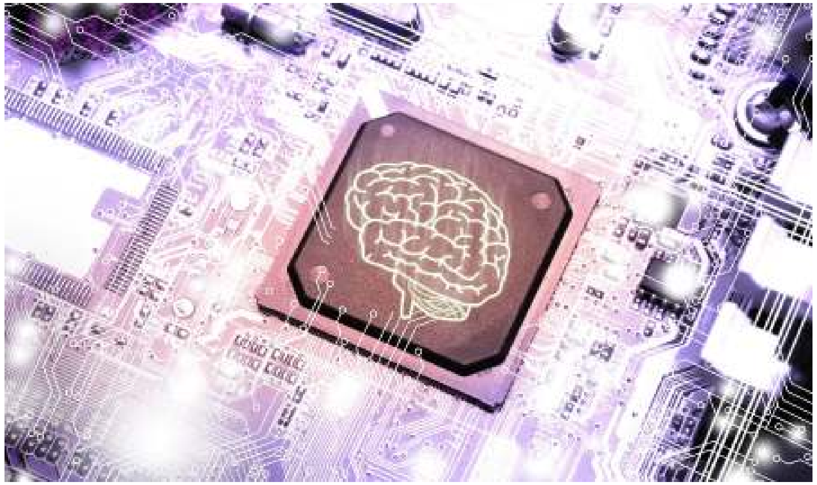


DEEP NEURAL NETWORK PROTOTYPING PLATFORM WITH AUTOMATED MULTI-TARGET HARDWARE EXPORTS AND BENCHMARKING

Nicolas Ventroux

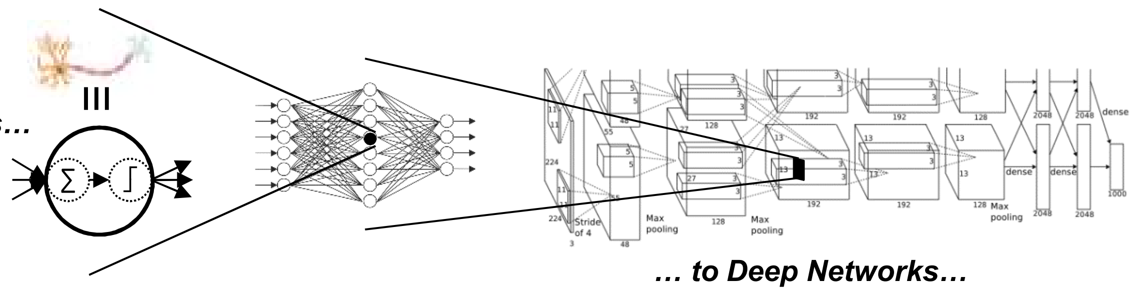
Nicolas.ventroux@cea.fr

Neural Networks

- **From neurons to Deep Neural Networks (NN) and Deep Learning**
  - Scaled-up NN contains millions of neurons
  - Trained with huge datasets (up to millions of images) with gradient descent technics
  - Recurrent NN (RNN) are effective for sequences recognition (speech)
  - Convolutional NN (CNN) use trainable convolution filters for image recognition

… to applications

From neurons…



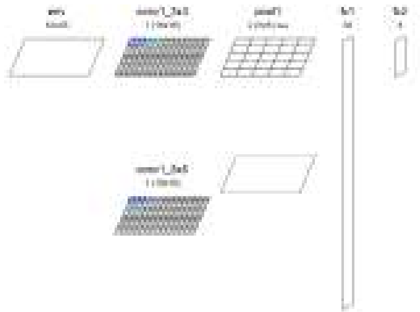
… to Deep Networks…

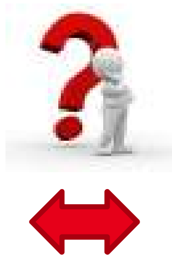
- **Novelties and current trends**
  - Today, the best approach for image/speech recognition, indexing, scene labeling...
  - Reach human performances on some applications
    - E.g. hand writing recognition, traffic sign, face recognition
  - Major players in learning activities
    - Google, Facebook, Microsoft, IBM, Baidu, Qualcomm, Synopsys, Ceva…

A need for fast exploration

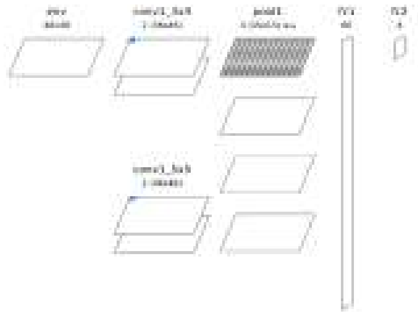
- **Many topology combinations with important result variations**
  - Example: embedded CNN for image classification based on the Caltech-101 image database with 4 categories
    - Motorbike, faces, planes and cars

Topology 1





Topology 2



10,458 MACS
87% recognition rate

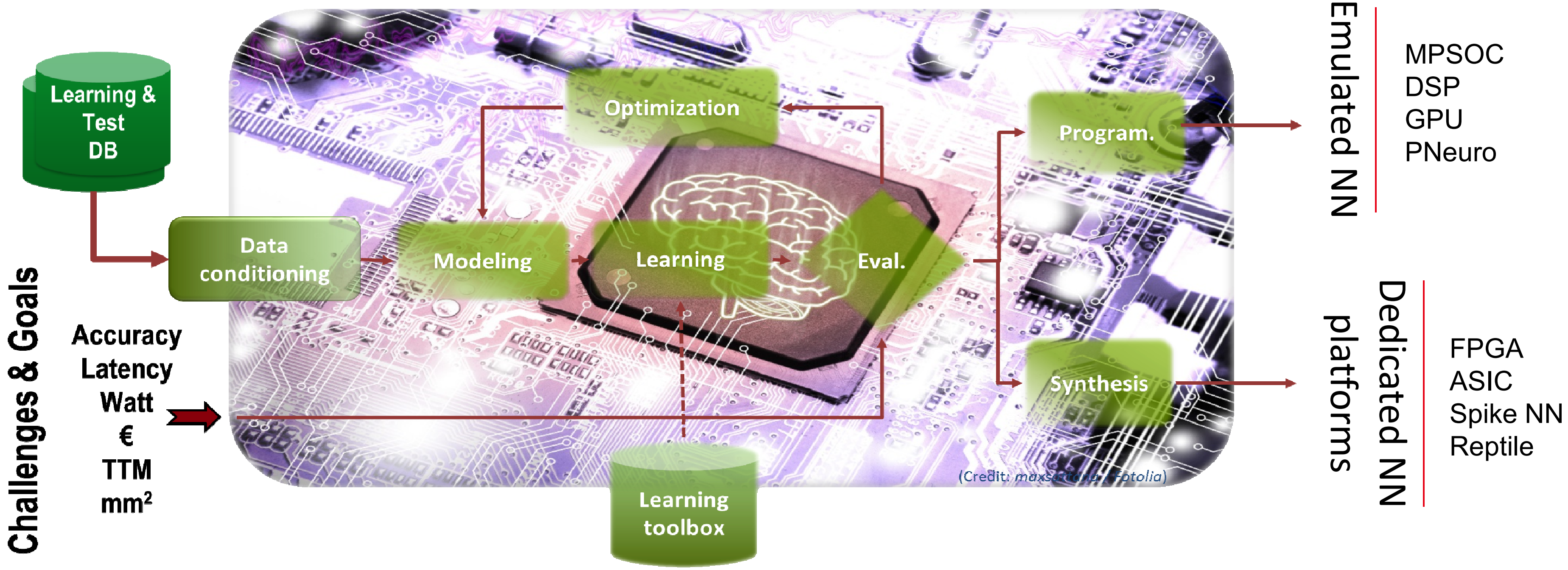
204,396 MACS
96% recognition rate

⇨ Our Positioning

- Build a unified DNN computing platform for *fast design space exploration* and automatic code generation

N2-D2 platform

- **A single platform to**
  - Explore Deep neural network (DNN) topologies
  - Experiment SoA learning technics with large databases
  - Benefit from approximate computing to generate optimized DNN



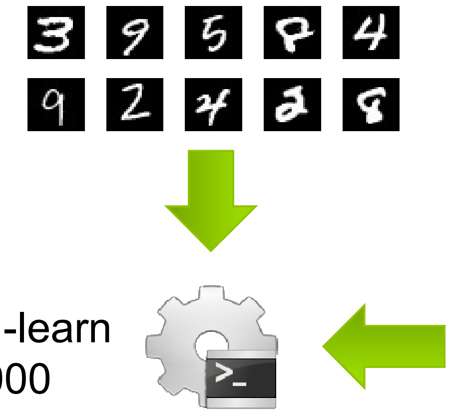
Database handling and data conditioning

- **A large set of available standard databases**
  - Caltech 101/256, FDDB, GSTDB, CIFAR, MNIST…
    - Can be used as benchmarks
    - Available optimized topologies for these benchmarks
  - Set of methods and tools to create new databases
    - For anykind of 1D, 2D or 3D data
    - Advanced Region of Interest (ROI) handling
      - Arbitrary ROI shapes (circular, rectangular, polygonal or pixelwise)
      - Convert ROIs to data point (pixelwise) labels

- **A set of methods for data pre-and/or post-processing**
  - To improve learning efficiency
    - Extract additional ROIs to feed the DNN
    - Elastic distortion, (random)image clipping, scaling, rotation, mirroring…
  - To improve classification
    - Histogram equalization (including CLAHE)
    - Convolutional filtering (Gaussian, Gabor...), DFT…

Neural Network topology exploration

- **Arbitrary combination (type and size) of typical network layers**
  - Convolution
  - Pooling (MAX, average)
  - Fully connected
- **Support of multiple execution models**
  - Formal models
  - Spike-based processing



convnet network.ini -learn 6000000 -log 10000

```
; Environment
[env]
SizeX=29
SizeY=29
ConfigSection=env.config

[env.config]
ImageScale=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

; Common config for static
model
[common.config]
NoBias=1
WeightsLearningRate=0.0005
Threshold=1.0
NoClamping=1
```

Network configuration example for digit recognition

Code generation and platform benchmarking

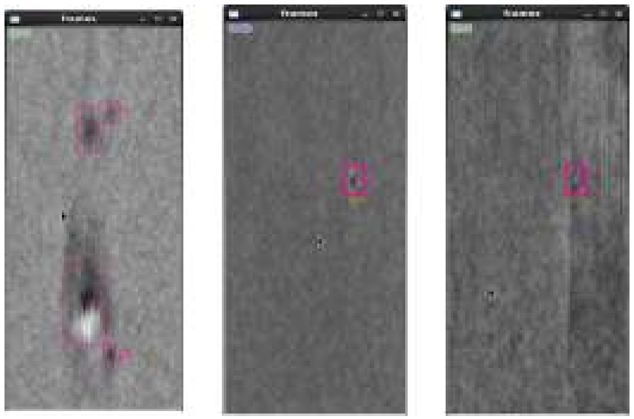
- **The N2-D2 platform generates multiple output formats**
  - Simple C code
    - No dynamic memory allocation and no floating point
  - C code accelerated with OpenMP
  - OpenCL code optimized for either CPU/DSP or GPU
  - CuDNN and CUDA code optimized for NVIDIA® GPUs
  - C code tailored for High-Level Synthesis (HLS) with Xilinx Vivado HLS
    - Direct synthesis to FPGA, with timing and utilization rate after routing
    - Maximum number of clock cycles desired to compute the network
    - FPGA utilization vs number of clock cycle trade-off analysis

- **Optimization through approximate computing**
  - DNN weights and signal data accuracy reduction
  - Multiple methods to round weights
  - Approximations of non-linear network activation functions

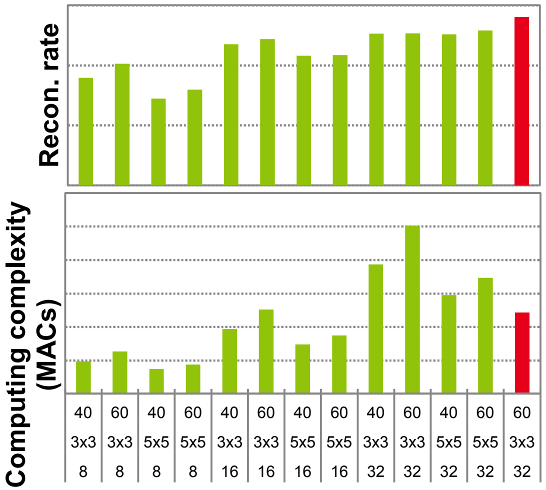
Putting all together : part inspection use-case (conformity, defects…)

- **Defects identification on metal after rolling**
  - Constrains
    - Real-time with extremely high throughput
    - Tiny and low contrasted defects
  - Solutions
    - Database labeling and pre-processing
    - Fast DNN topology exploration
    - Performances vs complexity analysis

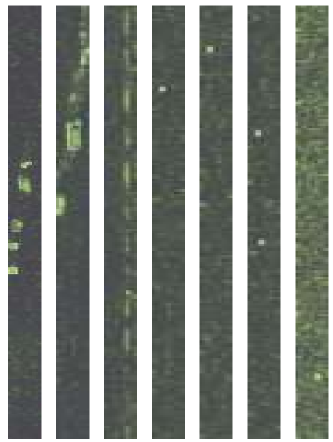
1) Defects labeling and visualization



2) NN Exploration and benchmarking



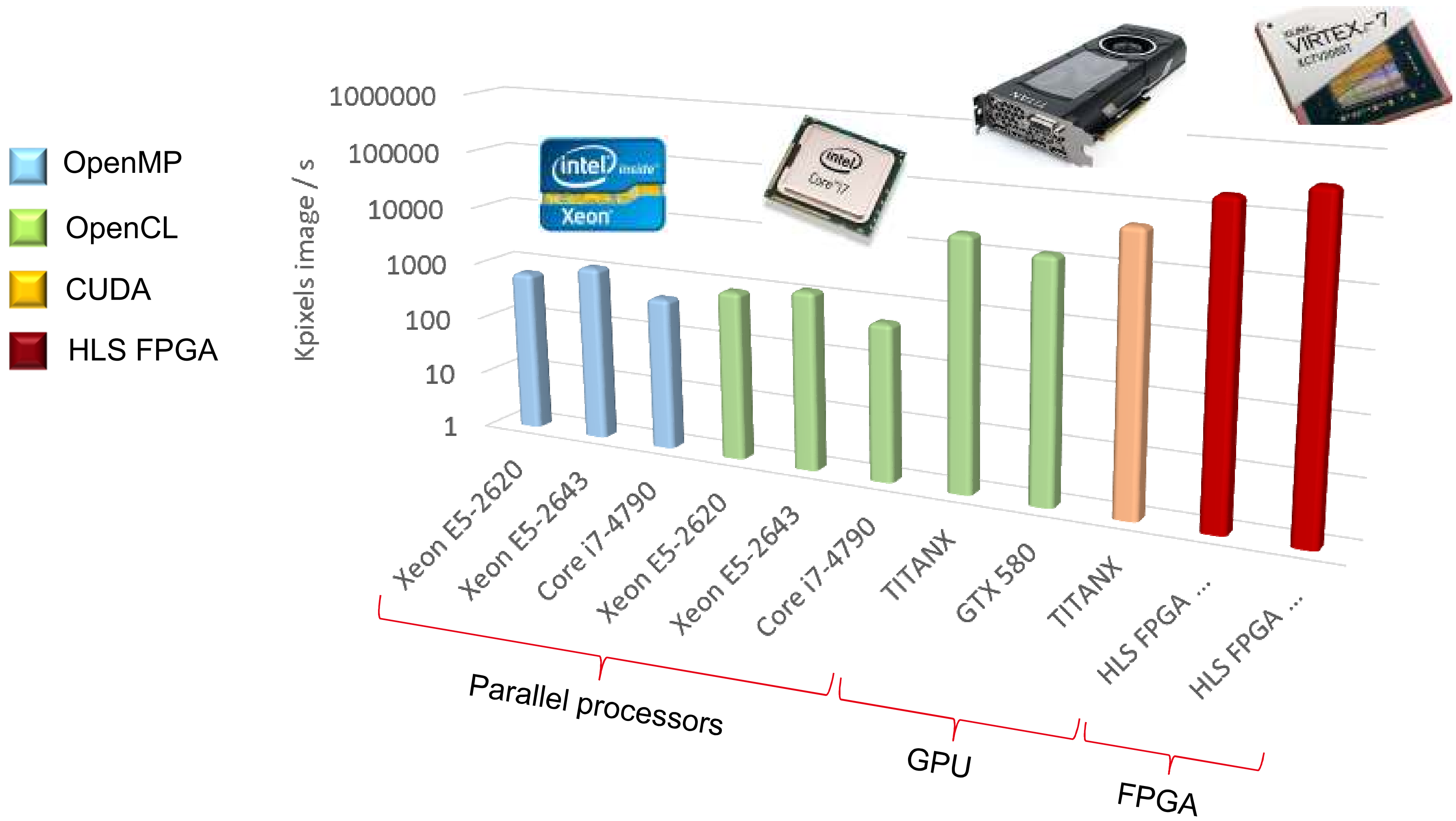
3) Defects identifications after NN learning



➔ From scratch exploration (database and NN construction) to industrial application

➔ 50,000 MACs NN synthetized in 100 cycles on FPGA @ 100 MHz (500 MACs/cycle)

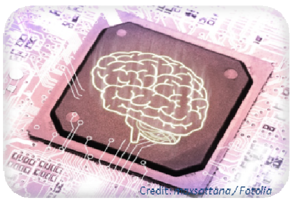
COTS benchmarking for part inspection DNN



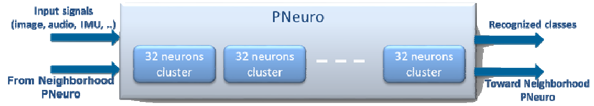
Conclusion

- **Our platform competes with some open-source frameworks in DNN exploration (e.g. Google Tensorflow, UC Berkeley Caffe, Facebook Torch…)**
  - But focuses on the optimization of real-time classification performances;
  - targets large set of computing solutions (FPGA, ASIC, GPU…)
  - and allows advanced DNN model exploration (e.g. spike-based processing)

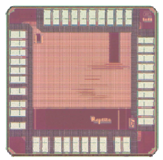
- **Our platform can generate code for DNN accelerators to outperform TOPS/Watt (PNeuro)**



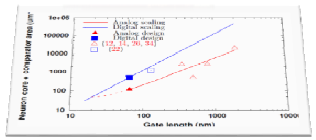
DNN Processor



Spike-based accelerator



Mix-Signal IC



Thank you

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