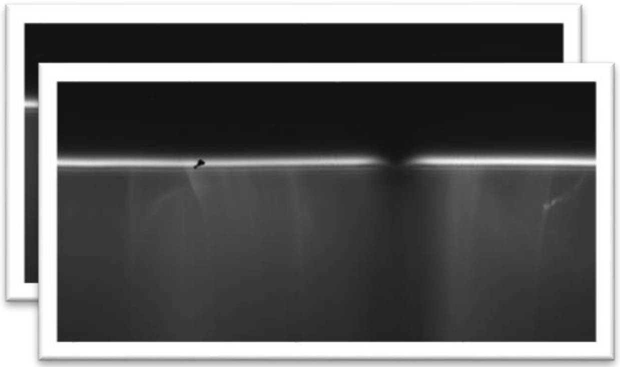


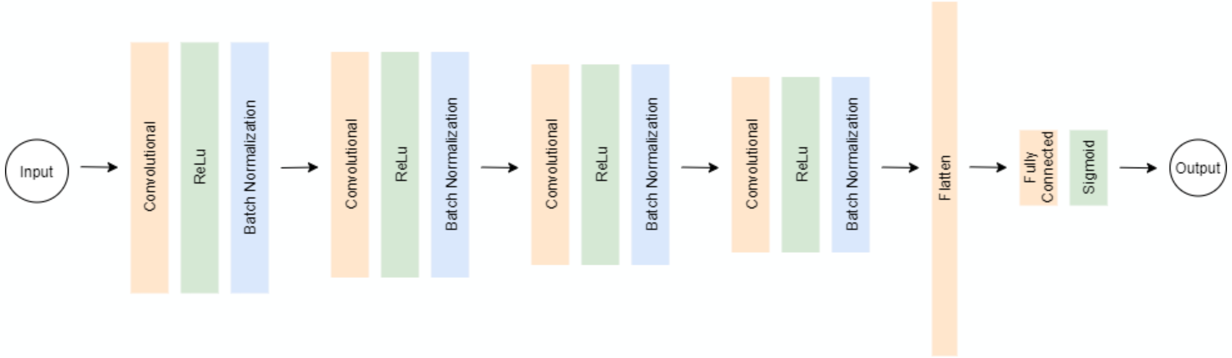
CNN Model Use Case and Architecture

- Initial research by Politecnico di Torino [1]
  - Dataset curated and first CNN model trained by Politecnico di Torino
- Internal R&D
  - Follow up on the initial CNN architecture
  - Models re-trained and updated to TensorFlow 2
- Potential application => on-board processing aiming at detecting Coronal Mass Ejections (CMEs).



CME captured by SOHO in 2012



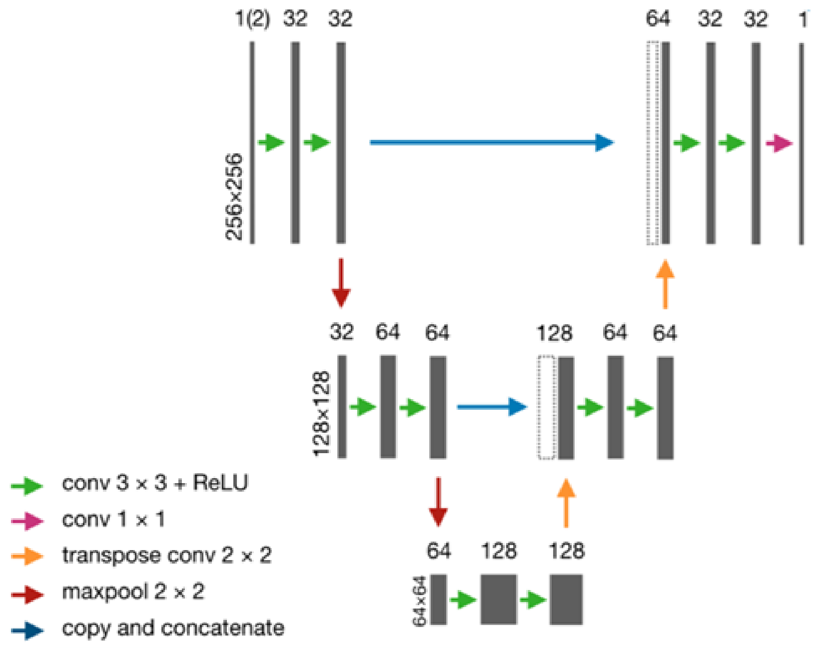


CME/ No CME

[1] D. Valsesia et al., "Detection of Solar Coronal Mass Ejections from Raw Images with Deep Convolutional Neural Networks," IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 2272-2275, doi: 10.1109/IGARSS39084.2020.9323169.

Particle Detection as an Image Segmentation Problem

- The neural network was trained to distinguish between background and radiation particles
- Each pixel is classified as radiation particle (0) or background (1) and saved in a mask



https://arxiv.org/pdf/1907.09500.pdf

Generate Training Data → Train Network → Inferencing → Evaluate Performance

- Training with Keras/TensorFlow 2
- Data augmentation by rotating/shifting of images

Previous Work: Hardware and Tool Survey

Tool	Developer	Type	Possible targets	DNN Frameworks	Non-DNN	Open-Source
XLA	Google	SW inference	Processors	TensorFlow (1.x, 2.x)	No	Yes
TensorFlow Lite	Google	SW inference	Processors	TensorFlow (1.x, 2.x)	No	Yes
Coral	Google	SW inference	Edge TPU	TensorFlow Lite	No	Yes
TFMin	Uni Surrey, Airbus	SW inference	LEON/SPARC	TensorFlow (>1.13.0)	No	Yes
TFLM	Google	SW inference	Cortex-M , ESP32	TensorFlow Lite for Microprocessors	No	Yes
ARM NN	ARM	SW inference	Cortex-A, Mali GPUs	TensorFlow Lite, ONNX	No	Yes
JetPack (TensorRT)	NVIDIA	SW inference	Jetson, CUDA GPUs	TensorFlow (1.x, 2.0), Caffe, ONNX	Yes	Partially
ROCm	AMD	SW inference	AMD SoC devices	TensorFlow (1.x, 2.x), Caffe2, PyTorch, ONNX, NNEF	Yes	Yes
OpenVINO	Intel	SW inference	Myriad devices	Caffe(2), MXNet 1.5.x, TensorFlow 1.15, 2.2.x, Kaldi, ONNX 1.7.0 (PyTorch, Keras, CNTK)	No	Yes
AccessCore (KaNN)	Kalray	SW inference	MPPA	TensorFlow, Caffe (ONNX to follow)	Yes	No
NOGAH	Ramon.Space	SW inference	RC64, RC256	Keras	Yes	Partially
TVM	Apache	See Devices	Processors, GPUs, FPGAs etc.	PyTorch (1.4, 1.7) , TensorFlow (1.x, 2.x), MXNet, ONNX, Keras, TF Lite 2.1.0, CoreML, DarkNet, Caffe2	Yes	Yes
Vitis	Xilinx	FPGA IP	Xilinx FPGAs	TensorFlow (1.15, 2.3), Caffe, PyTorch (1.2 - 1.4), Keras	Yes	Partially
FINN	Xilinx	HDL generator	Xilinx FPGAs	ONNX (Brevitas export)	No	Yes
hls4ml	Fast Machine Learning Lab	HDL generator	Xilinx FPGAs	Keras, TensorFlow, PyTorch, ONNX	Yes	Partially
VectorBlox	Microsemi	FPGA IP	Microsemi FPGAs	See OpenVINO frameworks	No	Partially
Core Deep Learning	ASIC Design Services	FPGA IP	Microsemi FPGAs	TensorFlow 1.14, Caffe	No	With license

TABLE VI
COMPARISON OF AVAILABLE TOOLS AND DNN FRAMEWORK COMPATIBILITY.

D. Steenari, K. Foerster, D. O'Callaghan, C. Hay, M. Cebecauer,M. Ireland, S. McBreen, M. Tali, and R. Camarero, "Survey of high-performance processors and FPGAs for on-board processing and machine learning applications," inOBDP2021, 2nd European Workshop on On-Board Data Processing. ESA/CNES/DLR, 2021.

Tested Hardware and Tools

Target	Component Class	Mission Target	Tool
Zynq-7000 SoC	COTS	NewSpace	Vitis AI (FPGA)
Zynq UltraScale+ MPSoC	COTS	NewSpace	Vitis AI (FPGA) TF Lite (ARM Cortex-A)
Kintex UltraScale FPGA	RT	Institutional (with radiation mitigation, and if software is qualified)	Vitis AI (FPGA)
Versal ACAP (AI Core)	COTS / RT*	NewSpace*	Vitis AI (AI Engine)
GR740/LEON 4	RHBD	Institutional	TF Lite Micro
Unibap iX5 CPU/GPU	COTS	NewSpace	TF Lite
Myriad X	COTS	NewSpace	OpenVINO

Used definitions:

COTS:
non-qualified components, may have been radiation tested

RT (rad tolerant):
Upscreened / space qualified COTS components with known radiation performance (an no SEL)

RHBD (rad-hard by design):
Space qualified, rad-hard components

*: Versal ACAP announced for release 2022. Can be suitable for institutional missions (with radiation mitigation) in the future.

Case 1) Vitis AI - Xilinx FPGAs

Xilinx Vitis AI is a development stack for AI inference on Xilinx FPGAs [2]

- FPGA IP core (DPU) & software stack running on Linux inside the SoC
- ML models are quantized and simplified by the Vitis-AI toolchain

Target	CME Detection Inference Time [s]	Particle Detection Inference Time [s]
Zynq-7000 SoC	0.093	10
Zynq UltraScale+ MPSoC	0.019	-
Kintex UltraScale FPGA (KU040)	7.692	-
Versal ACAP (AI Core)	0.006	0.5

- Space DPU: Constructing a Radiation-Tolerant, FPGA-based Platform for Deep Learning Acceleration on Space Payloads (OBDP 2021, [3])

[2] https://github.com/Xilinx/Vitis-AI
[3] https://az659834.vo.msecnd.net/eventsairwesteuprod/production-atpi-public/b337839b5ced47caa8880fcc03ac6aba

Case 2) TensorFlow Lite for Microcontrollers - GR740/LEON4

Tensorflow Lite Micro is designed to run machine learning models on microcontrollers with only a few kilobytes of memory

→ Small software runtime, almost no dependencies and static allocation → Qualification candidate

- Integer quantization and simplification of ML model with built-in Tensorflow Converter
- Ported to GR740/LEON4 without OS, also possible to integrate with RTEMS
  - Required patching of the Tensorflow Lite Micro runtime due to alignment and endianness

→ Parallelization possible with RTEMS + OpenMP (already implemented, testing to-be completed)

→ The processing time of an non-optimized implementation on a single core LEON4, is already **sufficient for on-board detection with better latency than detection on ground after downlinking.**

Target	CME Detection Inference Time [s]
GR740/LEON 4 (non-optimized, bare-metal, single core)	47.5

TensorFlow Lite - Zynq UltraScale+ MPSoC ZCU102 Arm A53

- TensorFlow Lite is an open source deep learning framework for on-device inference.
- Objective was to assess the performance of the deployment of the CNN model to the Arm Cortex A53 Quad-core CPU available in the Zynq UltraScale+ MPSoC ZCU102 board.
- Petalinux image provided by Xilinx was running on the board.
- The model inference was performed using the TensorFlow Lite runtime.

Target	CME Detection Inference Time [s]
ZCU102 ARM A53	0.11

Vitis AI

AI Inference on Xilinx FPGAs

Advantages

- Same model can be compiled for different Xilinx Hardware (COTS and RT)
- Supports Caffe, TensorFlow and PyTorch
- HW acceleration

Drawbacks

- Support for a limited subset of TensorFlow operations
- Specific to Xilinx Devices
- Level of parallelization depends on IP size - not every IP fits on each device
- Requires Linux
- Dependency on DPUs
- Performance is highly dependant on the memory bandwidth

TF Lite

Deep learning framework for on-device inference

Advantages

- Easy conversion from TensorFlow models
- Optimized for embedded devices
- Seamlessly deployment
- Compatibility with CPUs and GPUs

Drawbacks

- Support for a limited subset of TensorFlow operations
- Efficiency and optimization tradeoff is accuracy
- Requires Linux
- Dependency on TF Lite inference engine

TFLM

Deep learning framework for micro-controllers

Advantages

- No operating system needed - bare metal
- Also compatible with RTOS (RTEMS)
- Core runtime fits in just 16 KB (Arm Cortex M3)
- Potential candidate for qualification - Single library

Drawbacks

- Support for a limited subset of TensorFlow operations
- Support for a limited set of devices
- Low-level C++ API requiring manual memory management
- On device training is not supported

Unibap/SpaceCloud

- Unibaps iX5 heterogeneous computing module:
  - AMD SoC CPU/GPU with TFLite
  - Myriad X with OpenVINO
- Implemented as SpaceCloud app (Docker)
  - Python and C++ (required for GPU)
- Both apps fly as IOD on “Wild Ride” (D-Orbit) for on-board SEU detection in ML inference
  - In-flight inference result is compared to verification data → check for errors
  - No radiation upsets detected during execution during first in-flight testing, further analysis is on-going

Target	CME Detection Inference Time [s]	Particle Detection Inference Time [s]
Unibap iX5 CPU	~0.14	~50
Unibap iX5 GPU	~0.07	~100
Myriad X	~12.5	~33.33

Conclusion and Future Steps

Conclusions:

- Both ML based Space Weather application cases proven feasible to deploy in qualified space processors/FPGAs suitable for Institutional missions.
- Different approaches to meet different missions requirements
    - e.g. TFLite + COTS processor for NewSpace, TFLM + RHBD processor for Institutional missions
- Model design dependent on the final target/tool

Future steps:

- Further development of space targets:
    - GR740 (LEON4) – deployment is on-going internally. Bare metal achieved, RTEMS/OpenMP optimisation is on-going
    - Xilinx XQRKU060 - deployment is planned internally, achieved on KU040
- Possible other targets for future benchmarking: RT Versal AI Edge/Core; RT-PolarFire; Hisaor; HPDP, RC64
- Establishment of standard space benchmarks for ML inference (on-going within OBPMark (obpmark.org))

Problem - qualification of ML on-board:

- Space-qualified software for ML inference is missing: ESA internal discussion on target software framework.
- How to qualify neural networks? ECSS-E-HB-40-02A “Machine Learning Qualification for Space Applications Handbook” work on-going.

Thank you!

Development methods and deployment of machine learning model inference for two Space Weather on-board analysis applications on several embedded systems

Hugo Marques[1], Kyra Foerster[2], Malte Bargholz[2], Maris Tali[2], Luis Mansilla[1], David Steenari[2]

[1]: TEC-SWT, Software Technology Section

[2]: TEC-EDD, On-Board Computers and Data Handling Section

Contact:

Luis.Mansilla@esa.int (TEC-SWT, software)

David.Steenari@esa.int (TEC-EDD, hardware)

If you are interested in activities on these topics -- don't hesitate to get in contact!