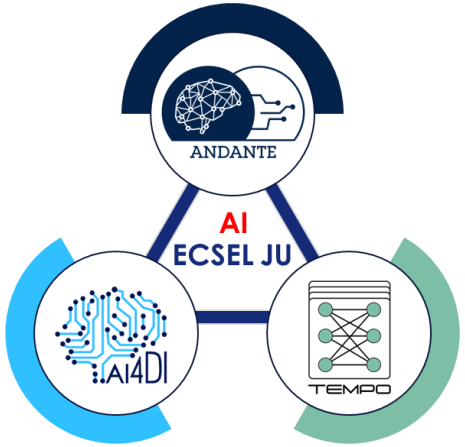

# International Workshop on Embedded Artificial Intelligence Devices, Systems, and Industrial Applications (EAI)

MILAN
19-22 SEPTEMBER 2022
ESSDERC
ESSCIRC

Milan, Italy 19 September 2022

# International Workshop on Embedded Artificial Intelligence Devices, Systems, and Industrial Applications (EAI)




## Meeting the latency and energy constraints on timing-critical edge-AI systems

**Ivan Miro-Panades, Inna Kucher, Vincent Lorrain, Alexandre Valentian**

**CEA-List**


**19 September 2022 Milan, Italy**

# Presentation Outline

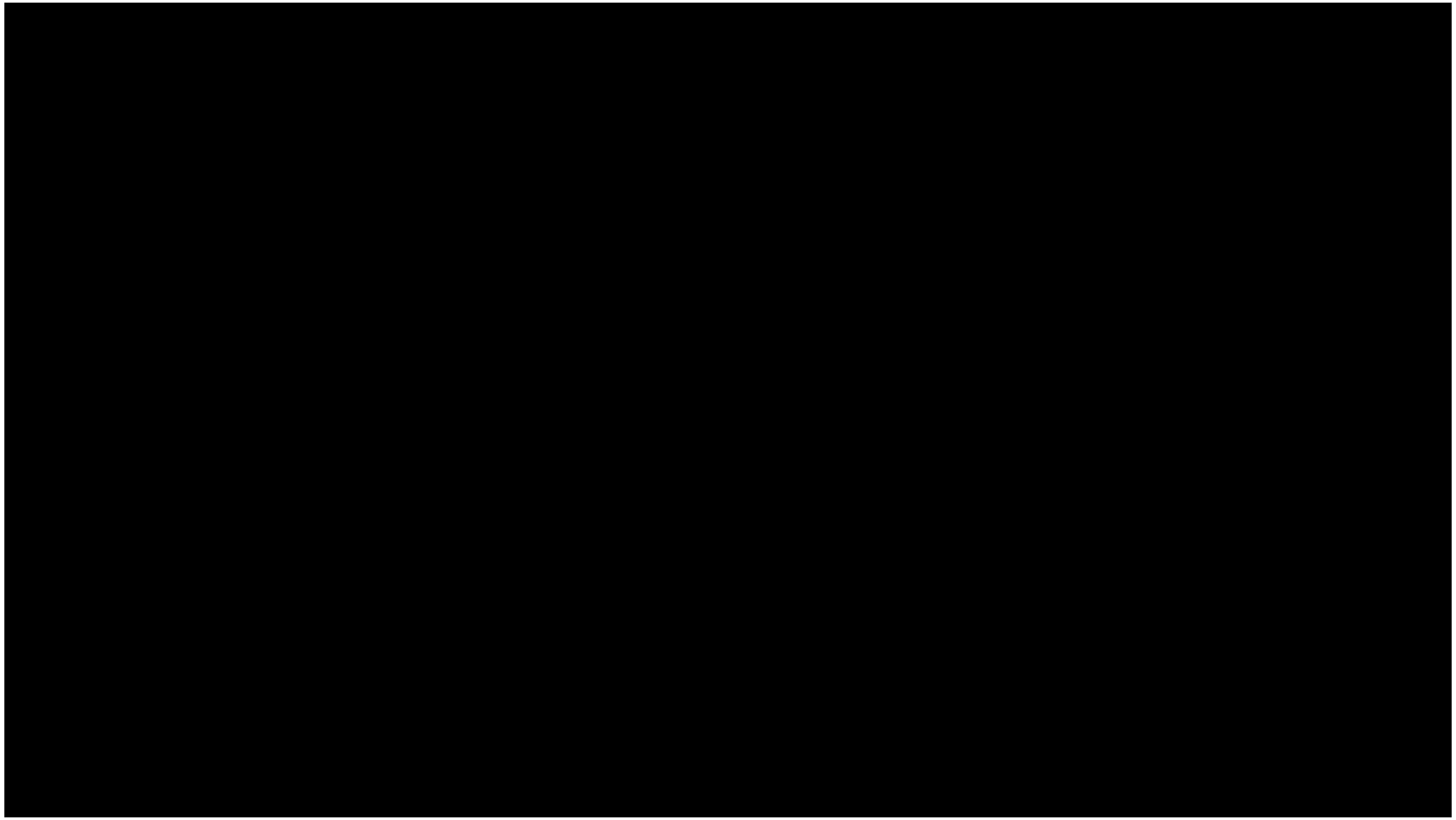
- Introduction
- Dual-system approach
- NeuroCorgi overview
  - Low energy
  - Low latency
  - ASIC design
- Conclusions

# Introduction

Instructions: Count how many times the players wearing white pass the basketball

Source: https://www.youtube.com/watch?v=vJG698U2Mvo

# Introduction



Instructions: How many gumdrops is the person eating?

Source: https://www.youtube.com/watch?v=dbjPnXaacAU

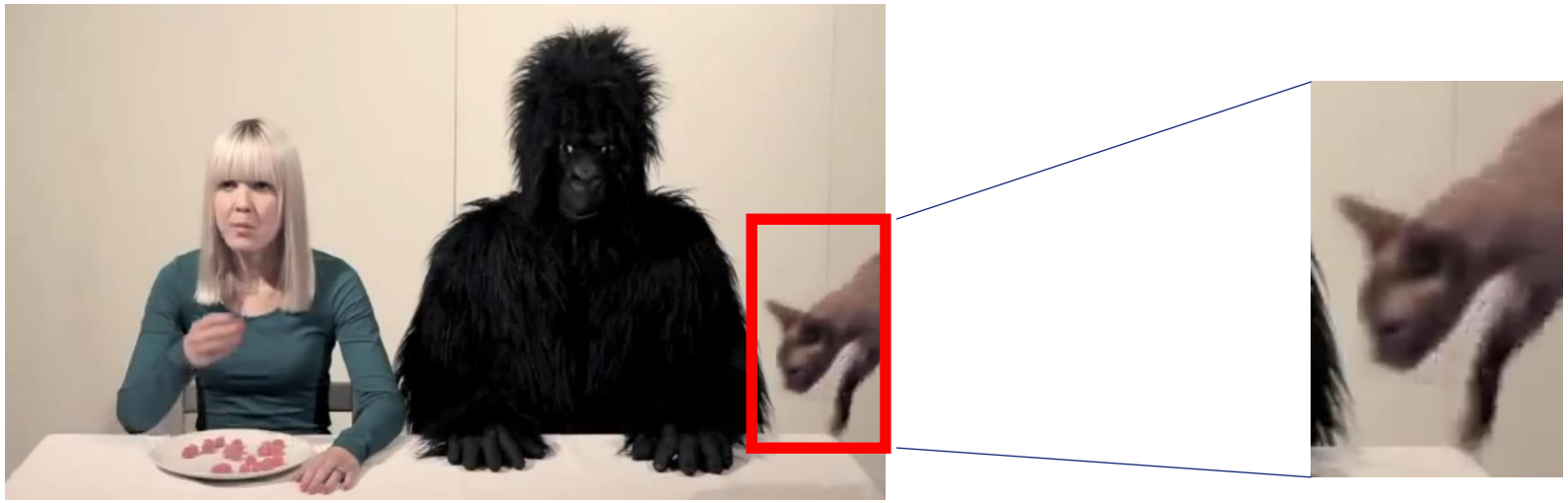
# Introduction

- Is our “brain AI” efficient?
  - When I count objects, I don’t see background images…
  - When I focus my attention, I lost my attention on sudden images…
- Can I trust my “brain AI” to drive a car, a plane?

*Performing an image detection for the **full** image is **too power consuming***
*⇒ Use bio-inspiration to overcome energy and latency constrains*

# Introduction: Vision system

- See something
  - Where?
- Watch something
  - What?

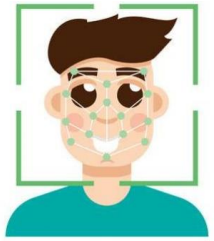


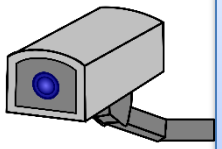
***Where ?***

***What ?***

*(Computed in portions of the image)*

# Dual system approach





Where ?

What ?

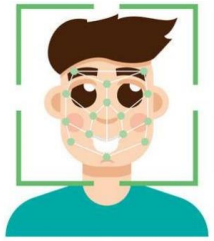
**Localize objects**

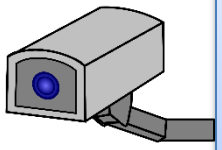
- Medium precision
  - False positives accepted
- Low latency
  - Lower latency => less buffering
  - Batch size = 1
- High energy efficiency
  - Performed on the full image

**Detect objects**

- High precision
  - Detect with precision the objects
- Latency is less critical
- Medium energy efficiency
  - Performed on small sub-images
  - Adapted to each application

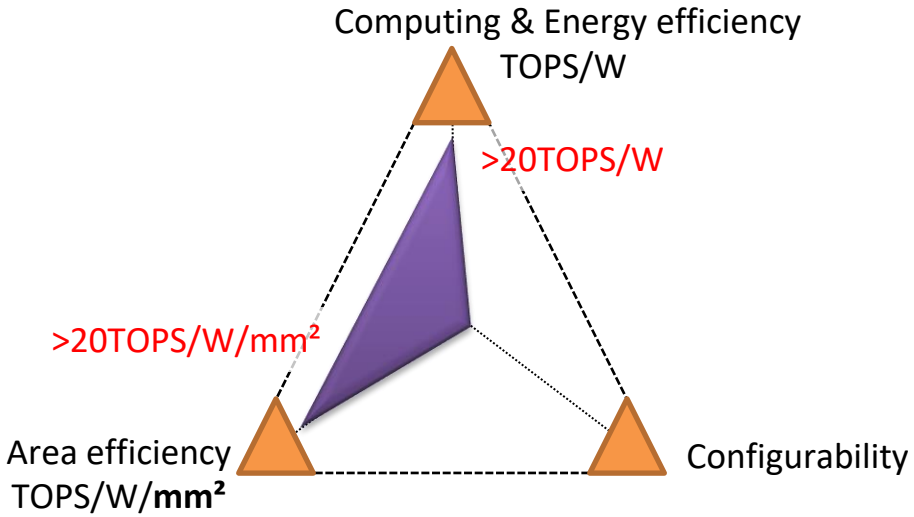
# Dual system approach

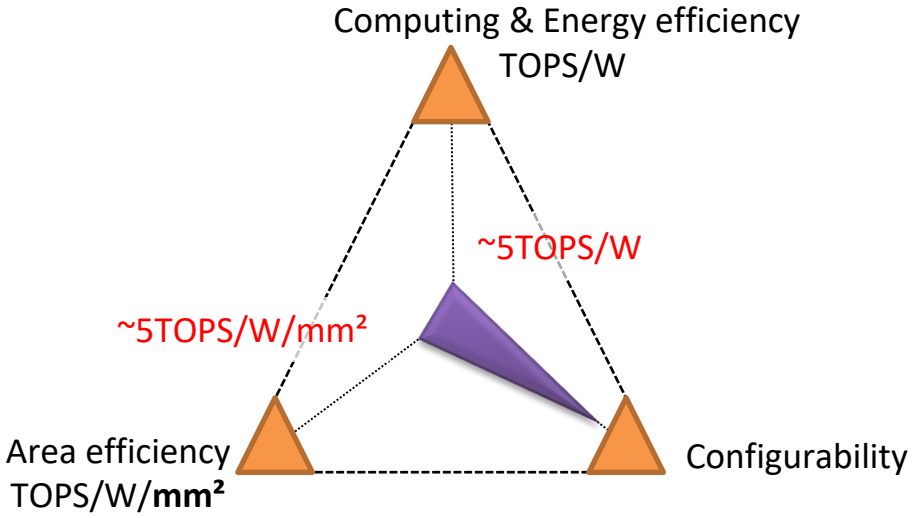




NeuroCorgi

What ?



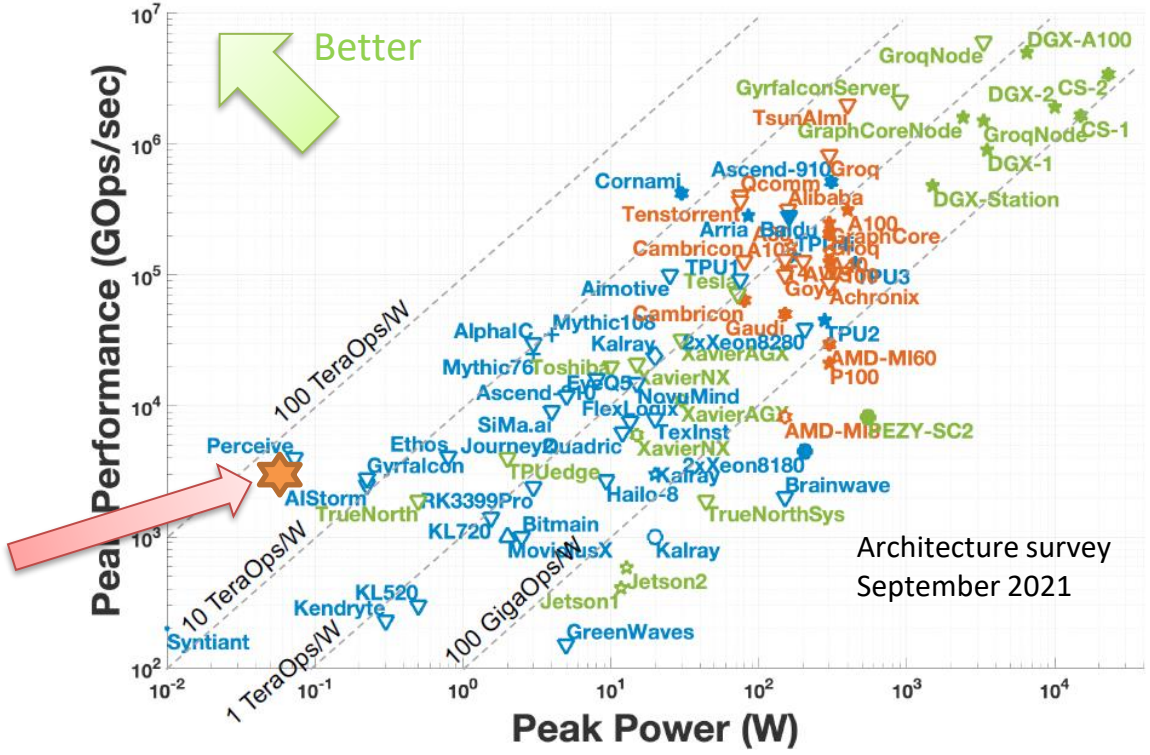
# NeuroCorgi overview

- Objective:
  - Low-power low-latency AI accelerator to compute the features of an image to classify and detect objects with a fixed topology and weights
  - HD images of 1280x720pixels at 30 frames per second
  - Preprocessing engine to identify the key points in the image before using a high-end AI engine

- Positioning w.r.t. the State of the Art
  - Best in class energy per inference with HD images
    **< 4mJ/inference** (HD images @30FPS)
  - Best in class latency with batch size 1
    **< 15ms** (HD images @30FPS)

- Planned a tape-out on November 2022
  - GF 22FDX technology
  - Embedded Non-Volatile Memory (NVM)

**Our target**



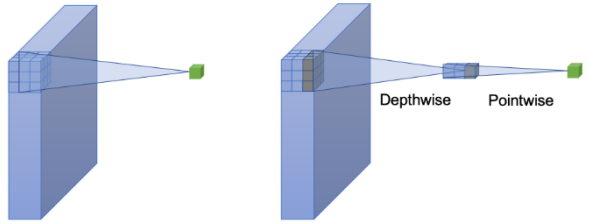

Architecture survey
September 2021

# NeuroCorgi - main pillars

- Low energy per inference
- Low latency

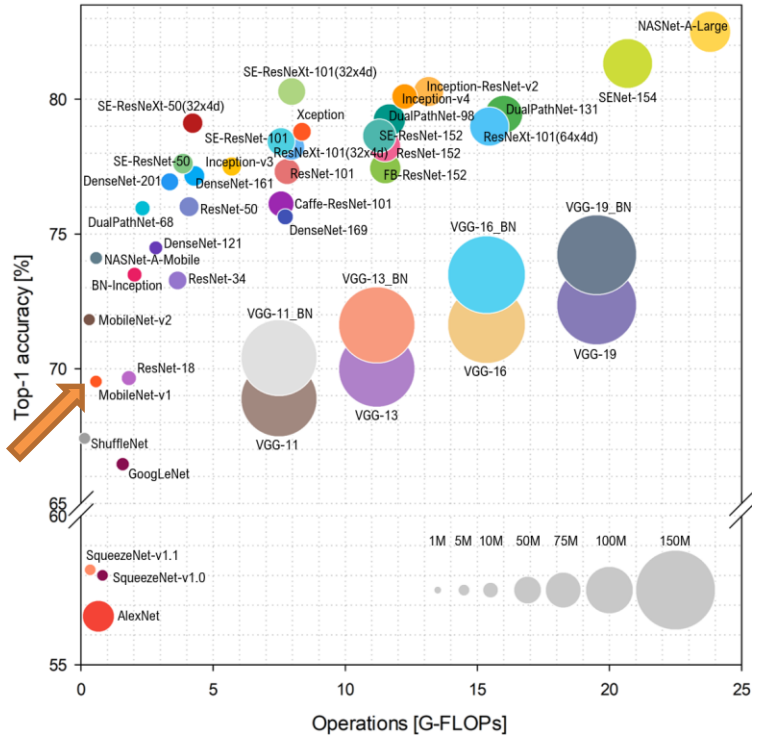
# Low energy per inference – Network architecture

- Select the best tradeoff between network complexity and operations per inference
  - Lower operations => lower MACs per images => lower energy
  - MobileNet v1 uses depth-wise and point-wise convolutions to reduce the computing complexity




Standard convolution  Depth-wise + point-wise convolution

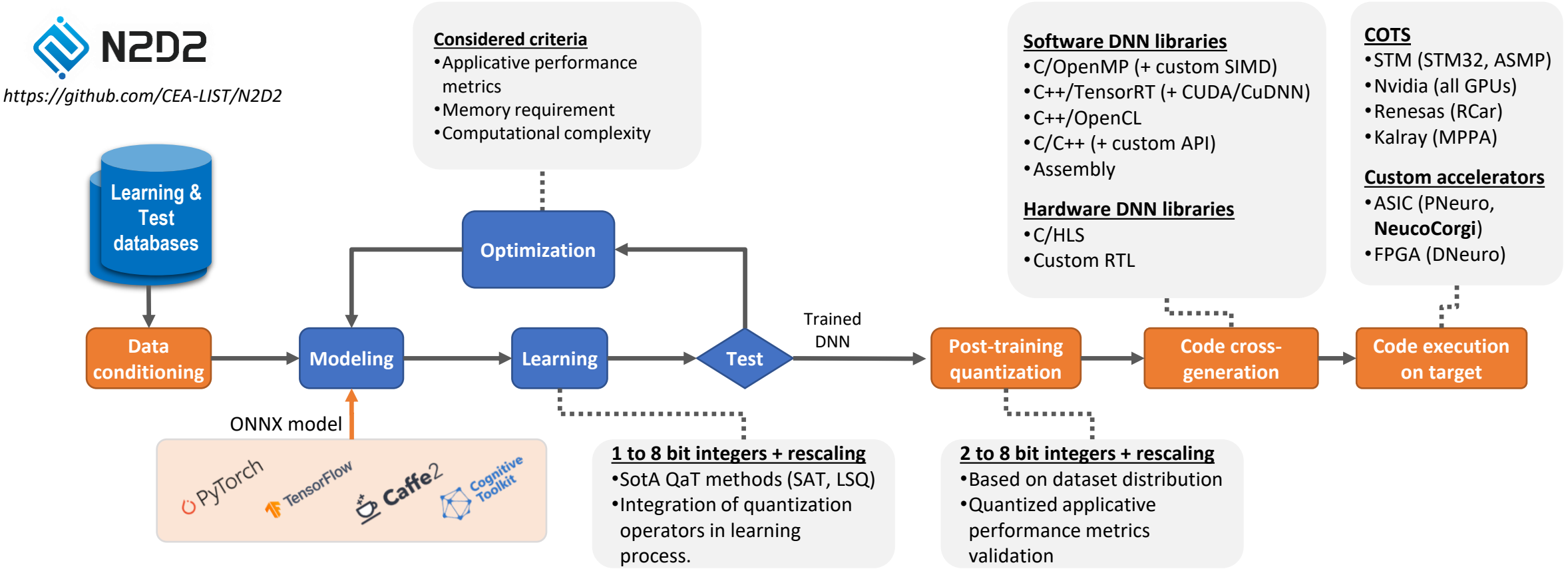
- Leverage on fixed topology and fixed weights
  - Fixed topology optimizes the buffering and the inter-layer communication throughput
  - Fixed weights allows to fix in the ASIC the weight values




S. Bianco, "Benchmark Analysis of Representative Deep Neural Network Architectures,"

# Low energy per inference - Quantization

- Quantization for weights and activations
  - Lower the energy per MAC operation
  - Lower the bits for weight and activation storage => less bits means less area and less leakage
  - Leverage on N2D2 framework => Quantization Aware Training



# Low latency – streamed architecture

- Streamed network layer architecture
  - Network layers are computed in pipelined fashion instead of computing layer-wise the network

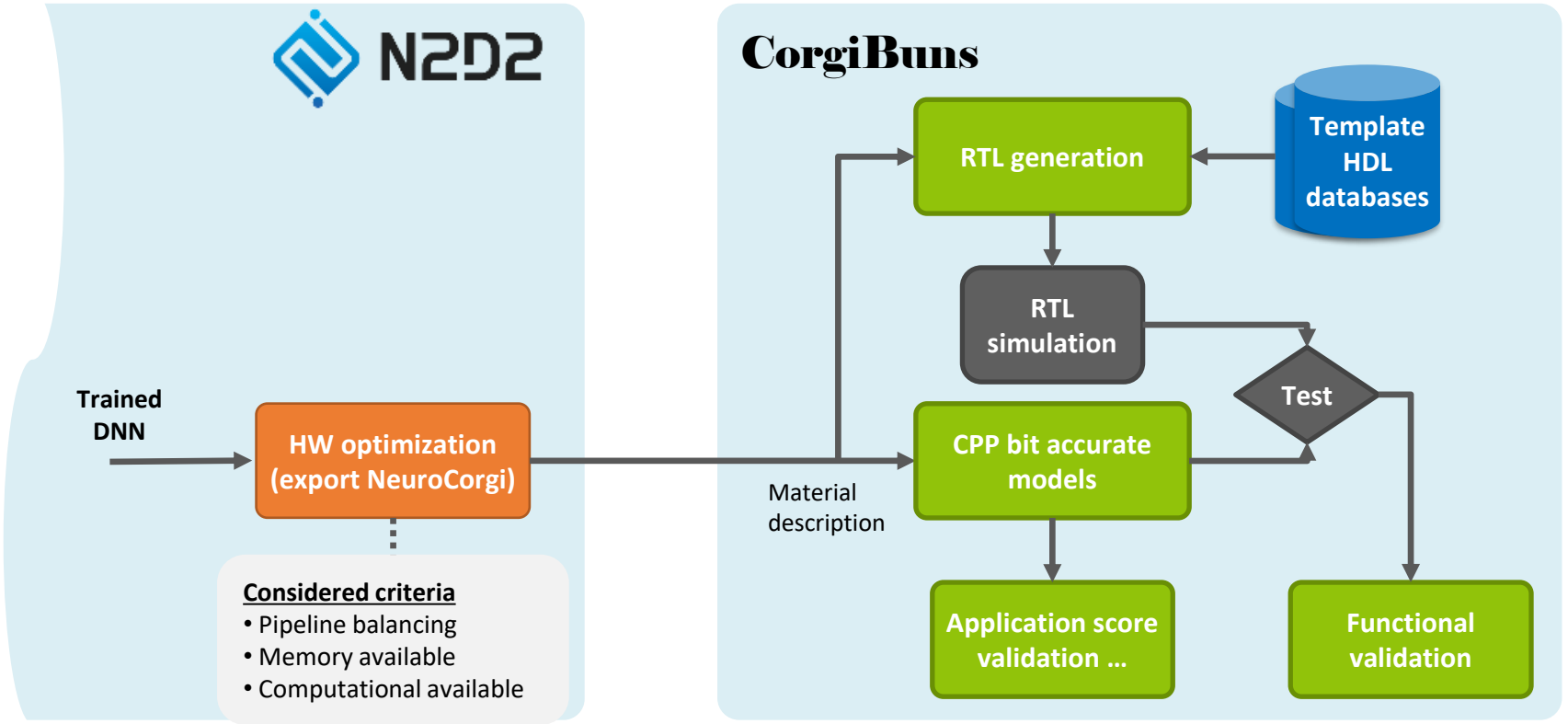
- **A streamed network layer architecture is normally less energy efficient than a non-pipelined architecture**
  - Layer-wise architecture allows to compute multiple images per layer (batch size) => weights are read once and reused for the multiple images => less energy
  - Streamed architecture means different HW elements for each network layer
    - Higher buffering, higher area, higher leakage
    - Not time division multiplexing

- **But batch size of 1 is mandatory**
  - It is not possible to compute multiple images in parallel
- **By fixing the network topology and the weight values it is possible to limit the area overhead of a pipelined architecture**

*⇒ Streamed architecture offers a good tradeoff between latency and energy efficiency*

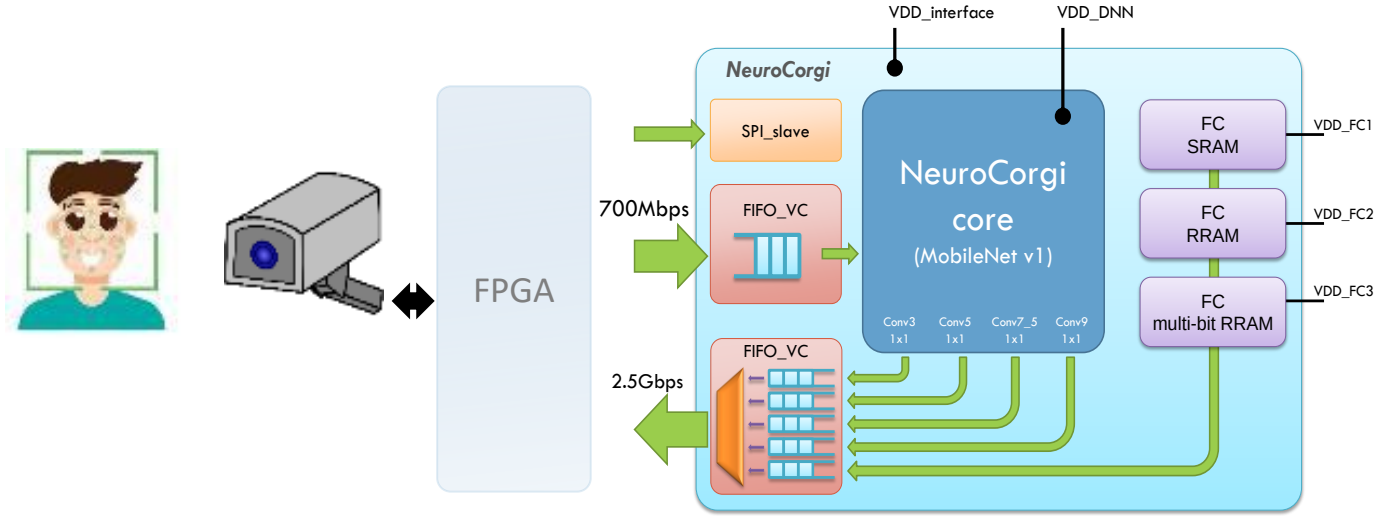
# NeuroCorgi architecture builder

- Automatic RTL generation and bit-accurate CPP model of DNN (ONNX)
- Functional validation of the generated architecture
- Customizable HW parameters (MACs per layer, bit precision, pipelining)
- Fast architecture exploration

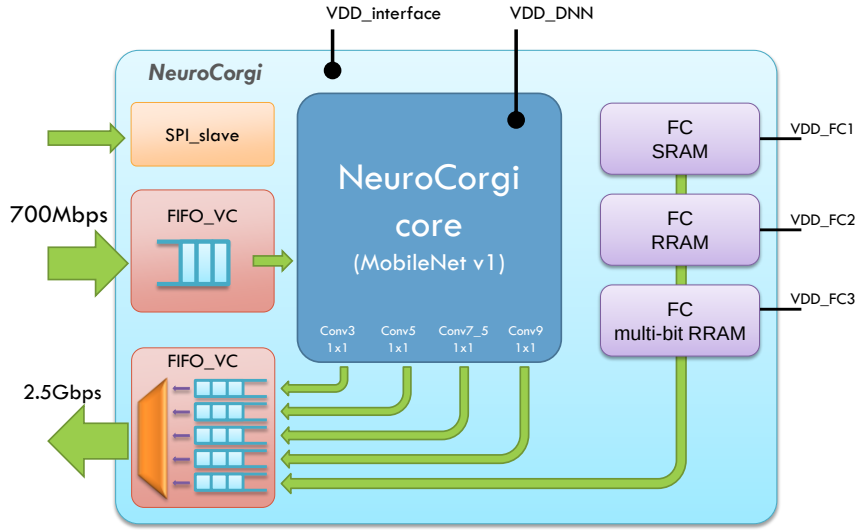


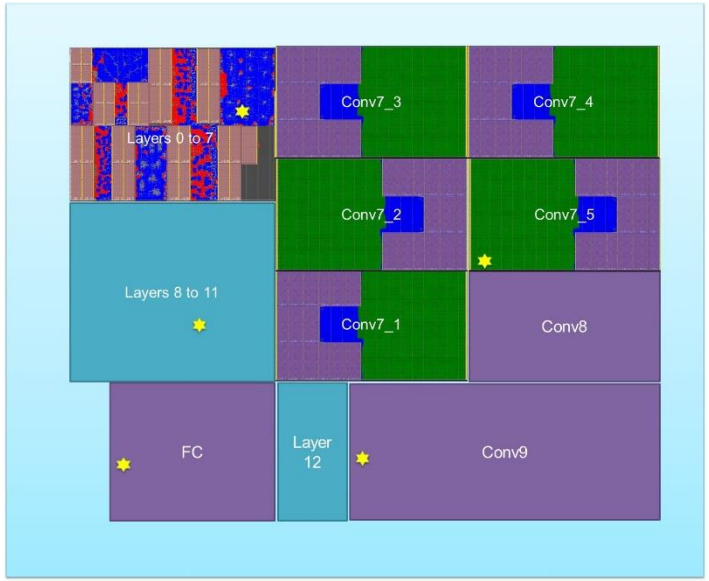
# NeuroCorgi ASIC

- NeuroCorgi is a IA feature extractor of HD images
- Requires an FPGA to input the image data and to extract the computed features
- High throughput interfaces >3Gbps with parallel data (FIFO_VC)
- Embeds a fully connected layer to classify the input images
- Implemented in three manners (SRAM, NVM, multi-bit NVM) for comparison
- Separate power domains for power/performance analysis



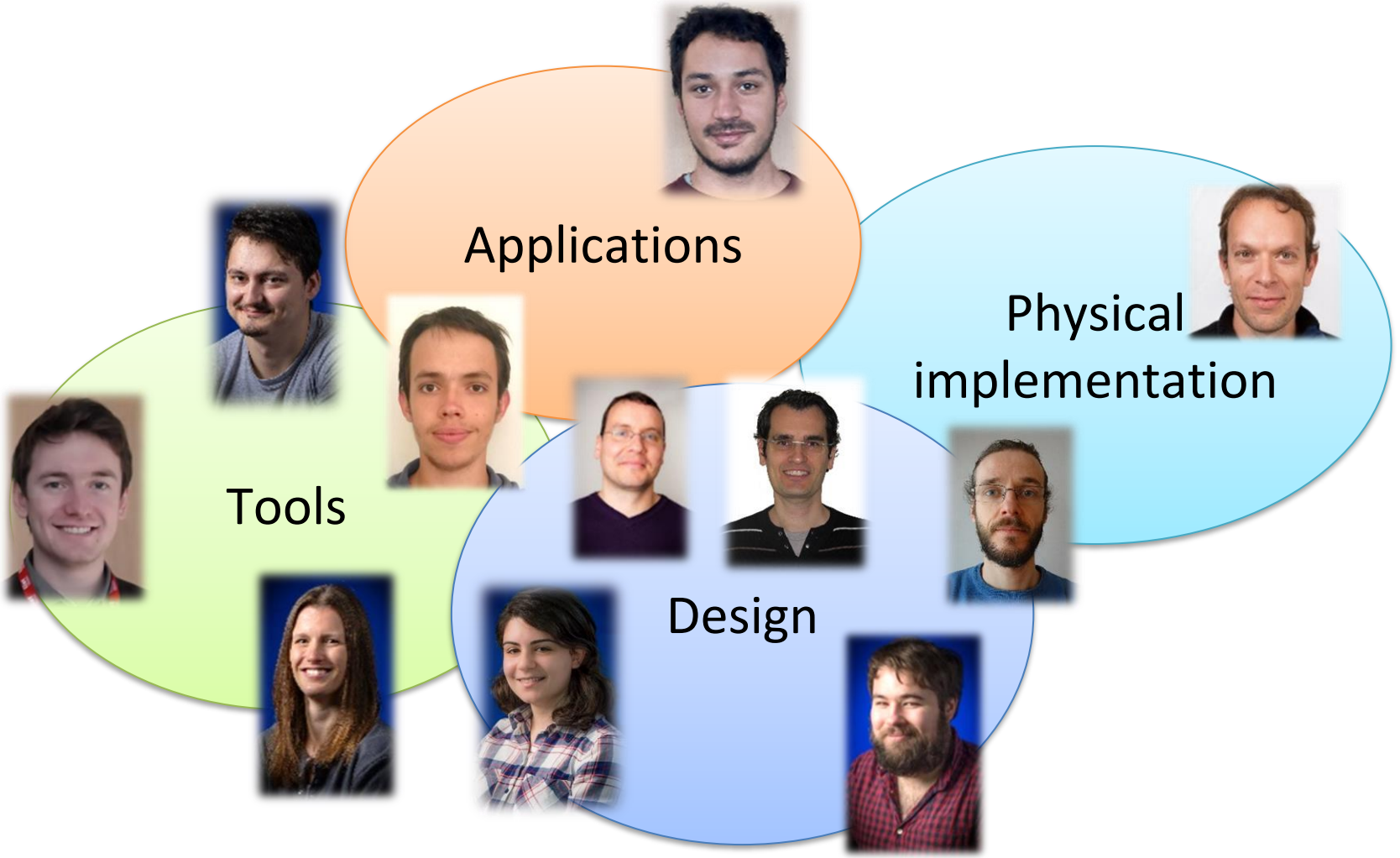
# NeuroCorgi ASIC



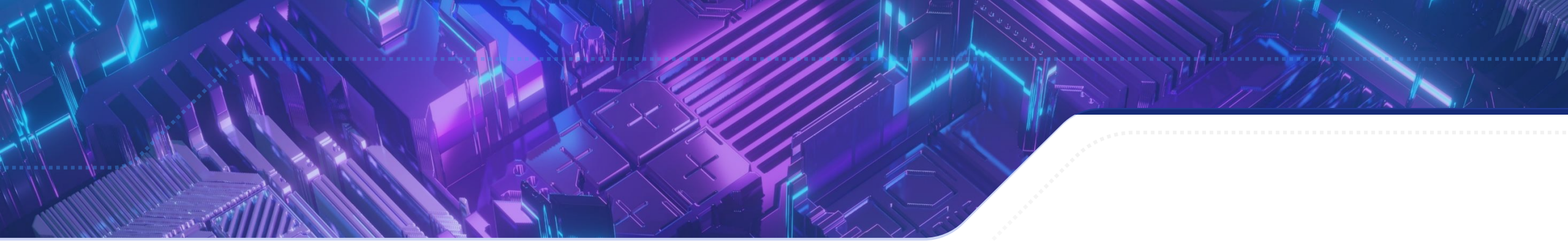

| | |
|---|---|
| **Topology** | MobileNet v1 |
| **Image size** | 1280x720 pixels @30 FPS |
| **Image pixel** | 24 bits RGB |
| **Weight quantization** | 4 bits |
| **Activation quantization** | 4 bits |
| **Inference latency** | < 15 ms |
| **ASIC target technology** | GF 22FDX |
| **Implementation** | Digital |
| **Expected area** | ~10 $mm^2$ |
| **Package** | QFN open cavity |
| **Power domains** | PADs, Periphery, Core |
| **Input data link** | 700 Mbps |
| **Output data link** | 2500 Mbps |
| **Output data channels** | 4 layers + classification |
| **Configuration interface** | SPI @ 40 MHz |
| **External memory** | No |
| **Embedded memory** | ~ 600 kbytes |
| **Expected efficiency** | > 20 TOPS/W |
| **Expected power consumption** | < 100 mW |
| **Voltage supply** | 1.8 V IOs<br>0.6 to 0.9 V core |

# NeuroCorgi Team



# Conclusions

- A dual system with where/what is a promising approach to address low-energy and low-latency systems
- NeuroCorgi leverages on a fixed-topology with fixed weighs to meet the energy and latency requirements
- N2D2 framework is used to train the DNN network with 4bits
- An architecture builder generates an RTL/CPP models
- Expected ASIC results are <4mJ/inference and <15ms latency on HD images at 30FPS



# Thank You

## For your attention

@ ivan.miro-panades@cea.fr