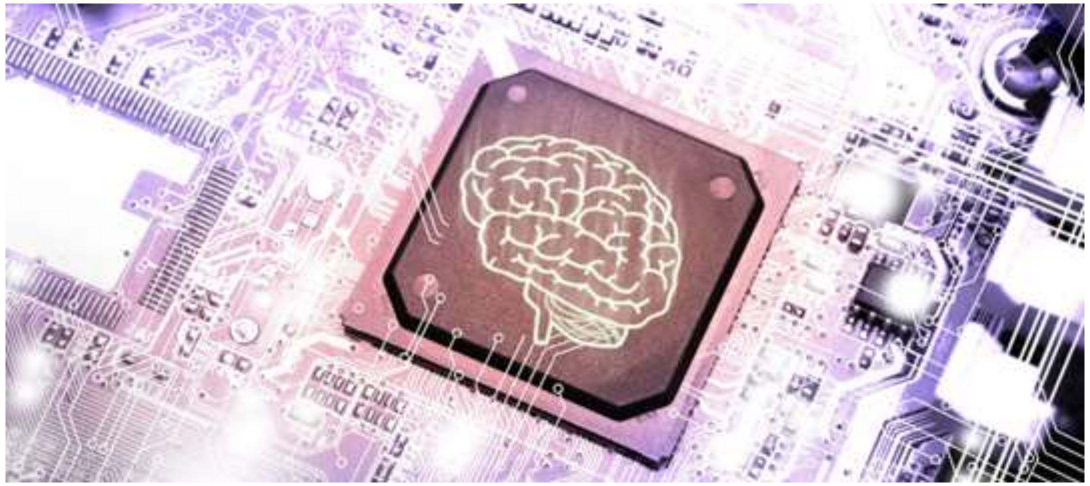


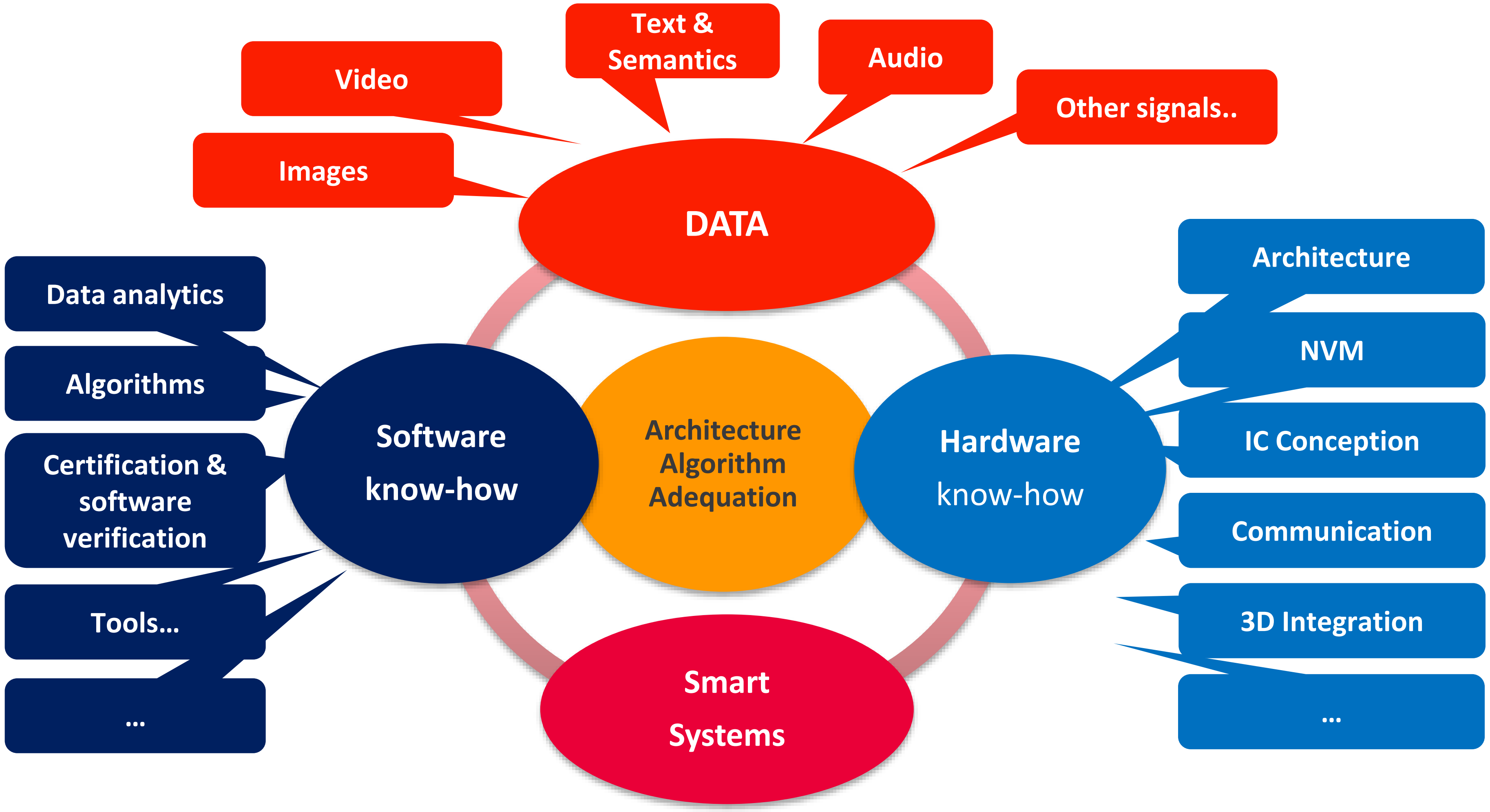
# CRESITT EVENT
# IA EMBARQUEE ET RECHERCHE AMONT

*CEA Presentation for CRESITT* | *October 17th, 2019*

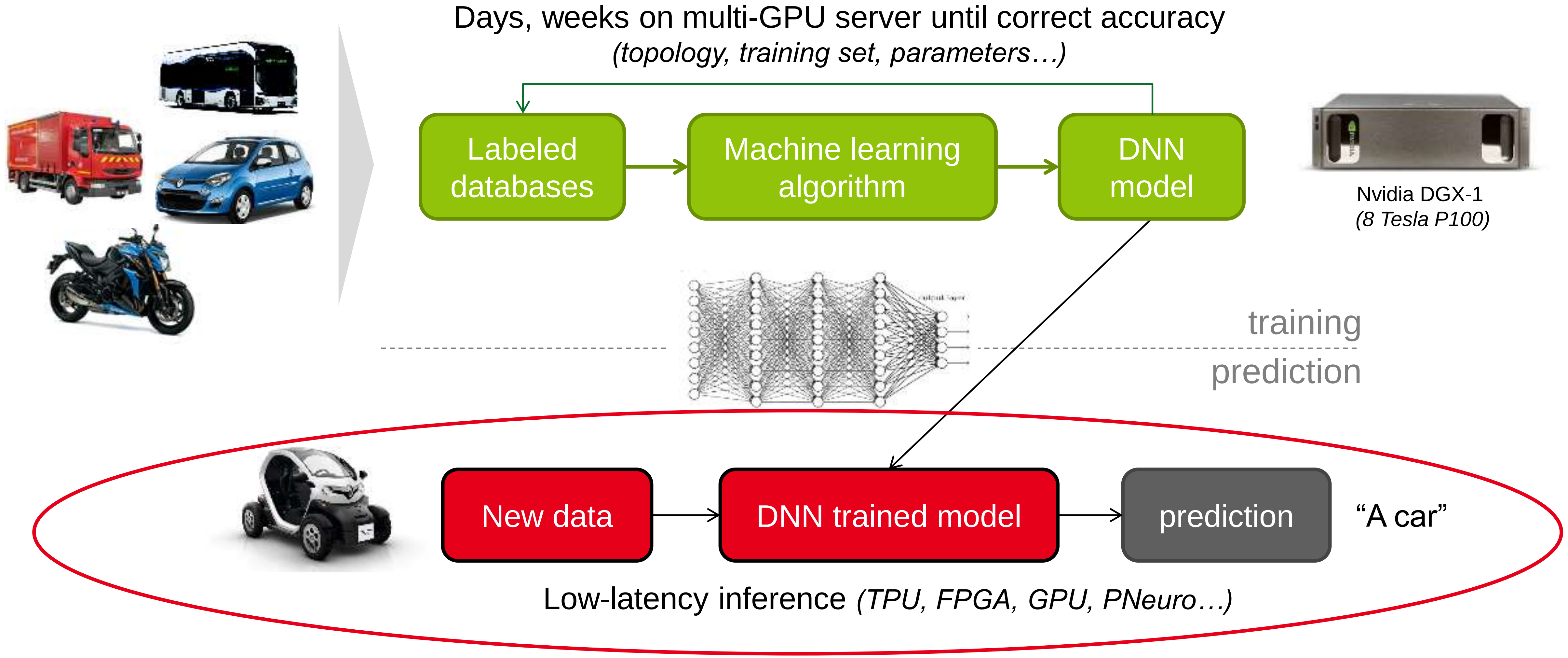
Sandrine Varenne, David Briand
**CEA LIST**
[sandrine.varenne@cea.fr](mailto:sandrine.varenne@cea.fr)

# IA EMBARQUÉE ET RECHERCHE AMONT

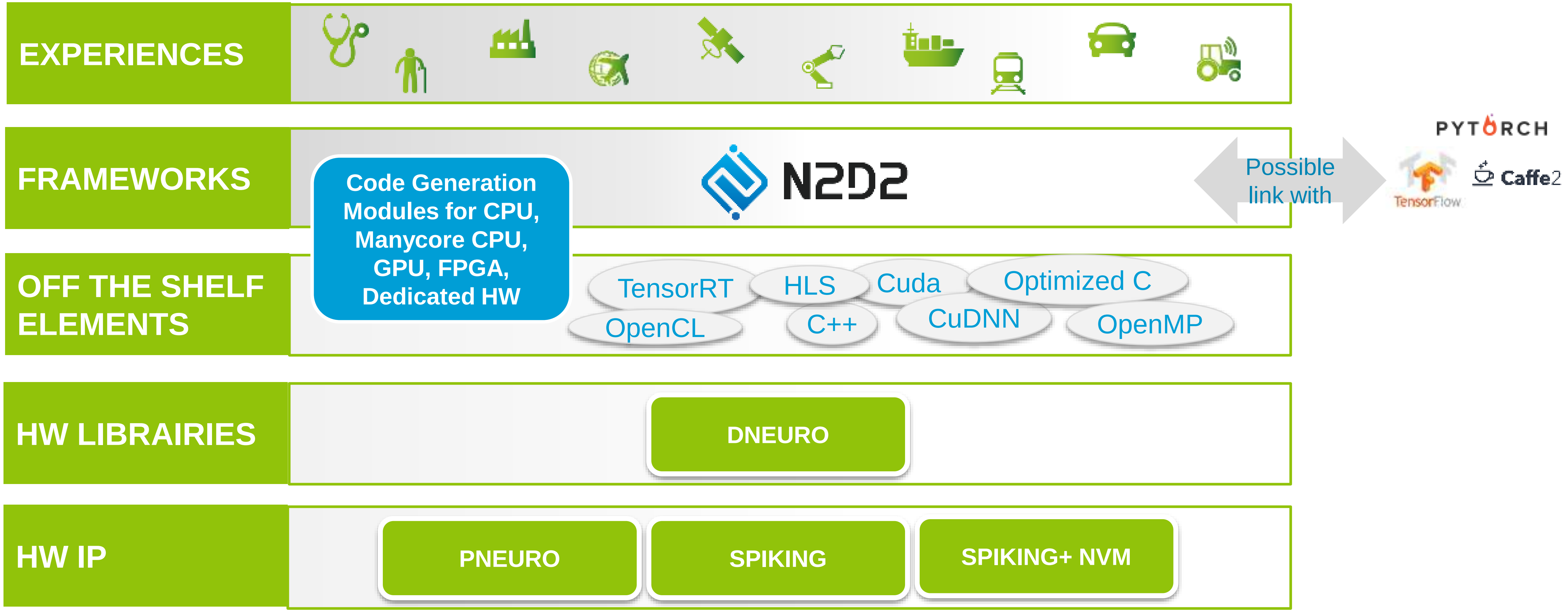
## CEA TECH & Artificial Intelligence



# CEA TECH & Artificial Intelligence
# To address the embedded Challenges


Days, weeks on multi-GPU server until correct accuracy
*(topology, training set, parameters…)*
Labeled databases
Machine learning algorithm
DNN model
Nvidia DGX-1
*(8 Tesla P100)*
training
prediction
New data
DNN trained model
prediction
"A car"
Low-latency inference *(TPU, FPGA, GPU, PNeuro…)*

# KNOW-HOW OF CEA IN DEEP LEARNING & EMBEDDED AI



# N2D2
# An European Platform to address Embedded Systems' Challenges

**N2D2 has been totally developed by CEA**

N2D2

| | |
|---|---|
| **Database Handling and Data Preprocessing Help**<br>• Data conditioning<br>• Semi automatic Data labelling | ✓ |
| **Standalone Code generation for**<br>• COTS* Components (CPU, GPU, FPGA)<br>• Specific Hardware Targets (ST, Kalray, Renesas…)<br>• NN Hardware Accelerators based on CEA IP<br>**>> Well adapted for embedded AI** | ✓ |
| **Decision help for the implementation phase**<br>• Hardware Cost & Form Factor<br>• Power Consumption<br>• Latency | ✓ |
| **Spike Coding** | ✓ |

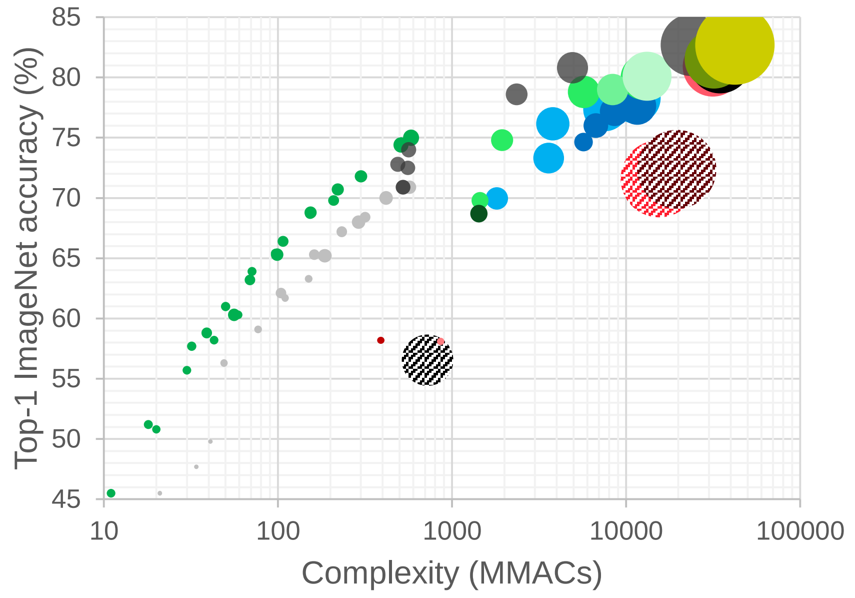
** COTS : Commercial Off-The-Shelf Components*

## Context / Motivations

- **Deep Neural Networks** (DNN) are very successful in the vast majority of classification/recognition benchmarks

  *…on high-end multi-250W GPU clusters*

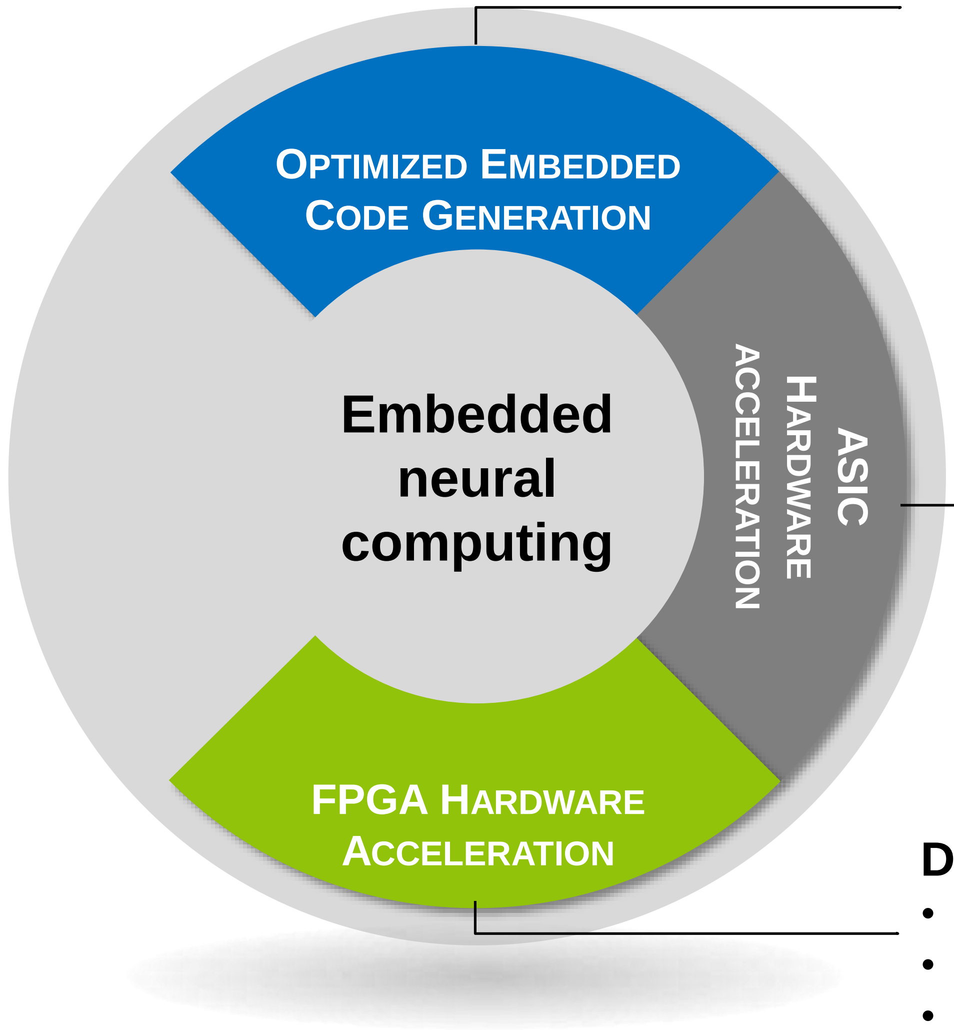
- Embedding low-power DNN remains challenging:
  - Must adapt and simplify DNN topologies
    - Reduce layers complexity (number of operations)
    - Reduce precision (8 bit integer or less)
  - Today's general purpose CoTS are inefficient for DNNs
    - Number of cores too low
    - Computing cores too complex (floating point computation)
    - Low MAC/cycle efficiency
    - Insufficient memory



➔ Balancing speed/power and applicative performances is a major challenge

➔ Need for a framework to automate DNN shrinking exploration and evaluation, performances projection and porting on embedded platforms

# Deep learning for embedded computing



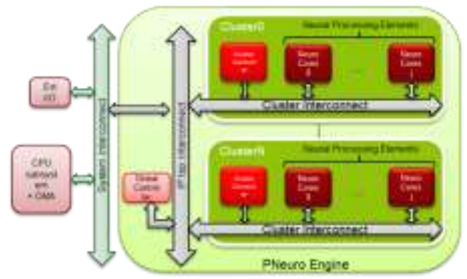
OPTIMIZED EMBEDDED CODE GENERATION
ASIC HARDWARE ACCELERATION
Embedded neural computing
FPGA HARDWARE ACCELERATION

## N2D2 : DNN design framework

- Unified modeling and NN exploration tool
- Custom applications building & optimization (CNN, Faster-RCNN…)
- Hardware mapping & benchmarking (CPUs, GPUs, FPGAs, ASIPs)
- N2D2 is available at https://github.com/CEA-LIST/N2D2/

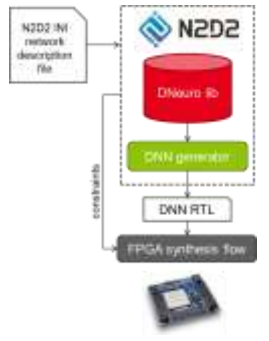
## Programmable processor PNeuro

- Clustered 8-bit SIMD architecture
- Designed for DNN processing chains and image processing
- Published at DATE 2018

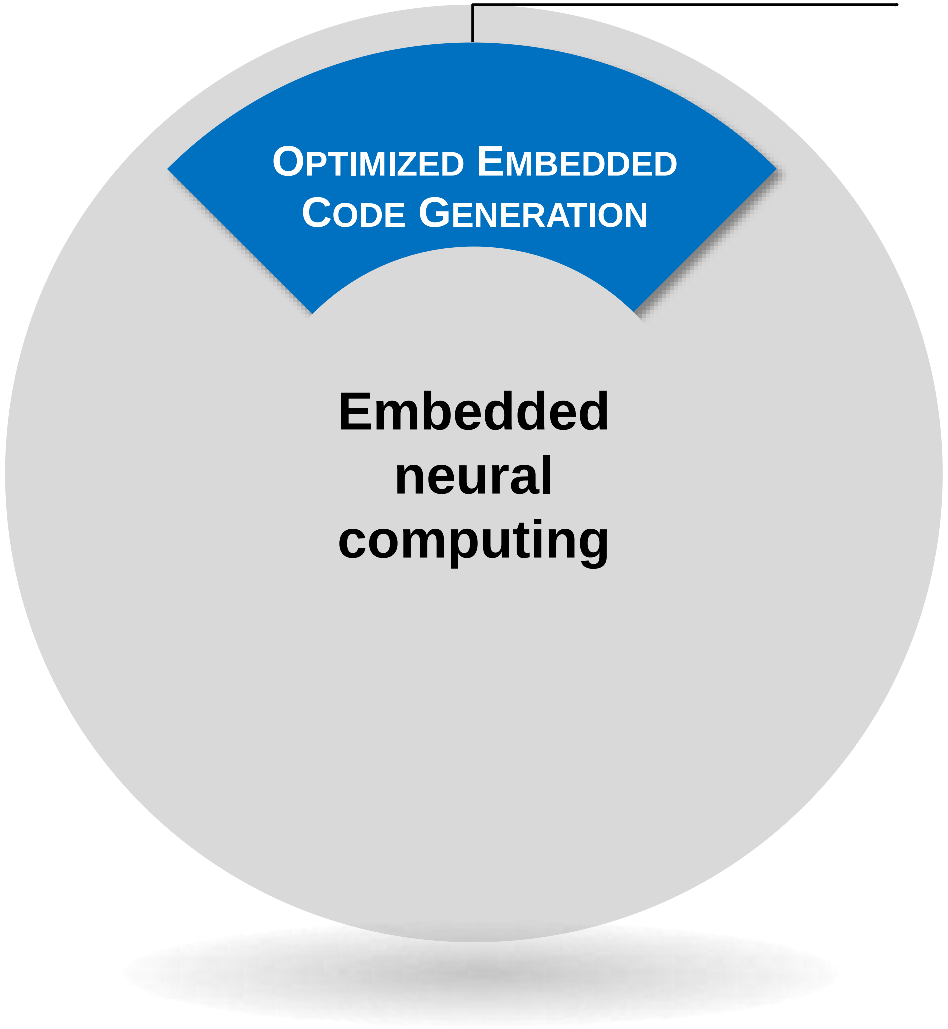


## Dataflow FPGA IP DNeuro

- Optimized RTL DNN layer kernels
- Automatic RTL generation through N2D2
- Dataflow computation, designed to use the DSP available on FPGA



# Deep learning for embedded computing



**N2D2 : DNN design framework**

- Unified modeling and NN exploration tool
- Custom applications building & optimization (CNN, Faster-RCNN…)
- Hardware mapping & benchmarking (CPUs, GPUs, FPGAs, ASIPs)
- N2D2 is available at https://github.com/CEA-LIST/N2D2/

**Motivations**

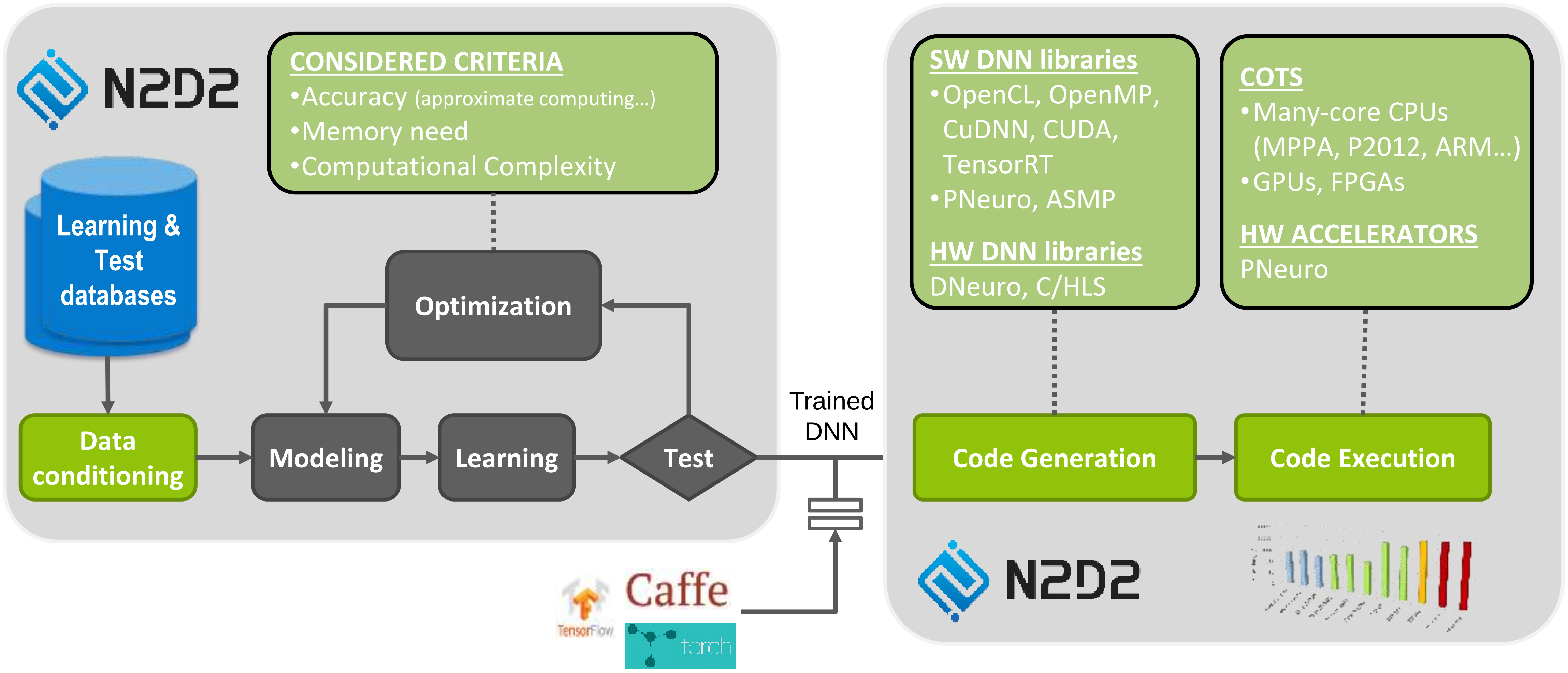
- Deep Neural Networks (DNN) are today extremely successful in the vast majority of classification/recognition benchmarks… on high-end multi-250W GPU clusters
- Embedding low-power DNN remains challenging:
  - Must adapt and simplify DNN topologies
    - Reduce layers complexity (number of operations)
    - Reduce precision (8 bit integer)

**➔ Balancing speed/power and applicative performances is a major challenge**

- **Need for a framework to automate DNN shrinking exploration and evaluation, performances projection and porting on embedded platforms**

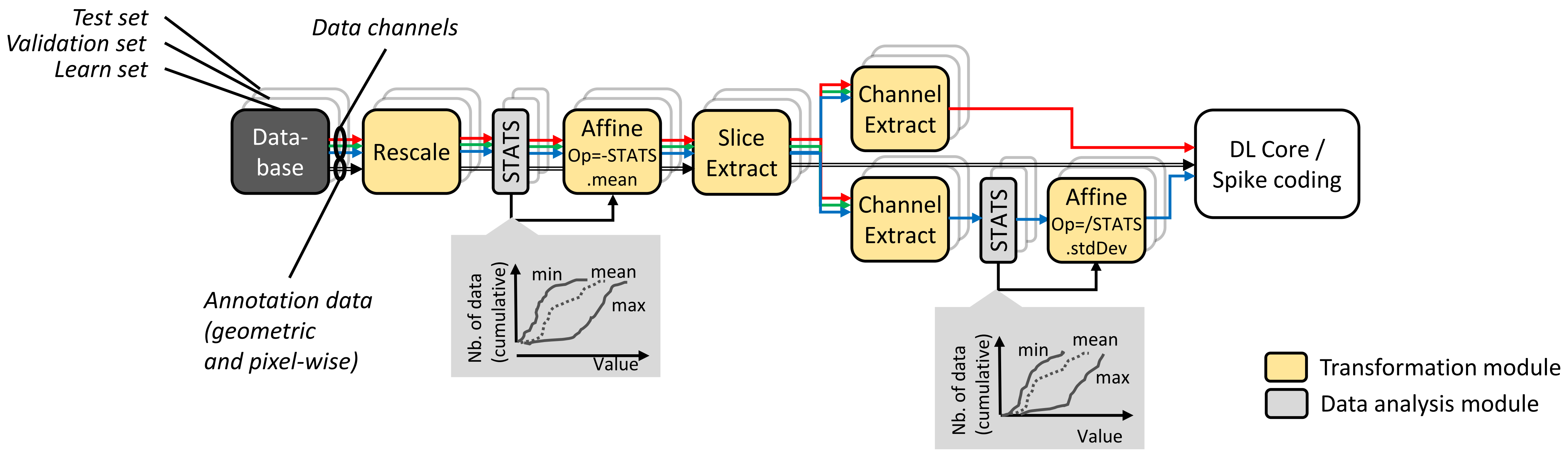
# N2D2: DNN Design Environment

- A unique platform for the design and exploration of DNN applications



# N2D2: Data Augmentation, Conditioning and Analysis

- N2D2 integrates data processing and analysis dataflow building
  - Genericity: process image and sound, 1D, 2D or 3D data
  - Associate a label for each data point, 1D or 2D labels
  - Support arbitrary label shapes (circular, rectangular, polygonal or pixel-wise defined)
  - Apply transformations to data, pixel-wise labels and geometrical labels
    - *Basic operations: rescaling, flipping, normalization, affine, filtering, DFT…*
    - *Advanced operations: elastic distortion, random slices/labels extraction, morphological reconstructions…*



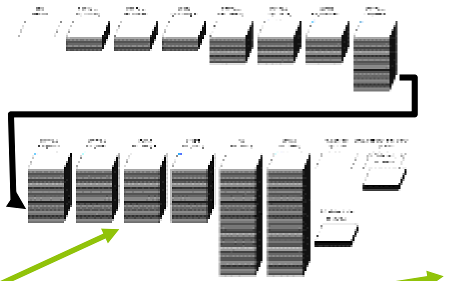
CONFIDENTIEL

# N2D2: Typical Outputs

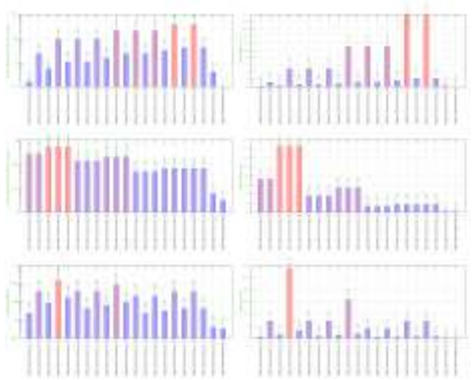
N2D2 INI network description file

```
; Database
[database]
Type=MNIST_IDX_Database
Validation=0.2

; Environment
[env]
SizeX=24
SizeY=24
BatchSize=128

[env.Transformation]
Type=PadCropTransformation
Width=[env]SizeX
Height=[env]SizeY

[env.OnTheFlyTransformation]
Type=DistortionTransformation
ApplyTo=LearnOnly
ElasticGaussianSize=21
ElasticSigma=6.0
ElasticScaling=36.0
Scaling=10.0
Rotation=10.0

; First layer (convolutionnal)
[conv1]
Input=env
Type=Conv
KernelWidth=5
KernelHeight=5
NbChannels=6
Stride=2
ConfigSection=common.config

; Second layer (convolutionnal)
[conv2]
Input=conv1
Type=Conv
KernelWidth=5
KernelHeight=5
NbChannels=12
Stride=2
ConfigSection=common.config

; Third layer (fully connected)
[fc1]
Input=conv2
Type=Fc
NbOutputs=100
ConfigSection=common.config

; Output layer (fully connected)
[fc2]
Input=fc1
Type=Fc
NbOutputs=10
ConfigSection=common.config

; Softmax layer
[soft]
Input=fc2
Type=Softmax
NbOutputs=10
WithLoss=1
ConfigSection=common.config

; Common solvers config
[common.config]
WeightsSolver.LearningRate=0.05
WeightsSolver.Decay=0.0005
Solvers.LearningRatePolicy=StepDecay
Solvers.LearningRateStepSize=[sp]_EpochSize
Solvers.LearningRateDecay=0.993
```

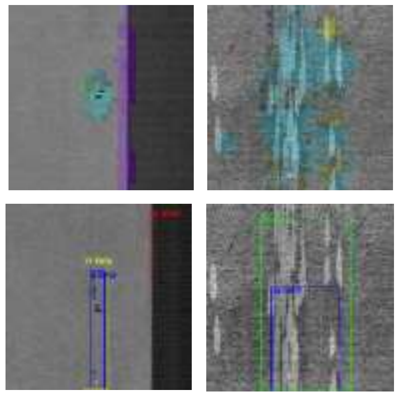
Dataflow visualization

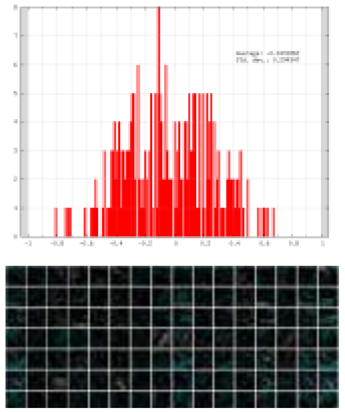


Layer-wise detailed memory and computing requirements



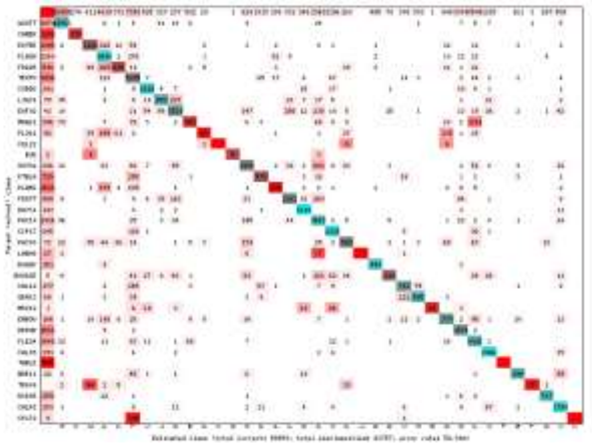
Results visualization:
- Pixel-wise segmentation
- ROI bounding box extraction and classification





Layer-wise weights and kernels visualization, distribution and data-range analysis

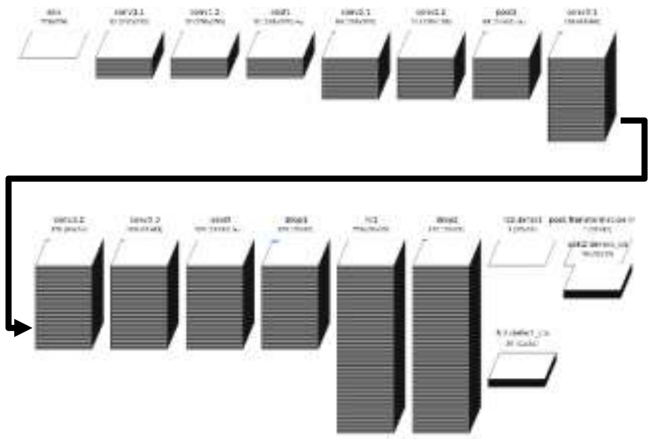
Layer-wise output visualization and data-range analysis



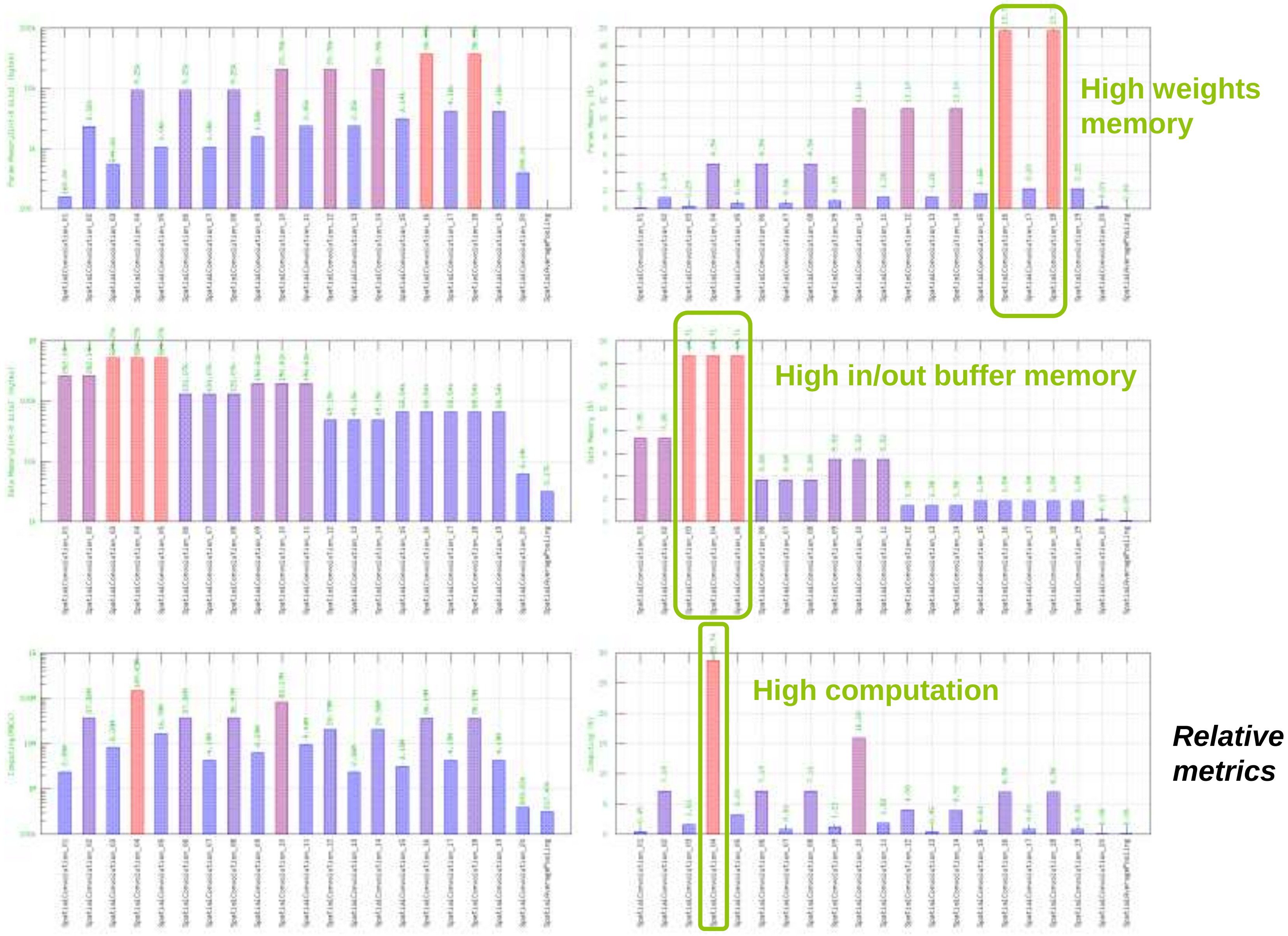
Pixel-wise and object wise confusion matrix reporting

CONFIDENTIEL

# N2D2: DNN Complexity Analysis



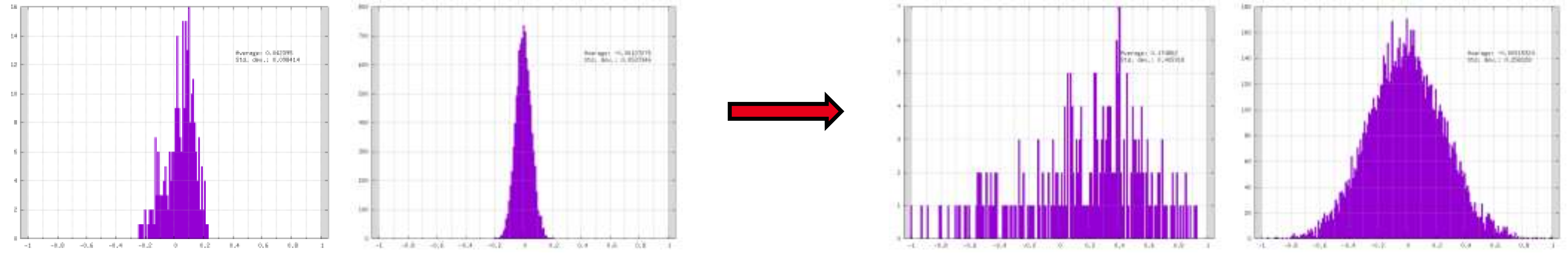
*Absolute metrics*



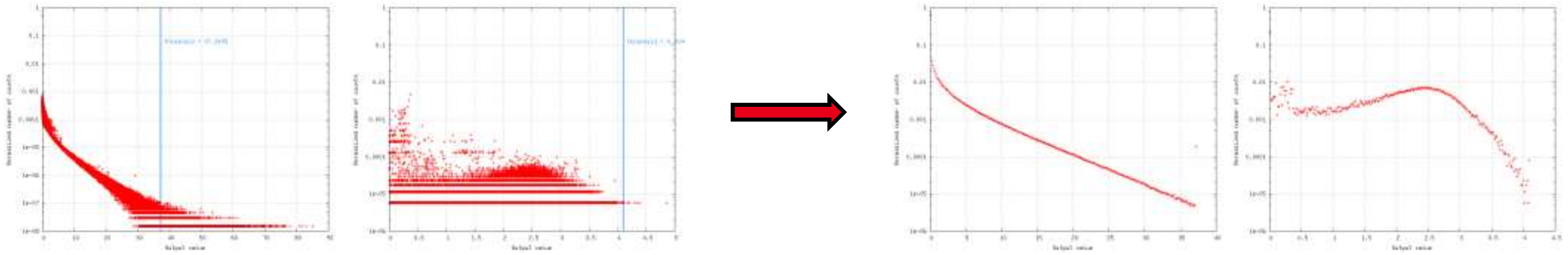
*Relative metrics*

# N2D2: Calibration for Integer Precision

- Weights clamping and/or normalization



- Layers output activation distribution quantization
  - Histogram analysis and optimal quantization threshold determination
  - Using Kullback–Leibler divergence



➔ Goal: automatic and guaranteed best result without retraining

# N2D2: Hardware Exports

N2D2 ➔ TensorRT on Drive PX2

**GPU** (NVidia)
C++/CUDA/CuDNN/
TensorRT

*Support SSD and Faster-RCNN*

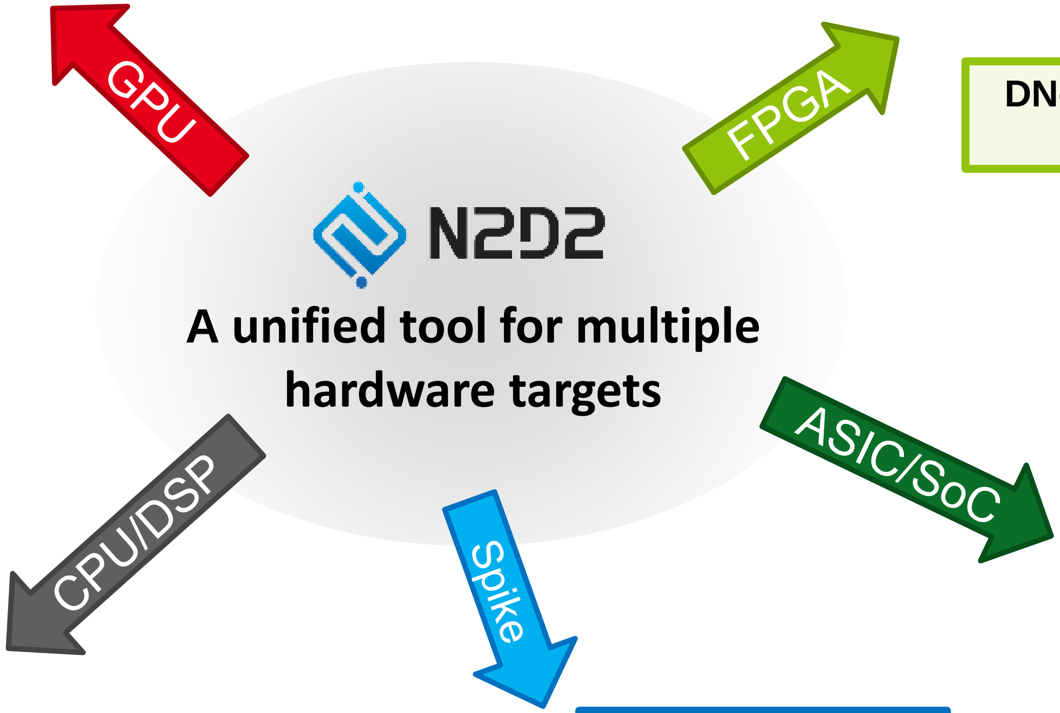
**GPU** generic
C++/OpenCL

HLS **FPGA** (Intel)
C++/OpenCL

HLS **FPGA** (Xilinx)
C/HLS

**DNeuro** ( )
RTL

*Dataflow configurable RTL library*



**MPPA** ( )
C++/OpenCL
KaNN API

**CPU** x86 / ARM / DSP
C/OpenMP
C++/OpenCL

**ASMP** ( )
C/OpenMP/CVA8

**PNeuro** ( )
RTL/ASM

*DSP-like programmable SIMD processor*

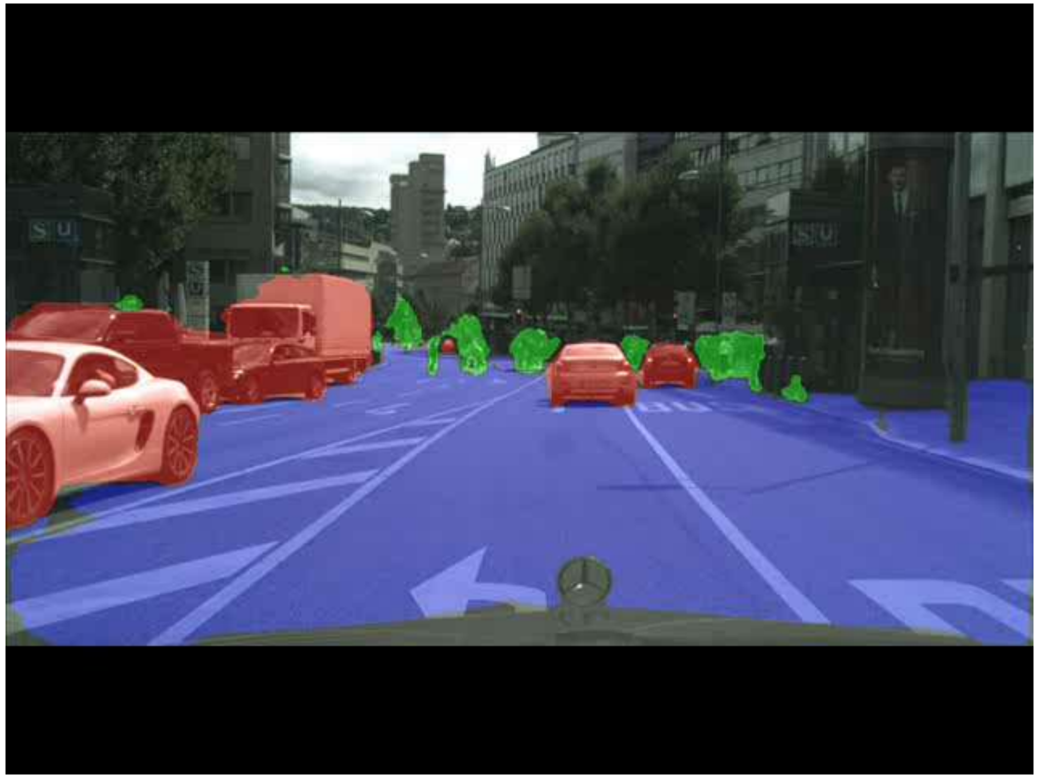
**R-Car** ( )
CNN-IP C API

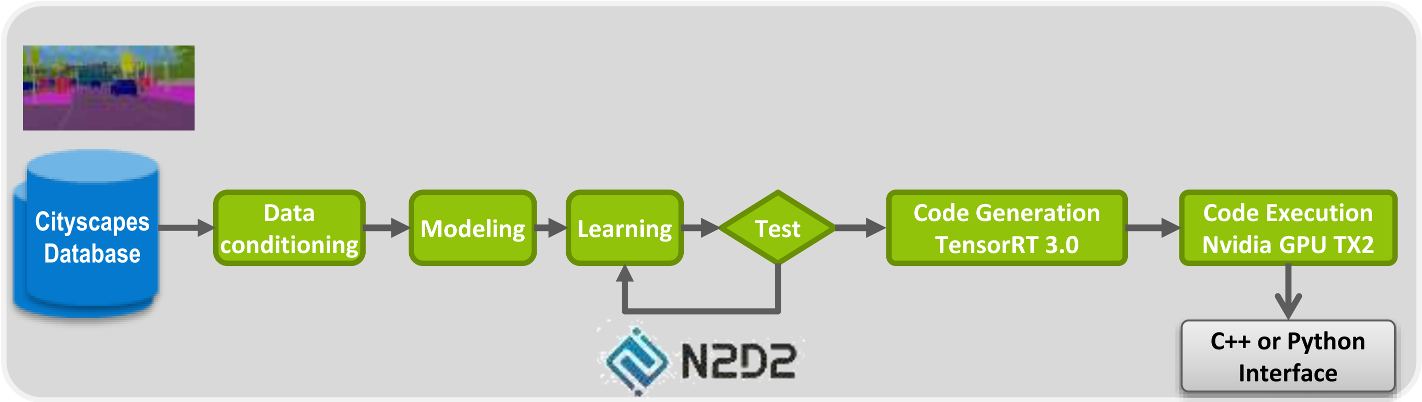
**NeuroSpike** ( )
RTL

Generic spike
SystemC

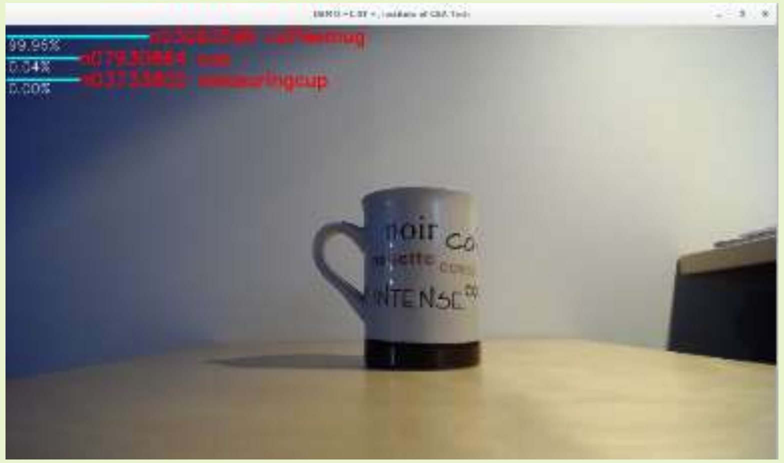
*Generic / not optimized for a specific product*

## N2D2: DNN Design Environment





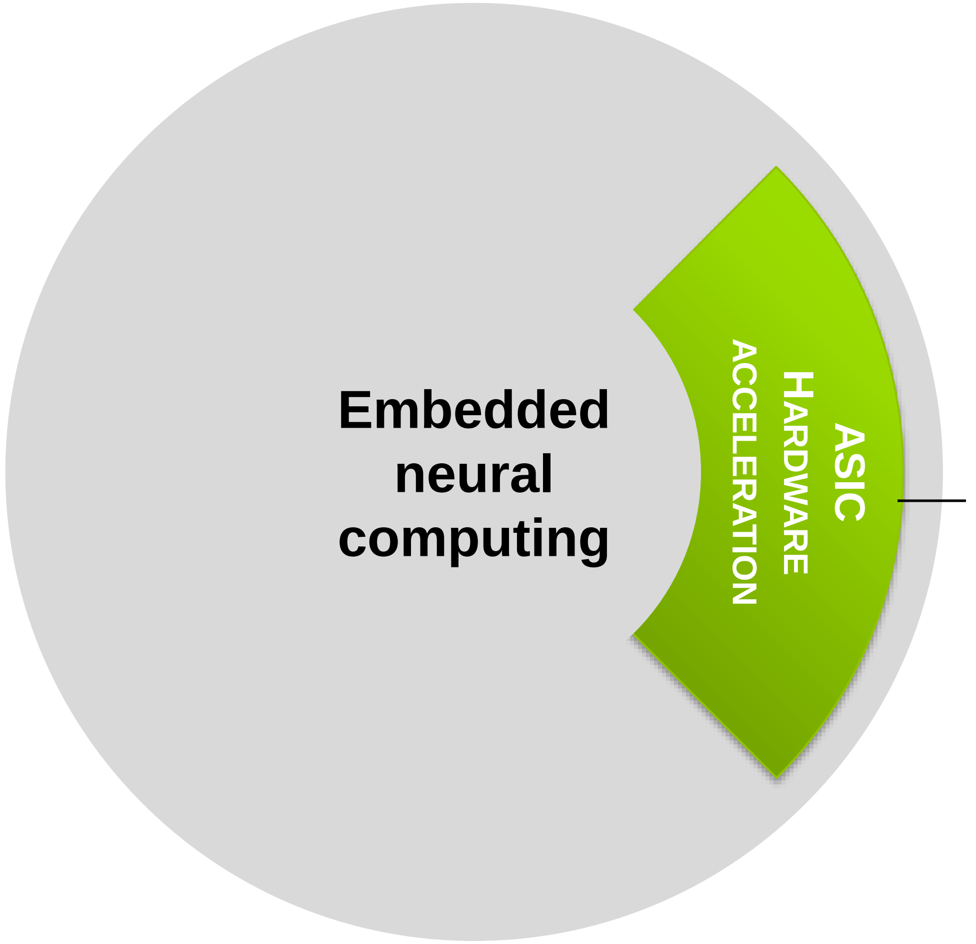
# Try N2D2 NOW!

| AppObjectRecognition/ | AppFaceDetection/ | AppRoadDetection/ |
|---|---|---|
| **Live object recognition application** *based on ILSVRC2012 (ImageNet) dataset* | **Live face detection application, with gender recognition** *based on the IMDB-WIKI dataset* | **Simple road segmentation application** *based on the KITTI Road dataset* |
|  | | |

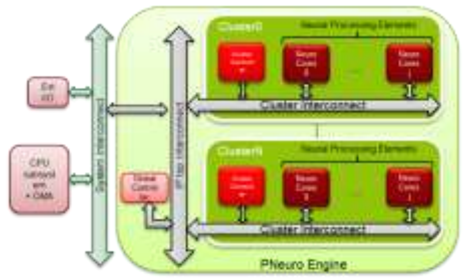
## N2D2 is available at [https://github.com/CEA-LIST/N2D2/](https://github.com/CEA-LIST/N2D2/)

- Smallest dependencies and requirements among major frameworks:
  Min requirements: GCC 4.4 or Visual Studio 12 / OpenCV 2.0.0
- Easily extendable with a “plug-and-play” modular system for user-made modules

# Deep learning for embedded computing

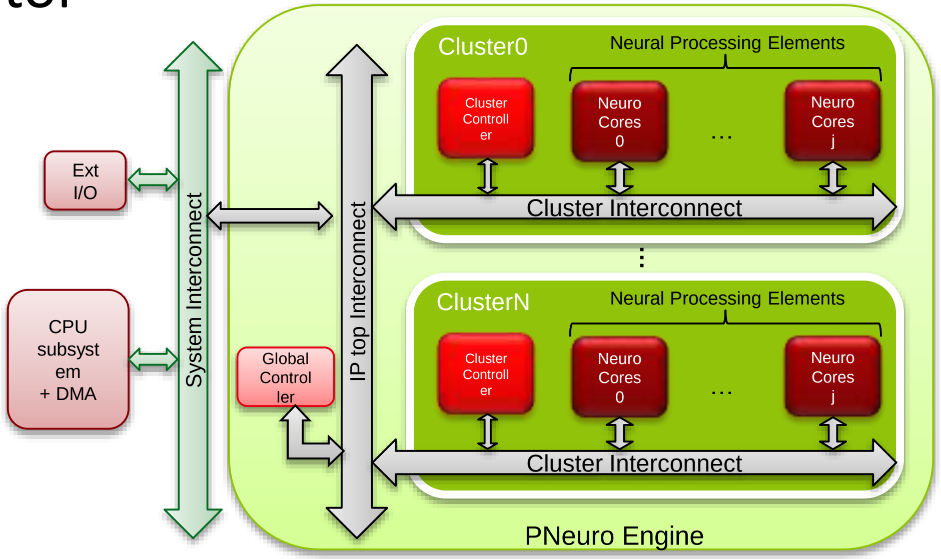


**Programmable processor PNeuro**
- Clustered 8-bit SIMD architecture
- Designed for DNN processing chains and image processing
- Published at DATE 2018



# PNeuro: a Neural DSP Processor

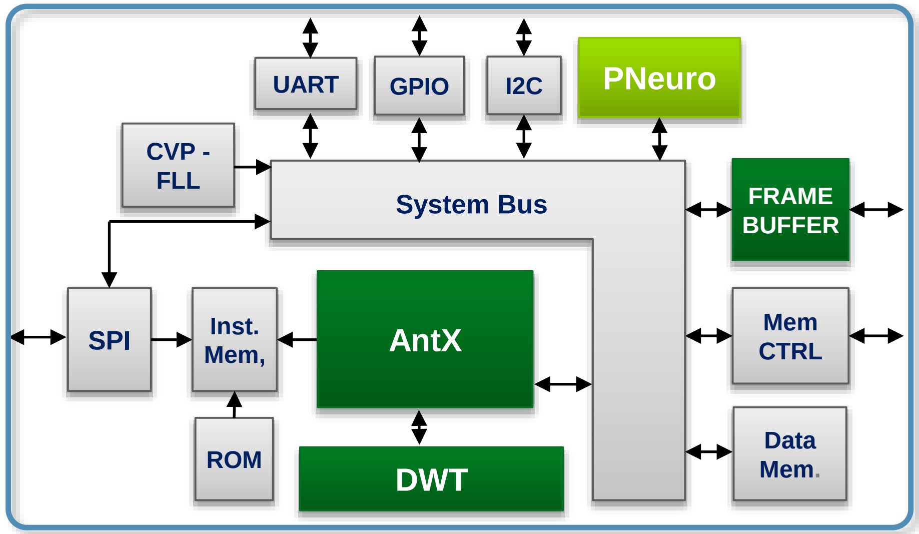
- Fully-programmable energy efficient hardware accelerator
  - N2D2 full-development flow
    - Full DNN framework for optimized embedded computing
  - Designed for DNN processing chains
    - Pre/post-processing phases
    - CNN, HMax, RNN (under development)
  - Supporting traditional image processing operations
    - Filtering, etc.

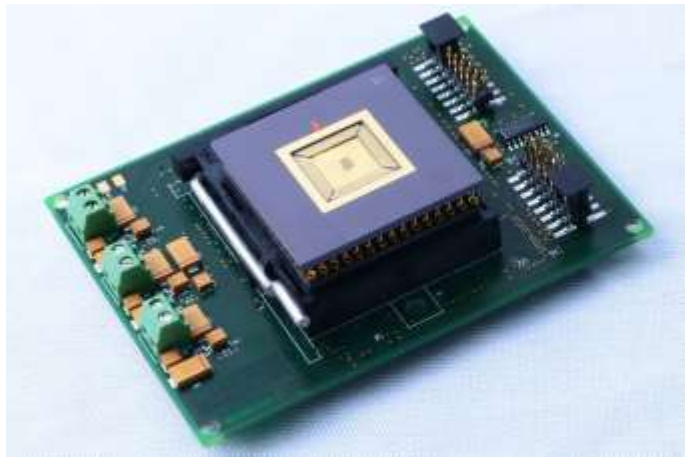


- Clustered SIMD architecture
  - Optimized operators for MAC & Non-Linearity approximation
  - Optimized memory accesses to perform efficient data transfers to operators
  - ISA including ~50 instructions (control + computing)

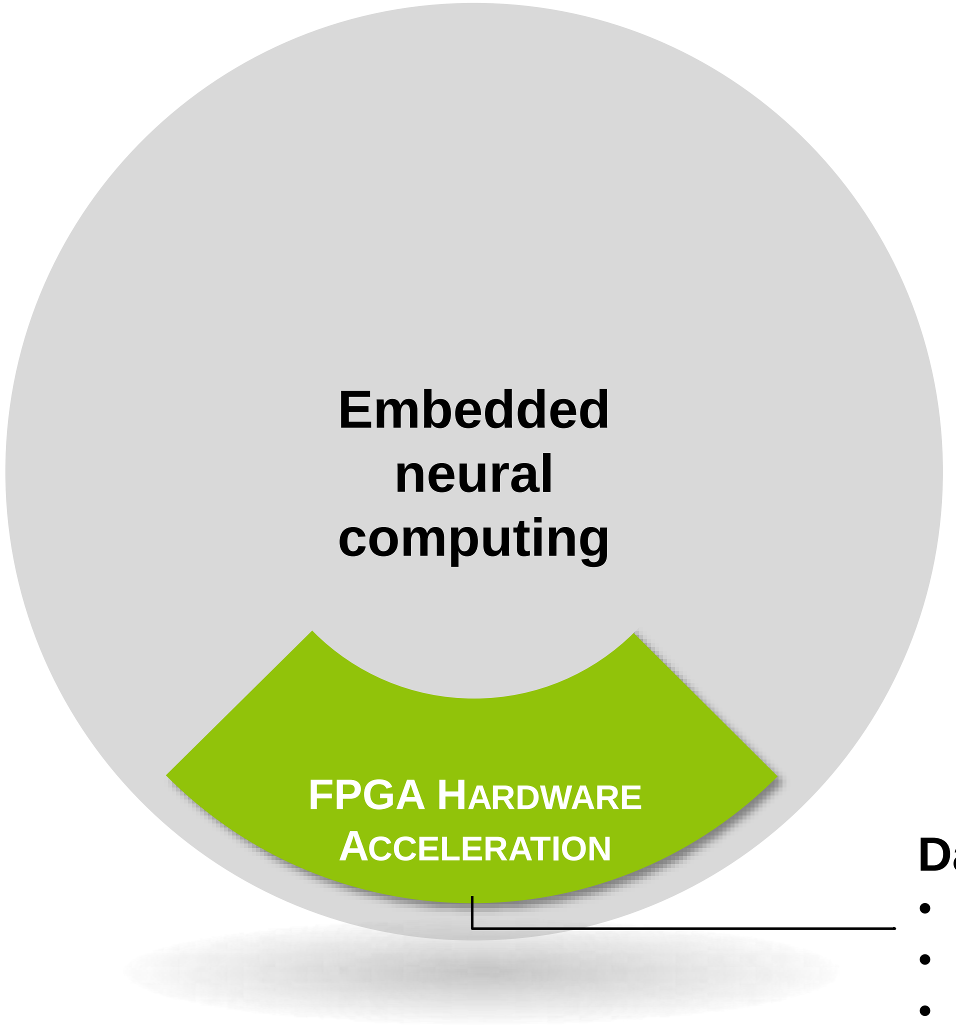
# PNeuro: FDSOI 28nm Test-chip

- Low-power programmable solution for smart IoT
  - PNeuro neural computing IP (1 cluster)
- 28 FDSOI CMP via Silicon Impulse
  - 0.126 mm$^2$ for a single PNeuro cluster (8 PE) and its control
  - PNeuro performance : 571 GMACs/s/W
- PNeuro roadmap
  - Integration in the N2D2 toolflow (ongoing)
  - Integration of debug in the architecture (ongoing)
  - 2 started Ph.D. related:
    - Online (onchip) unsupervised specialization of DNN
    - Automatic generation for optimal scalability and energy efficiency



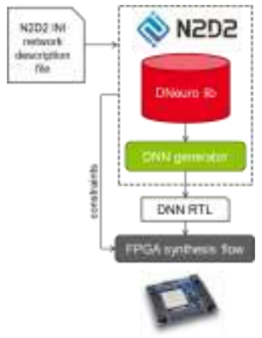


# Deep learning for embedded computing



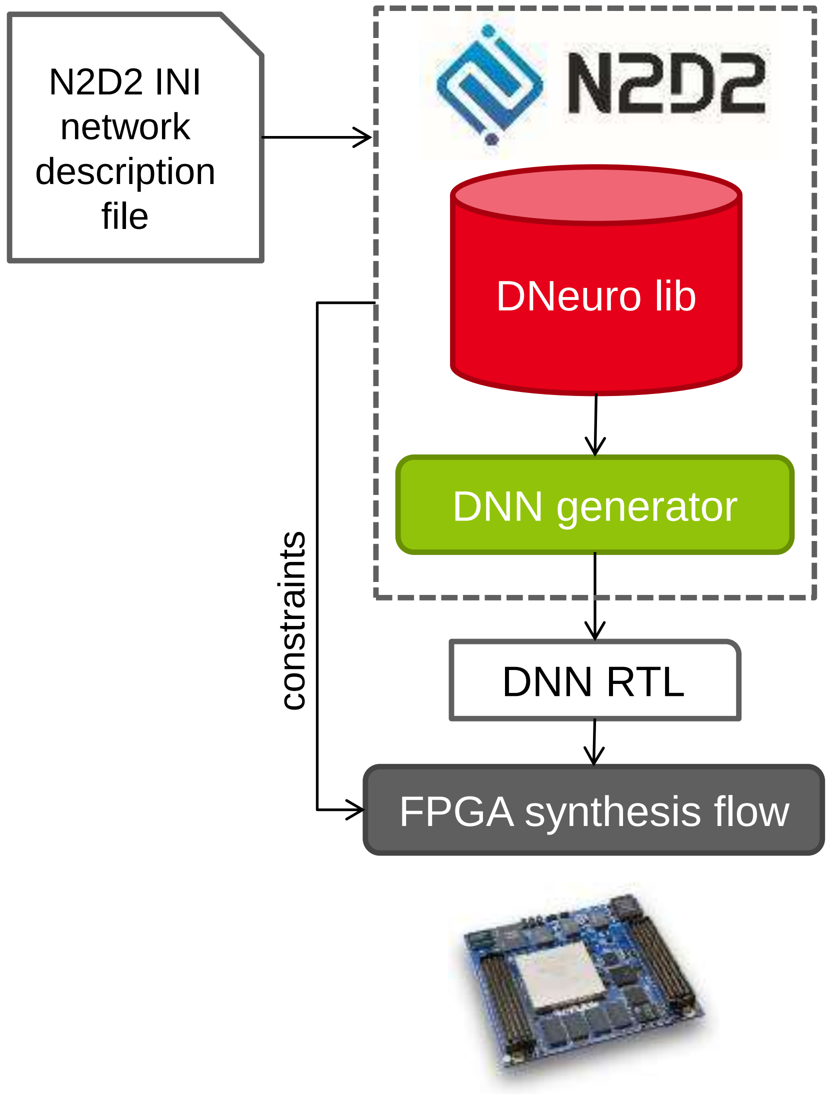
**Dataflow FPGA IP DNeuro**

- Optimized RTL DNN layer kernels
- Automatic RTL generation through N2D2
- Dataflow computation, designed to use the DSP available on FPGA



# DNeuro: RTL HW Library

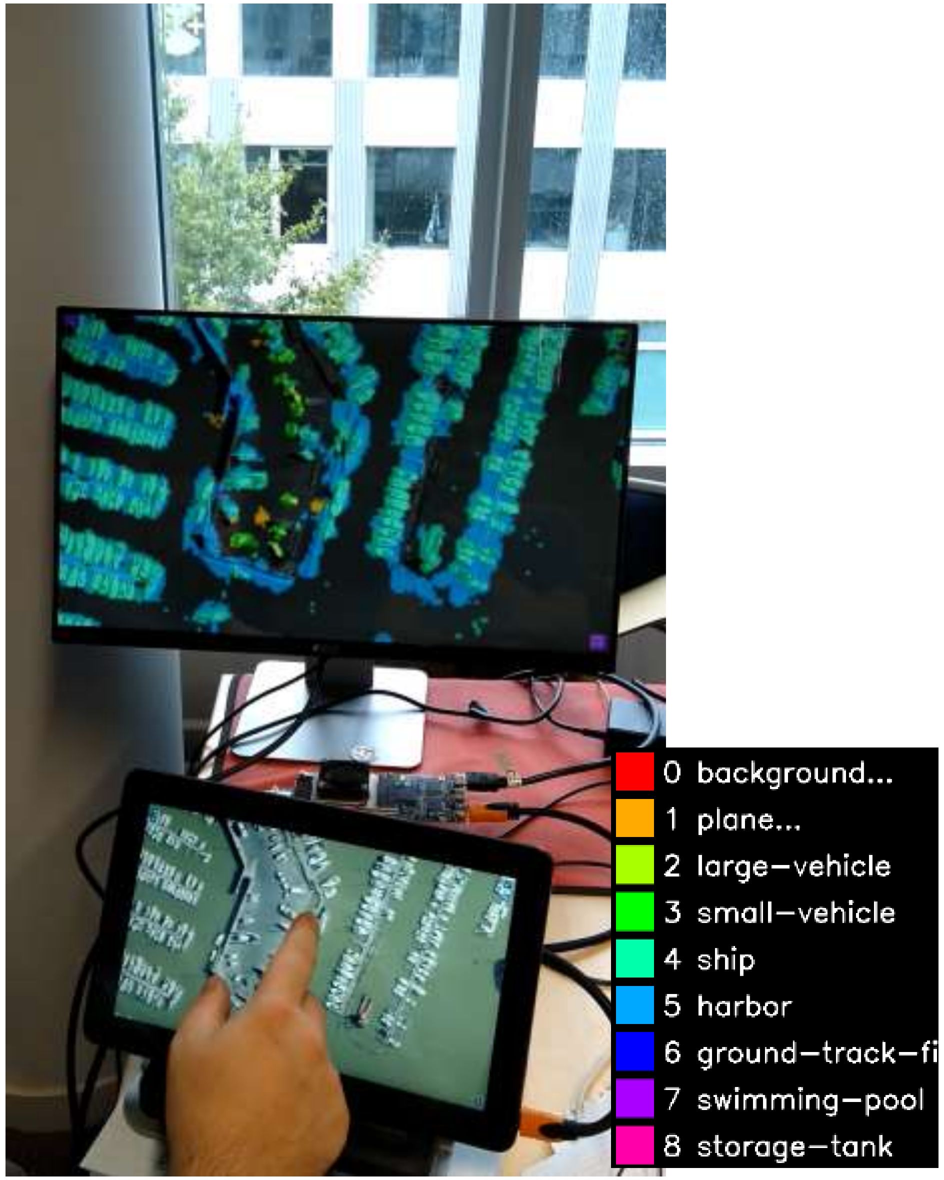
- DNeuro, RTL HW library for FPGA
  - Complete and independent RTL IP for DNN integration on FPGA
  - Dataflow computation, designed to use the DSP available on FPGA
  - Generated in a few steps from the DNN description and weights
- Main features
  - Data flow architecture requiring few memory (potentially **no** DDR)
  - Very high use rate of the DSP per cycle (**> 90%**)
  - Configurable precision (integers from 4 to 16 bits, typically **8 bits**)
  - Up to 4 MAC/DSP operations per cycle
- Low complexity IP, optimized for Intel and Xilinx FPGA
- Support convolutional layers (Fully-CNN)
  - Convolution and max pooling layers
  - Unit map connectivity and stride support
  - Future work: objects detector layers support (SSD, Yolo, Faster-RCNN…)



# Ex 1 : AI for Real Time Image Segmentation in constraint environment Avionics

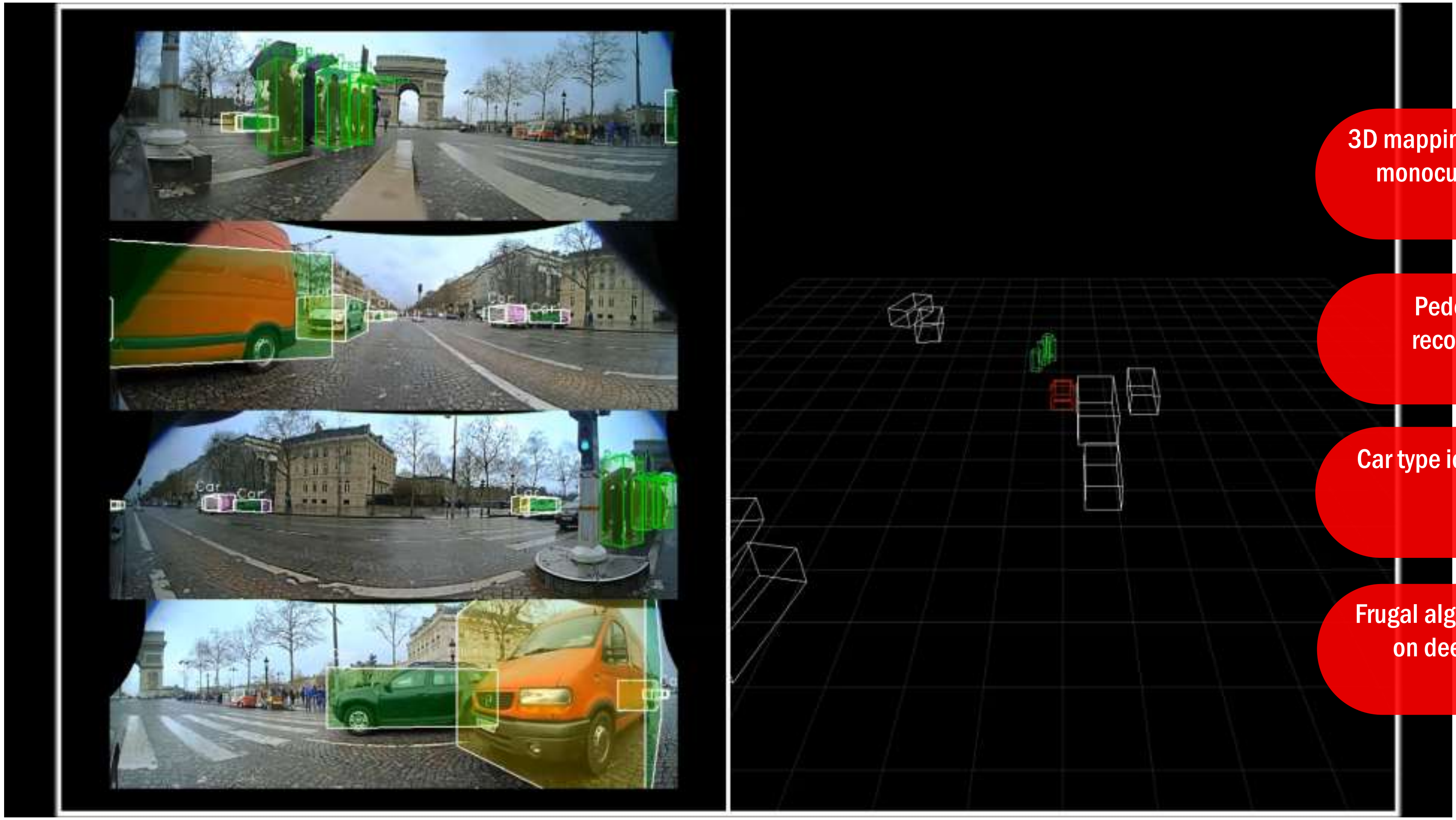
- DOTA dataset segmentation with MobileNet-based DNN
  - Automated DNeuro IP RTL generation from the DNN description and weights
  - Achieves ~160 FPS on Arria 10 SX270 for 640x480 images @ 200 MHz (w/o external DDR) ➔ **300 GOPS**

Arria FPGA•SoC
Stratix FPGA•SoC
KINTEX UltraSCALE
VIRTEX UltraSCALE



| | |
|---|---|
| 🟥 | 0 background... |
| 🟧 | 1 plane... |
| 🟩 | 2 large–vehicle |
| 🟩 | 3 small–vehicle |
| 🟩 | 4 ship |
| 🟦 | 5 harbor |
| 🟦 | 6 ground–track–fi |
| 🟪 | 7 swimming–pool |
| 🟪 | 8 storage–tank |

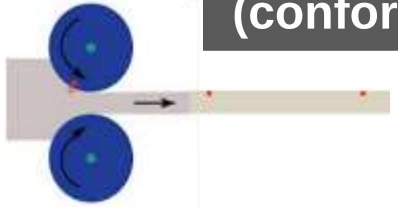
# Ex 2 : AI for Real Time Environment Perception Transport



- 3D mapping with a single monocular camera
- Pedestrian recognition
- Car type identification
- Frugal algorithms based on deep learning

# Ex 3 : AI for Real Time Quality Control Manufacturing

**Part Inspection (conformity, defects..)**



## CONSTRAINTS

- Real time with very high throughput (20m/s)
- Tiny defect (~mm) with low contrast
- Complex environment (oil vapor, few room for inspection..)

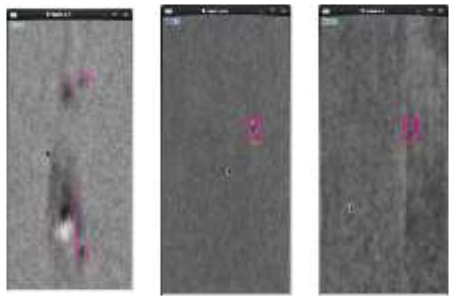
## SOLUTION

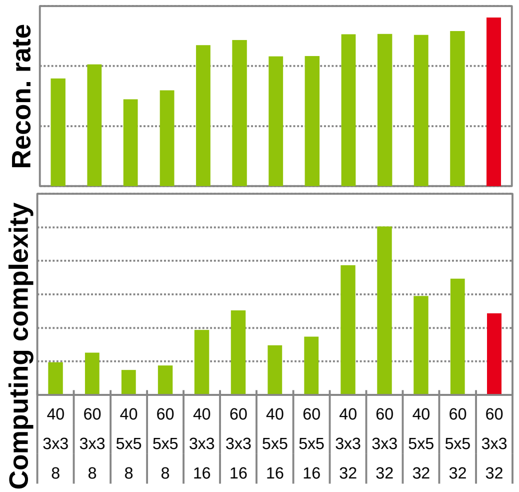
- Database labelling and Processing
- Fast NN topology Exploration
- Performance vs complexity analysis

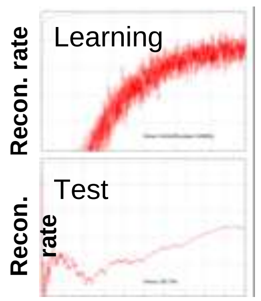
***1) Defects labeling and visualization***

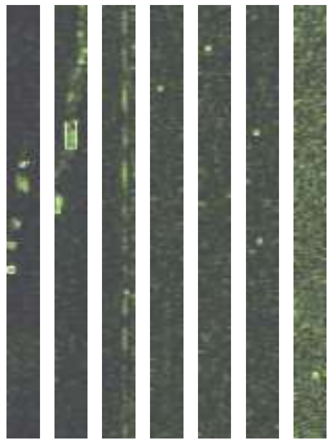
***2) NN Exploration and benchmarking***

***3) Defects identifications after NN learning***









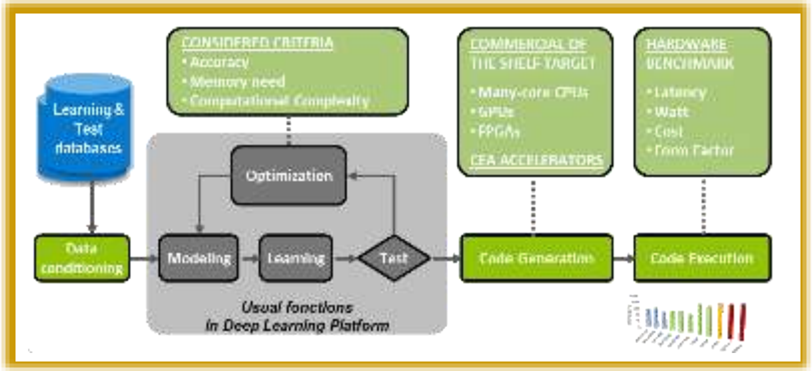
# Advance your Deep Learning Roadmap

***Use Cases***

Security, Defense
Manufacturing
Transport
Marketing
Automation…

***Software frameworks***

N2D2 deep learning framework
N2D2 HW exports
Benchmarking



**list & leti**

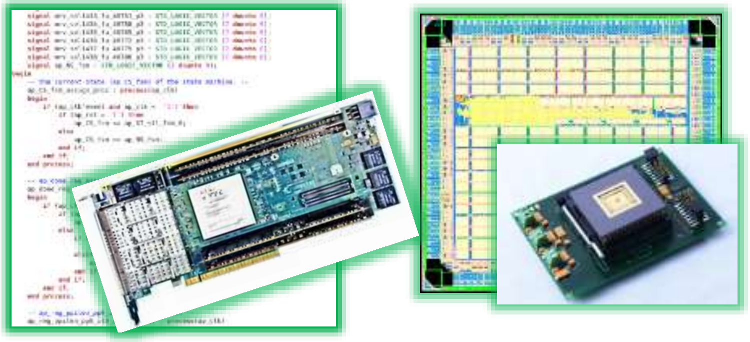
*Neuro computing platform*

N2D2

**From the algorithm to its integration**

***Hardware architectures***

PNeuro
DNeuro
HLS

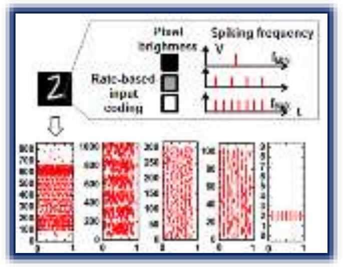


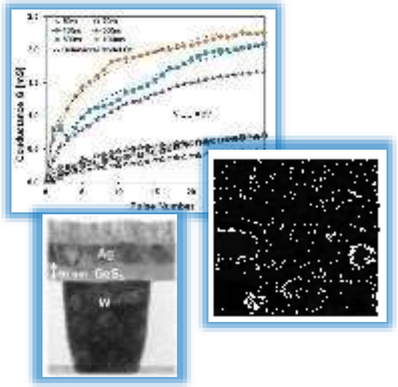
***Deep learning research***

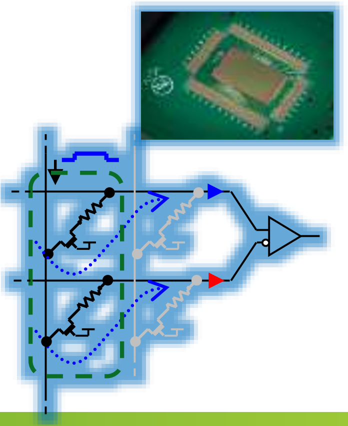
Event-based N2D2
Spike coding
Bio-inspired sensors
Unsupervised learning

***Advanced implementations***

RRAM synapses
3D stacking
Mixed A/D design
FDSOI 28nm







## Thank you

**Neural Networks Design and Deployment for Constrained Embedded Systems with N2D2 Framework**

Sandrine Varenne (sandrine.varenne@cea.fr)
David Briand (david.briand@cea.fr)

Centre de Saclay
Nano-Innov PC 172 - 91191 Gif sur Yvette Cedex