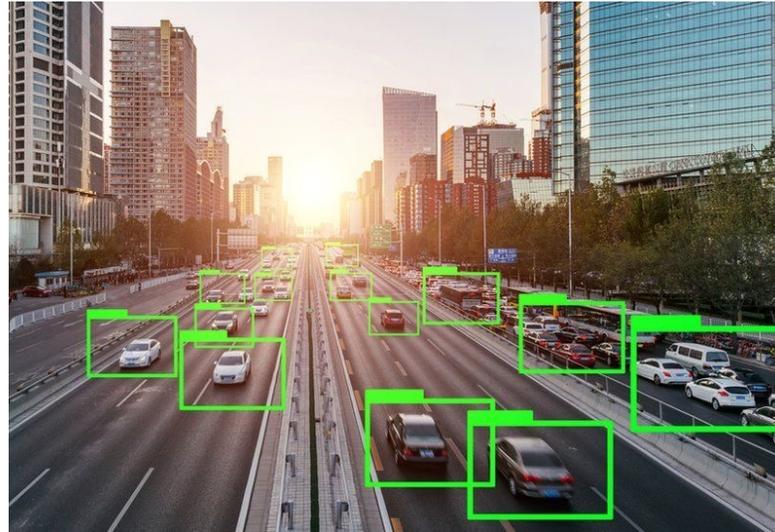


Deep Neural Networks in the Embedded Space: Opportunities and Challenges

Christos-Savvas Bouganis

Intelligent Digital Systems Lab
Dept. of Electrical and Electronic Engineering

www.imperial.ac.uk/idsl



Machine Learning stories dominate our lives

Let's ask ChatGPT

ChatGPT 3.5 ▾



You
structure for a presentation on challenges of AI in embedded systems

ChatGPT
Creating a presentation on the challenges of AI in embedded systems involves addressing technical, ethical, and practical aspects. Here's a suggested structure for your presentation:

1. Title Slide:

- Title: "Challenges of AI in Embedded Systems"
- Subtitle: Navigating the Complex Landscape
- Your name and date

2. Introduction:

- Brief overview of AI in embedded systems
- Importance of AI in embedded systems
- Examples of AI applications in embedded systems (e.g., IoT devices, edge computing)

3. Definition and Basics:

- Define embedded systems
- Explain the integration of AI into embedded systems
- Briefly discuss the significance of real-time processing in embedded AI

4. Key Challenges:

4.1 Technical Challenges:

- **Hardware Limitations:**
 - Discuss the constraints of embedded systems hardware (power, memory, processing capabilities).
 - Explore how these limitations impact AI algorithms and models.
- **Real-time Processing:**
 - Address the need for real-time processing in embedded systems.
 - Discuss the challenges of achieving low latency AI in embedded devices.
- **Optimization:**



ChatGPT 3.5 ▾

Real-time processing:

- Address the need for real-time processing in embedded systems.
- Discuss the challenges of achieving low-latency AI in embedded devices.

Optimization:

- Explore the challenges of optimizing AI algorithms for resource-constrained environments.
- Discuss techniques for model compression, quantization, and efficient inference.

4.2 Ethical and Security Challenges:

Data Privacy:

- Address concerns related to data privacy in embedded AI systems.
- Discuss how sensitive data is handled and potential risks.

Security Risks:

- Explore potential security vulnerabilities in embedded AI.
- Discuss the importance of securing both the AI models and the data they process.

Ethical Considerations:

- Discuss ethical challenges related to AI decision-making in embedded systems.
- Address issues such as bias, fairness, and accountability.

5. Case Studies:

- Present real-world examples of challenges faced in embedding AI.
- Discuss how companies or projects have addressed or mitigated these challenges.

6. Solutions and Innovations:

- Discuss emerging technologies and strategies to overcome challenges.
- Explore ongoing research and development in the field.

7. Future Trends:

- Discuss potential advancements in embedded AI technology.
- Predict how future innovations may address current challenges.

8. Conclusion:

- Summarize key challenges discussed.
- Emphasize the importance of addressing these challenges for the advancement of embedded AI.

9. Q&A:





You

how large is your LLM model

700 GB

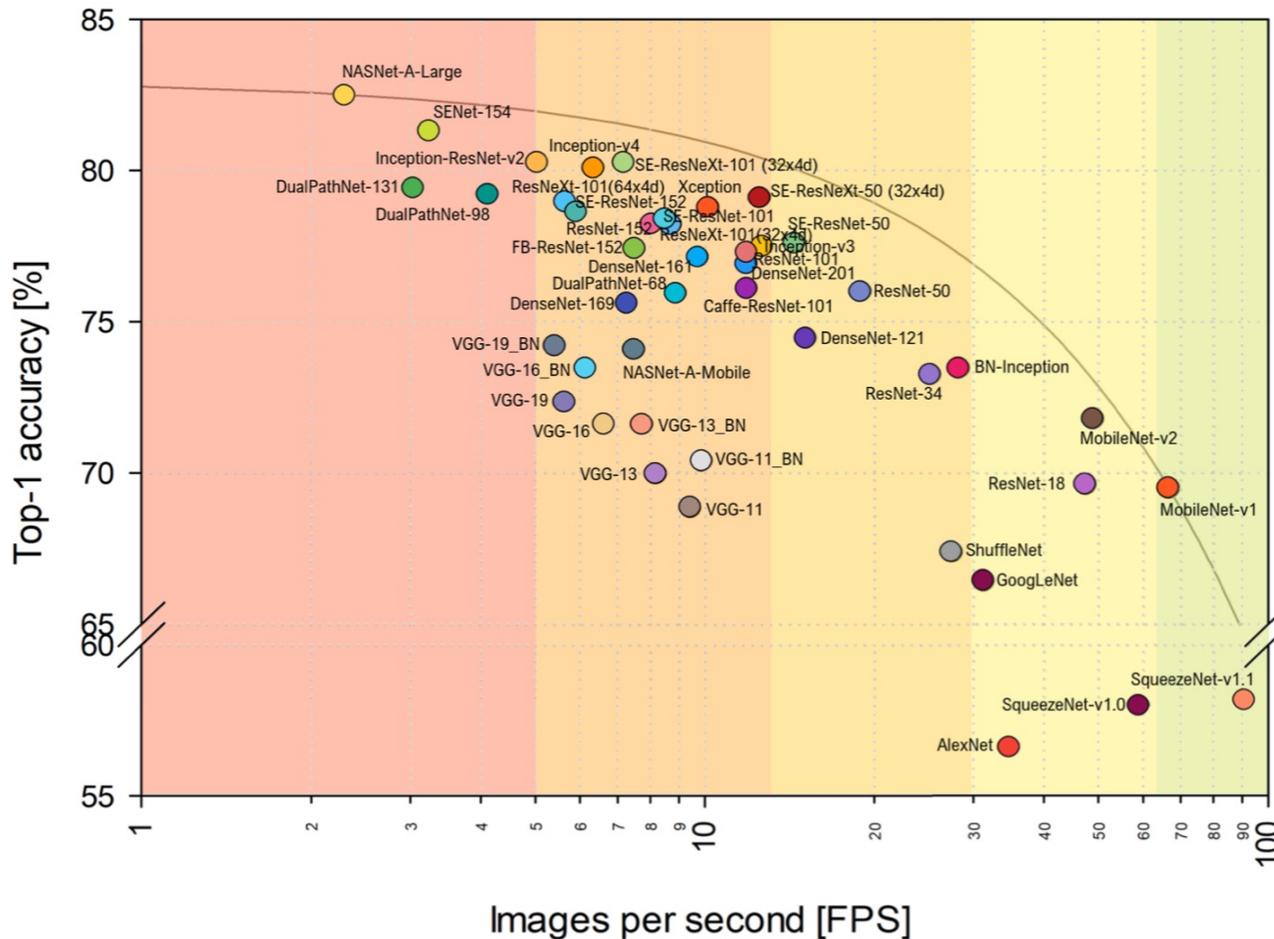


ChatGPT

I am based on the GPT-3.5 architecture, which is one of the largest language models created by OpenAI. GPT-3.5 has 175 billion parameters, making it significantly larger than its predecessor, GPT-3. The large number of parameters contributes to its ability to understand and generate human-like text across a wide range of topics and tasks. Keep in mind that the size of a language model is a key factor in determining its capabilities, but it's not the only factor—training data, fine-tuning, and other aspects also play crucial roles.



Evolution of ML classification models



Observation:
A fast-evolving Pareto front that requires fast developing cycles

Efficiency comes from customisation

Platform Layer



DSPs
Qualcomm Hexagon,
Apple Neural Engine,



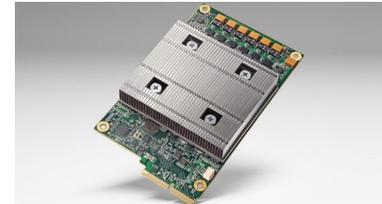
GPUs
Tegra K1, X1 and X2



FPGAs
Custom datapath
Custom memory subsystem



ASICs
TPU

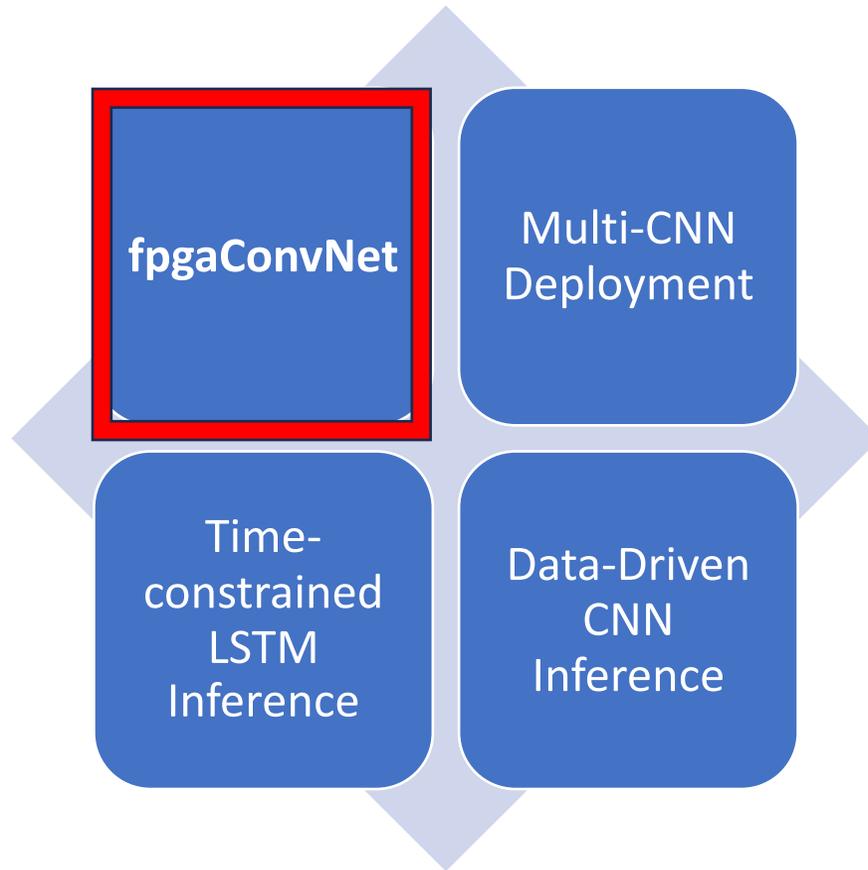


Customisation



Algorithmic Layer

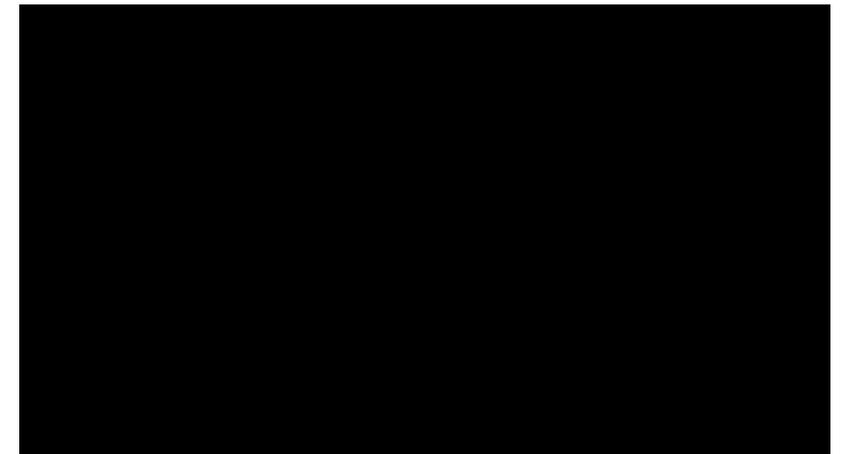
Tools



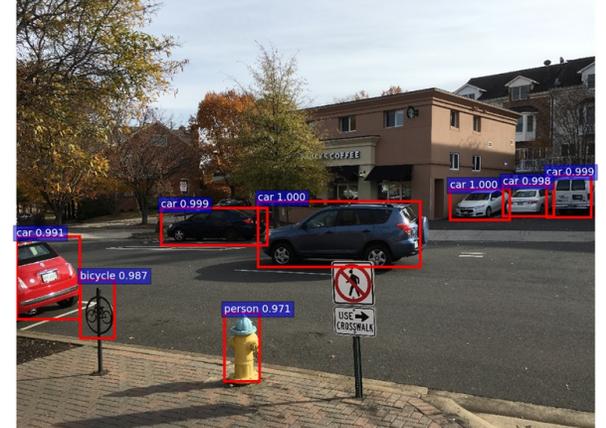
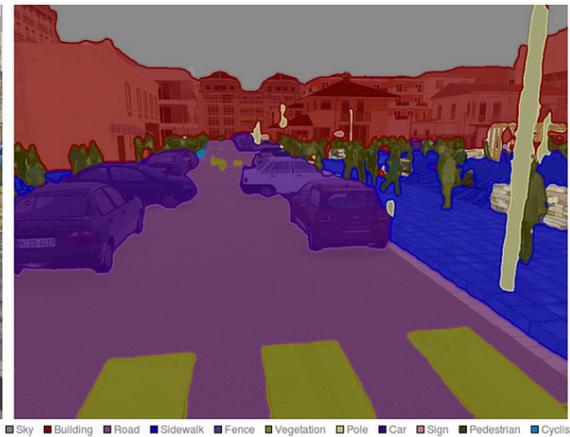
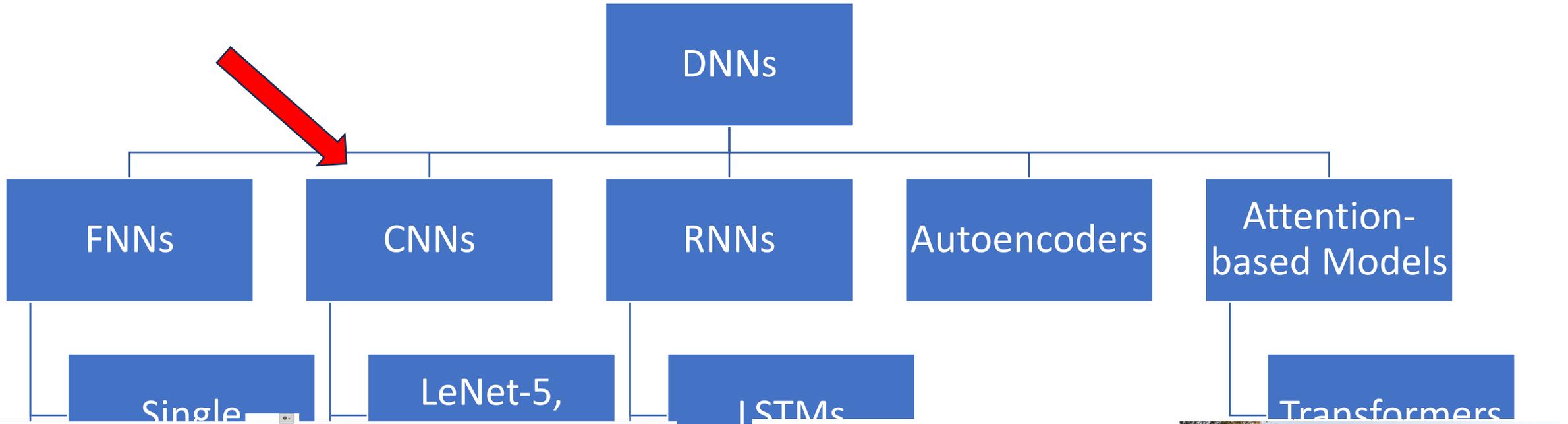
Autonomous Navigation



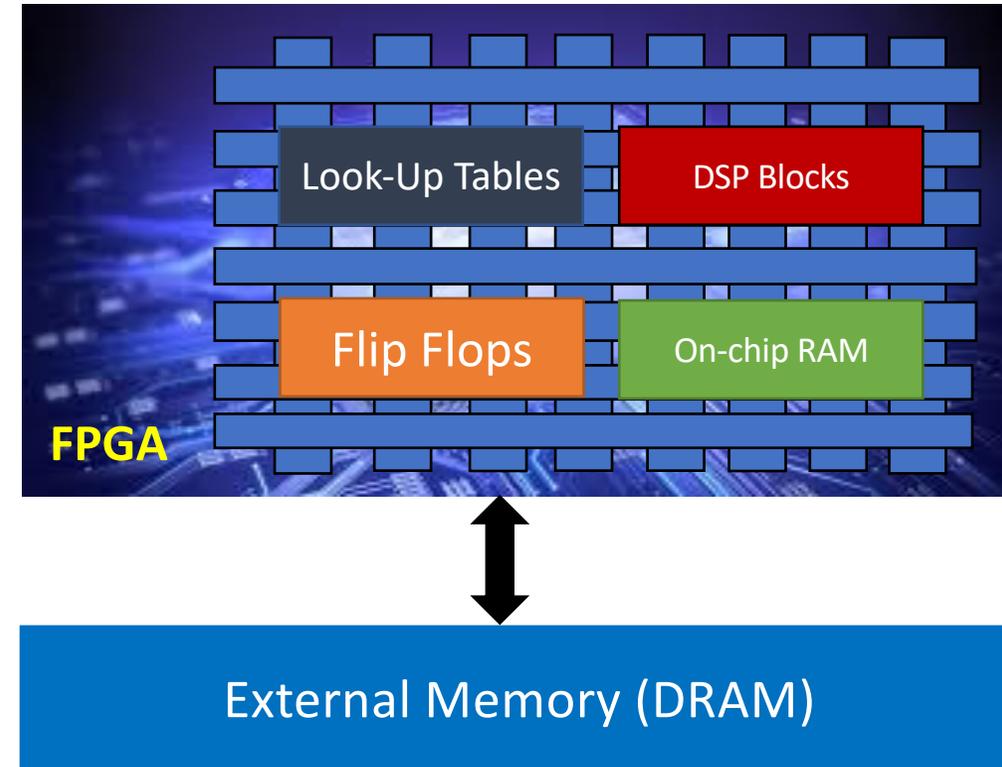
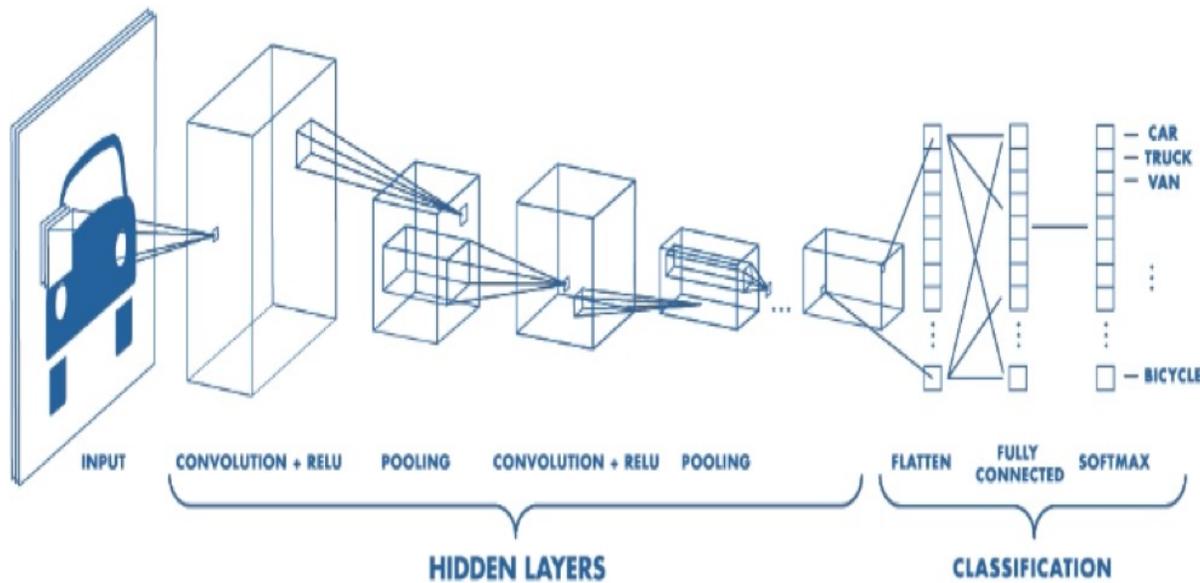
Traffic Detection



A taxonomy of DNNs



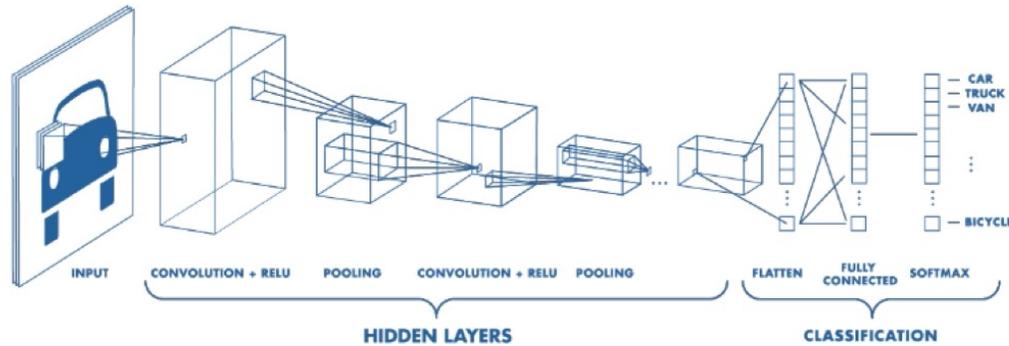
CNN acceleration through an FPGA



Characteristics

- Custom datapath
- Custom memory subsystem
- Programmable interconnections
- Reconfigurability
- Heterogeneous
- Difficult to program

The Challenge of the Mapping Problem



Challenges:

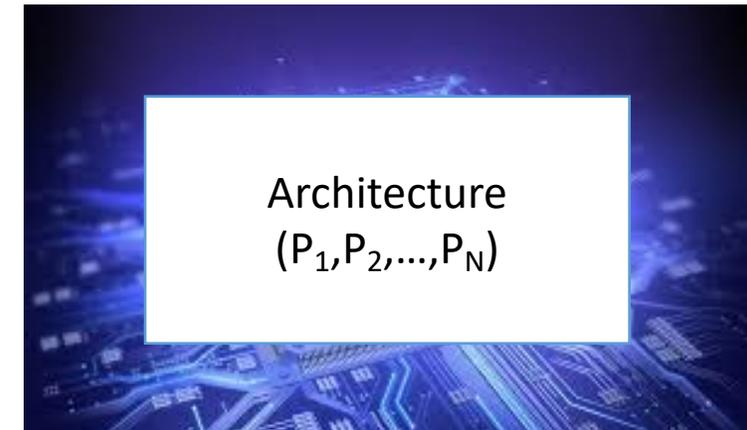
- Competition (or need for performance) => **Highly customised architecture**
- Diversity of operations in modern DNN
- Diversity and resources of modern FPGAs
- Large number of parameters in the target architecture => DSE



| Parameters | Value |
|-----------------|-------|
| LC | 2M |
| BRAMS (36kbits) | 1,880 |
| DSPs | 3,360 |

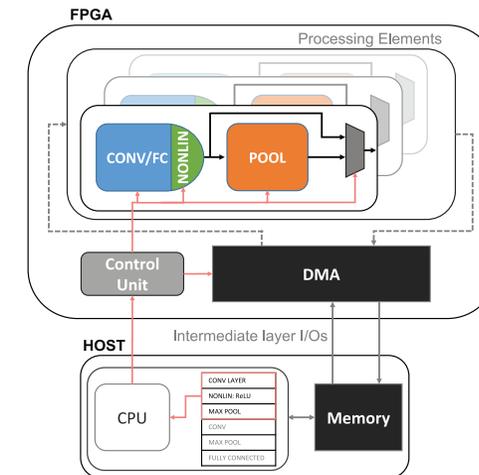
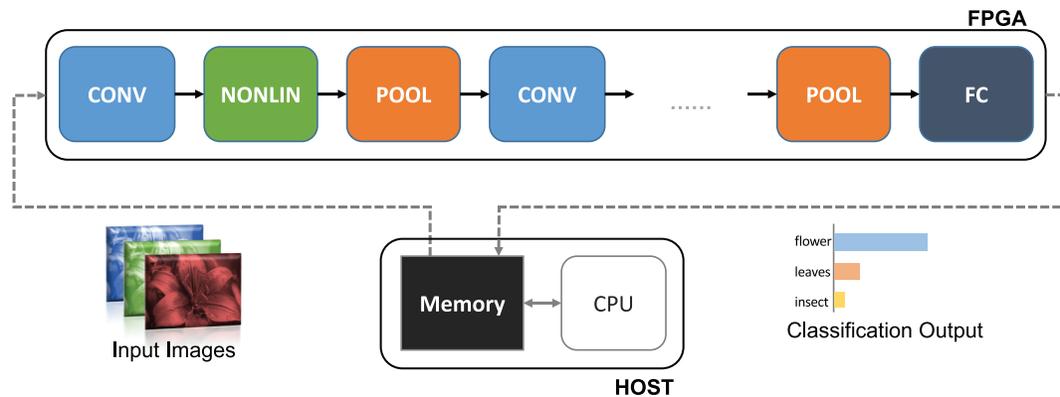
Specifications

- Latency
- Throughput
- Power consumption



Landscape / Approaches

| Toolflow Name | Interface | Year |
|--------------------------|--------------------------|----------------|
| fpgaConvNet [85–88] | Caffe & Torch | May 2016 |
| DeepBurning [90] | Caffe | June 2016 |
| Angel-Eye [23, 24, 68] | Caffe | July 2016 |
| ALAMO [55–59] | Caffe | August 2016 |
| HADDOC2 [1, 2] | Caffe | September 2016 |
| DNNWEAVER [75, 76] | Caffe | October 2016 |
| Caffeine [98] | Caffe | November 2016 |
| AutoCodeGen [54] | Proprietary Input Format | December 2016 |
| FINN [19, 84] | Theano | February 2017 |
| FP-DNN [22] | TensorFlow | May 2017 |
| Snowflake [10, 21] | Torch | May 2017 |
| SysArrayAccel [91] | C Program | June 2017 |
| FFTCCodeGen [95–97, 100] | Proprietary Input Format | December 2017 |



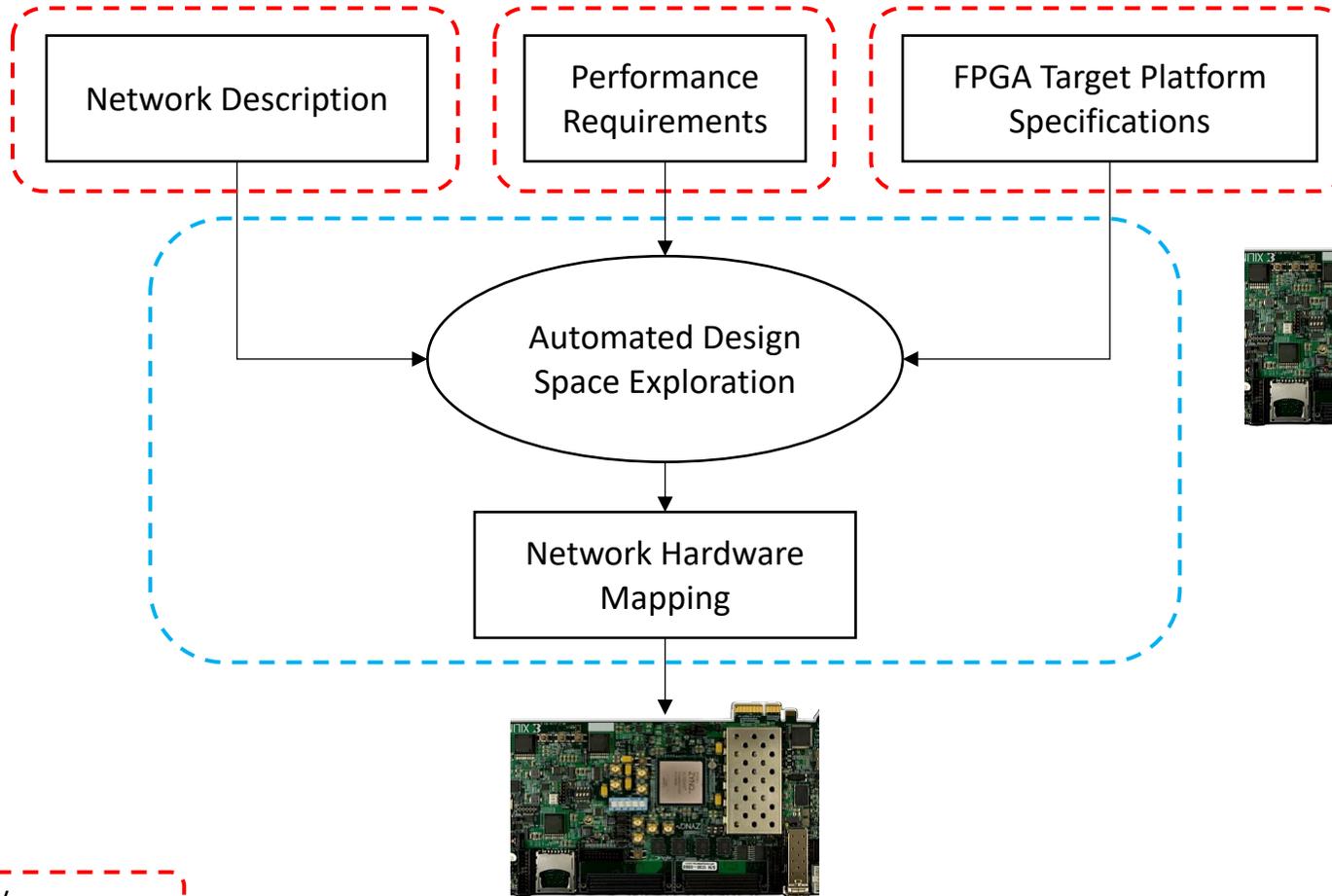
Tailored to workload



Tailored to operation

fpgaConvNet: Mapping CNNs to FPGAs

Caffe
torch



Supplied by
Deep Learning Expert

fpgaConvNet

2016 IEEE 24th Annual International Symposium on Field-Programmable Custom Computing Machines

fpgaConvNet: A Framework for Mapping Convolutional Neural Networks on FPGAs

Stylianos I. Venieris
Department of Electrical and Electronic Engineering
Imperial College London
Email: stylianos.venieris10@imperial.ac.uk

Christos-Savvas Bouganis
Department of Electrical and Electronic Engineering
Imperial College London
Email: christos-savvas.bouganis@imperial.ac.uk

Abstract—Convolutional Neural Networks (ConvNets) are a powerful Deep Learning model, providing state-of-the-art accuracy to many emerging classification problems. However, ConvNet classification is a computationally heavy task, suffering from rapid complexity scaling. This paper presents fpgaConvNet, a novel domain-specific modelling framework together with an automated design methodology for the mapping of ConvNets onto reconfigurable FPGA-based platforms. By interpreting ConvNet classification as a streaming application, the proposed framework employs the Synchronous Dataflow (SDF) model of computation as its basis and proposes a set of transformations on the SDF graph that explore the performance-resource design space, while taking into account platform-specific resource constraints. A

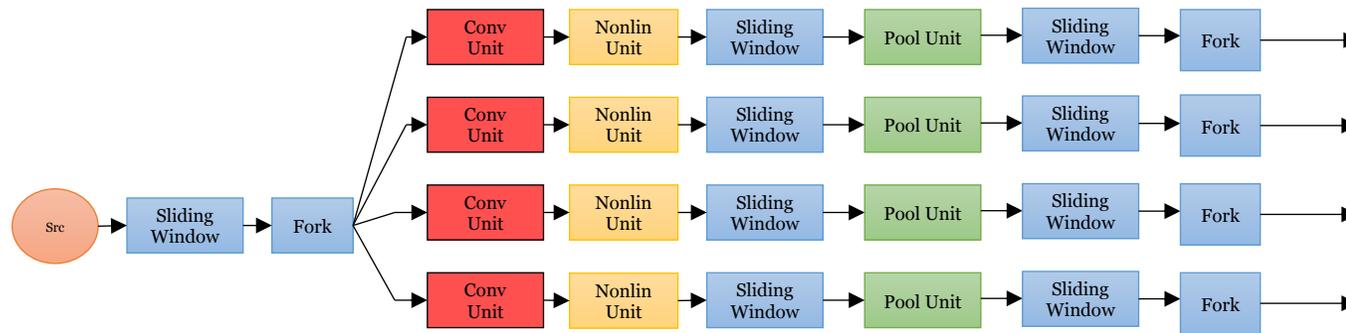
of a particular FPGA-based platform and guarantee portability and scalability. Portability would secure that a Deep Learning model implementation can be modified to operate on FPGA platforms with different characteristics. Scalability would ensure the ability to sustain or improve performance in case of an increase in the amount of available resources.

This work focuses on the Design Space Exploration (DSE) for the classification task of the Deep Learning model of Convolutional Neural Networks (ConvNets) mapped onto reconfigurable FPGA-based platforms by means of a domain-specific modelling framework. The proposed methodology aims to provide the infrastructure and the analytical tools that

Proc. IEEE Symposium on Field-Programmable Custom Computing Machines, 2016

fpgaConvNet – Streaming Architecture for CNNs

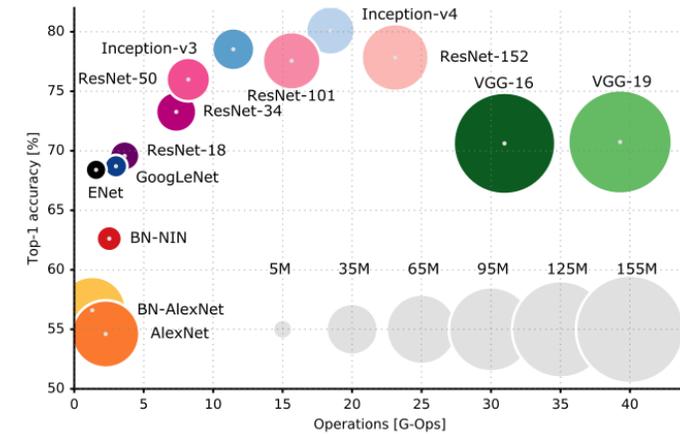
CNN Hardware SDF Graph



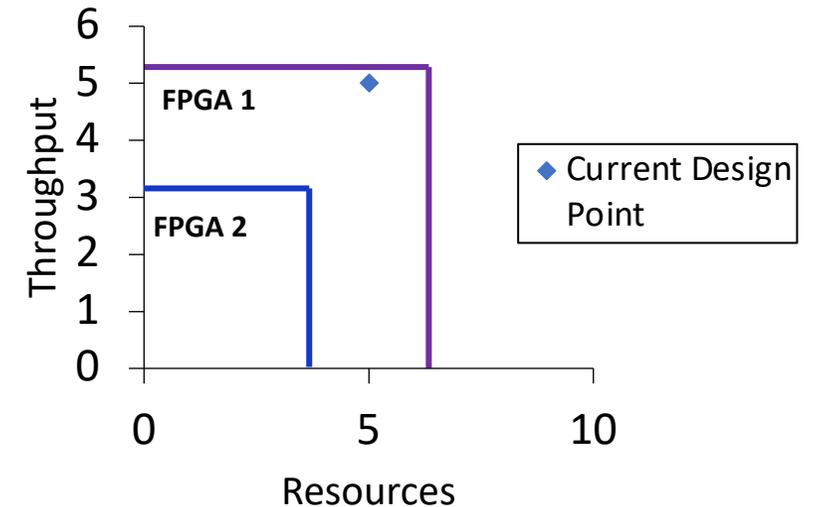
Complex Model → Bottlenecks:

- Limited *compute resources*
- Limited *on-chip memory capacity* for model parameters
- Limited *off-chip memory bandwidth*

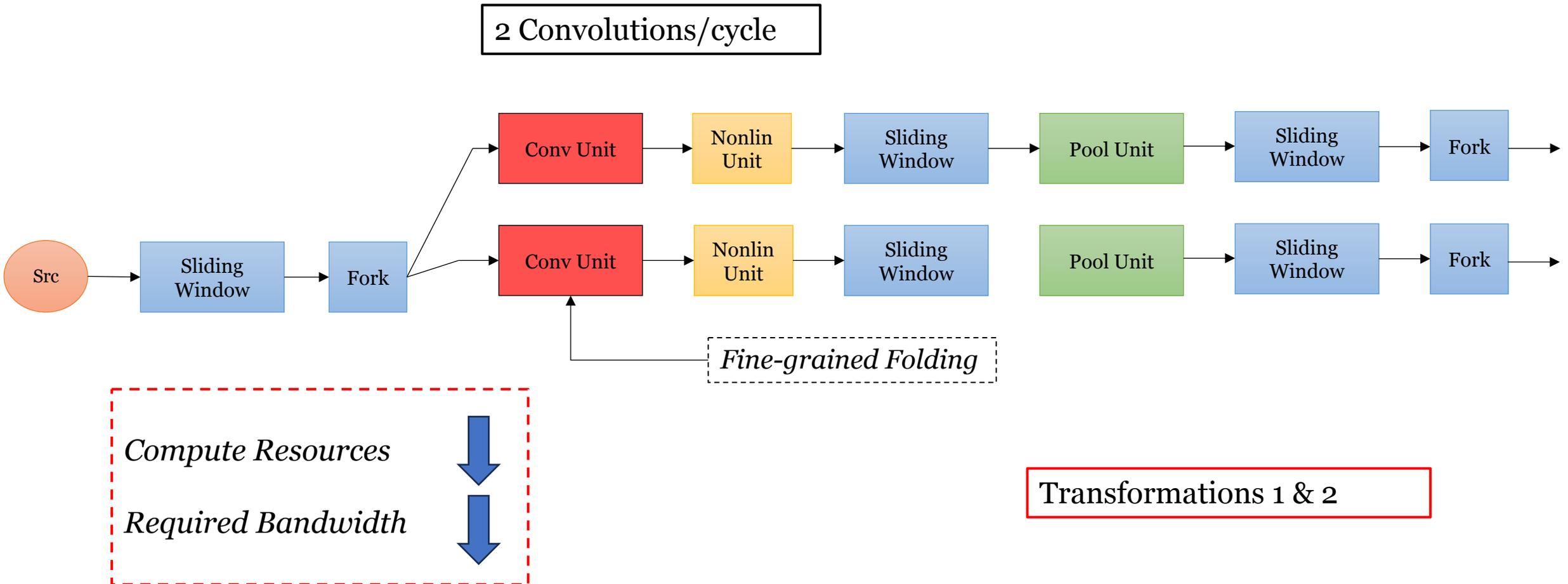
Define a set of **graph transformations** to traverse the design space in **fast** and **principled** way



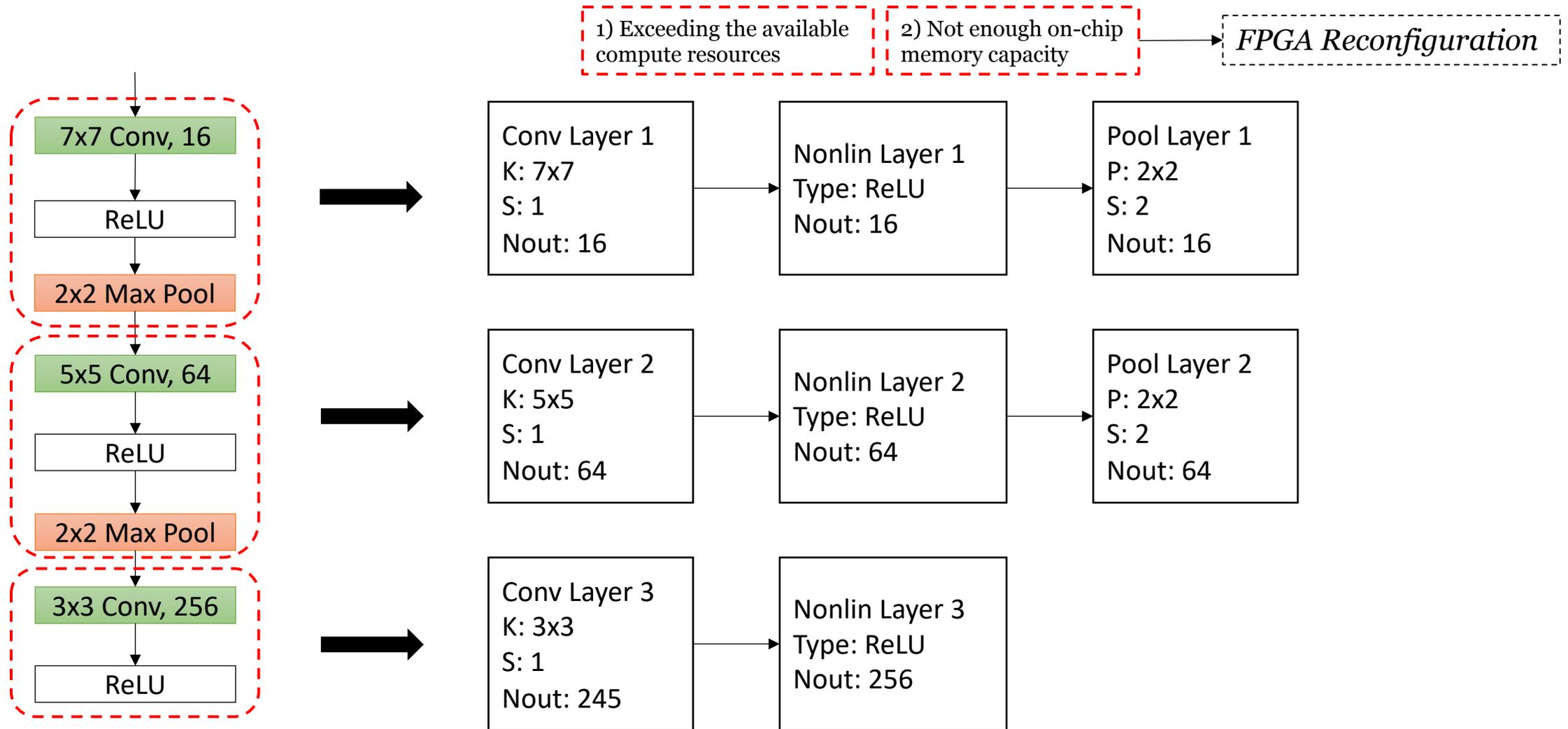
Design Space



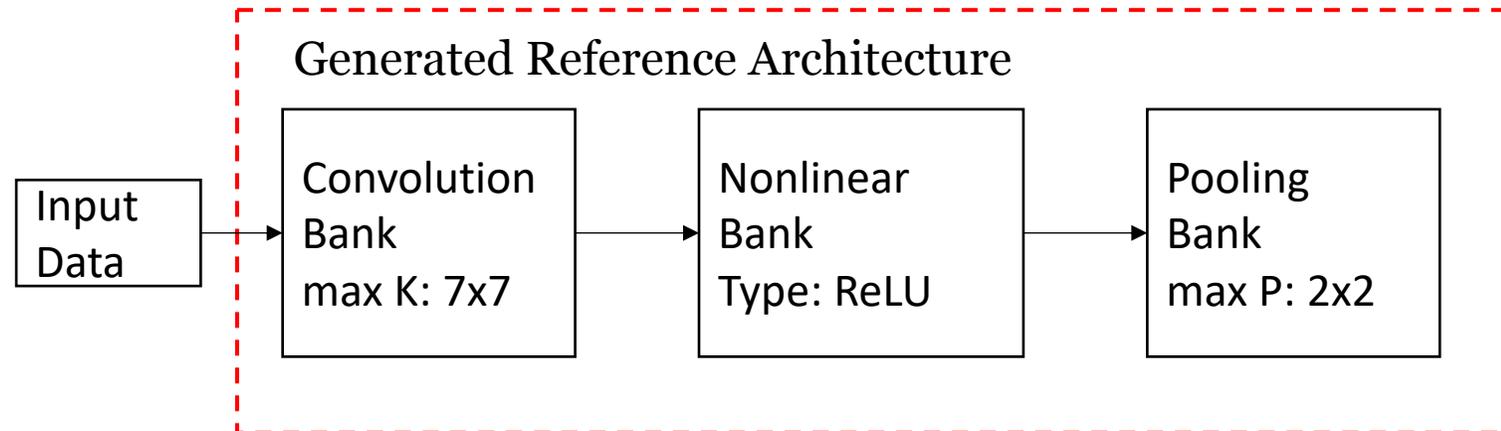
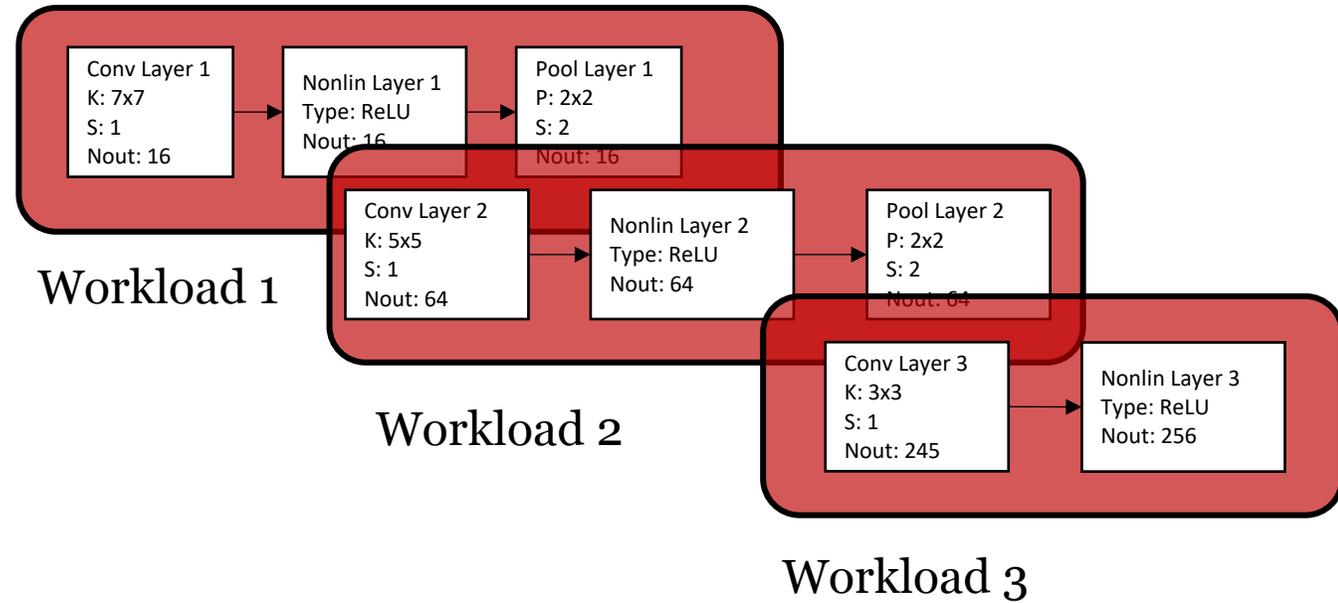
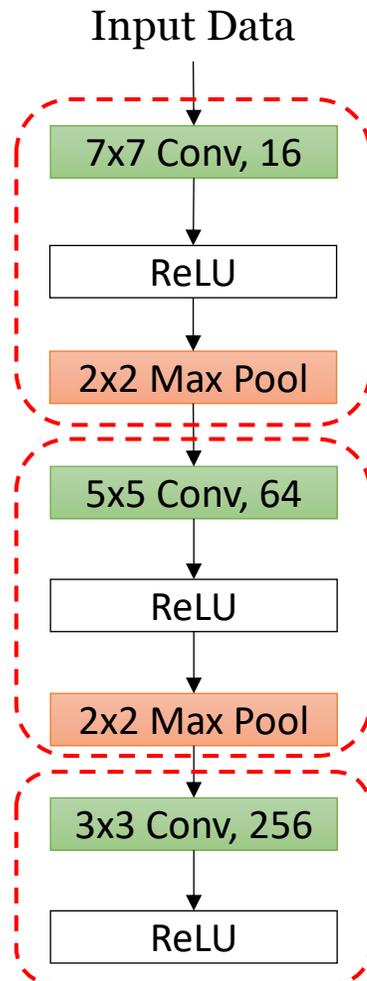
Transformations 1 & 2: Coarse- and fine-grained Folding



Transformation 3: Graph Partitioning with Reconfiguration

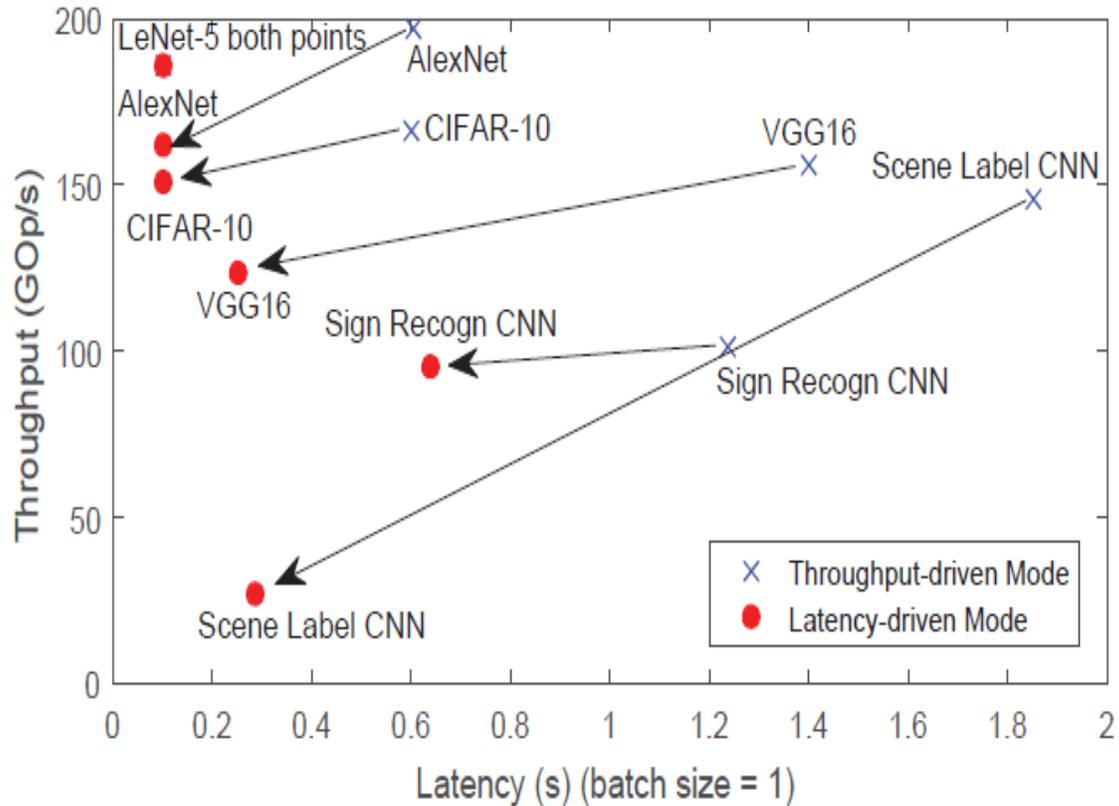


Transformation 4: Weights Reloading

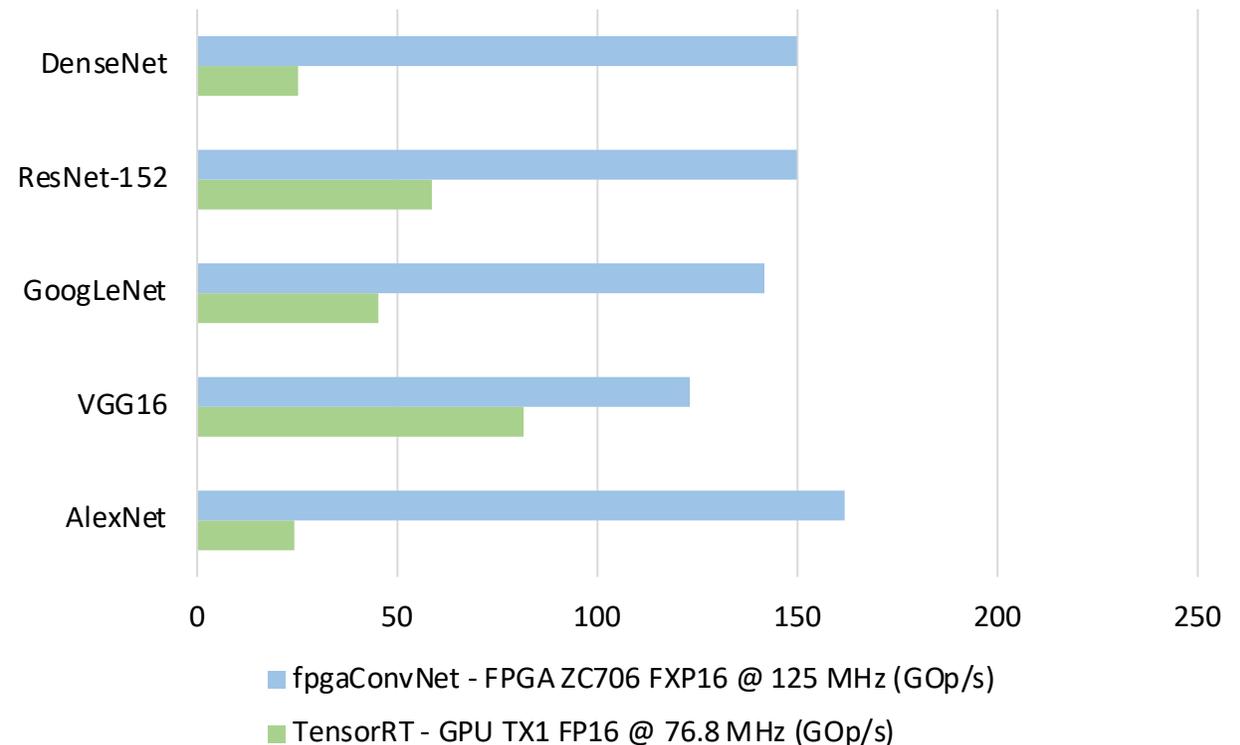


Meeting the performance requirements

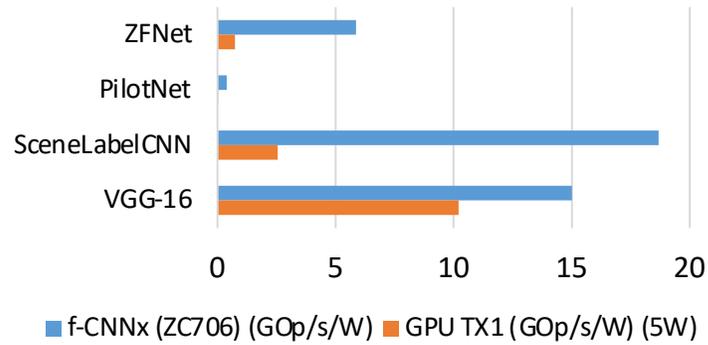
Latency – Throughput trade-off



Latency-driven scenario, power target 5W
Up to 6.65× speedup with an average of 3.95×

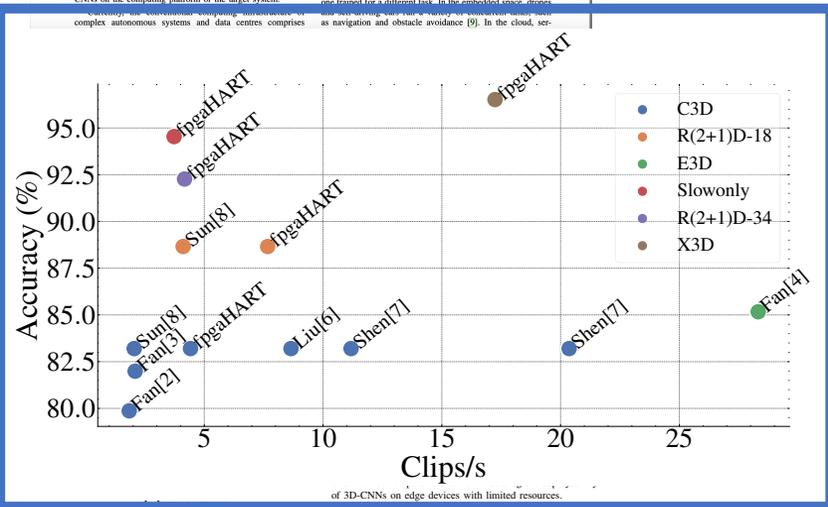
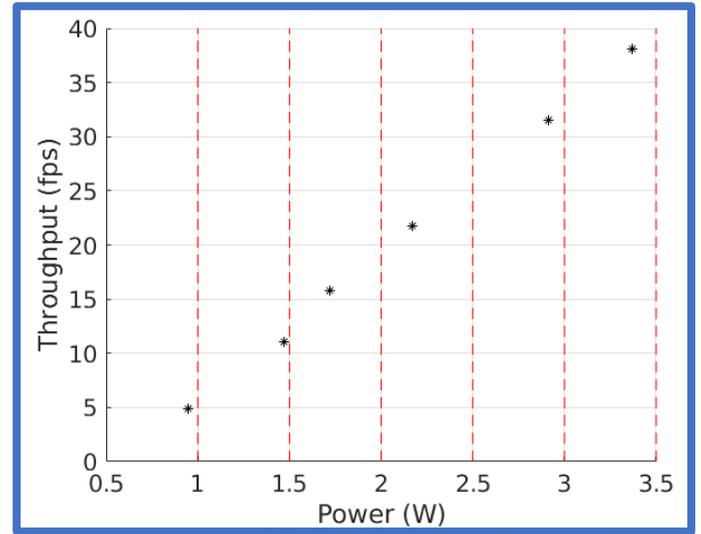
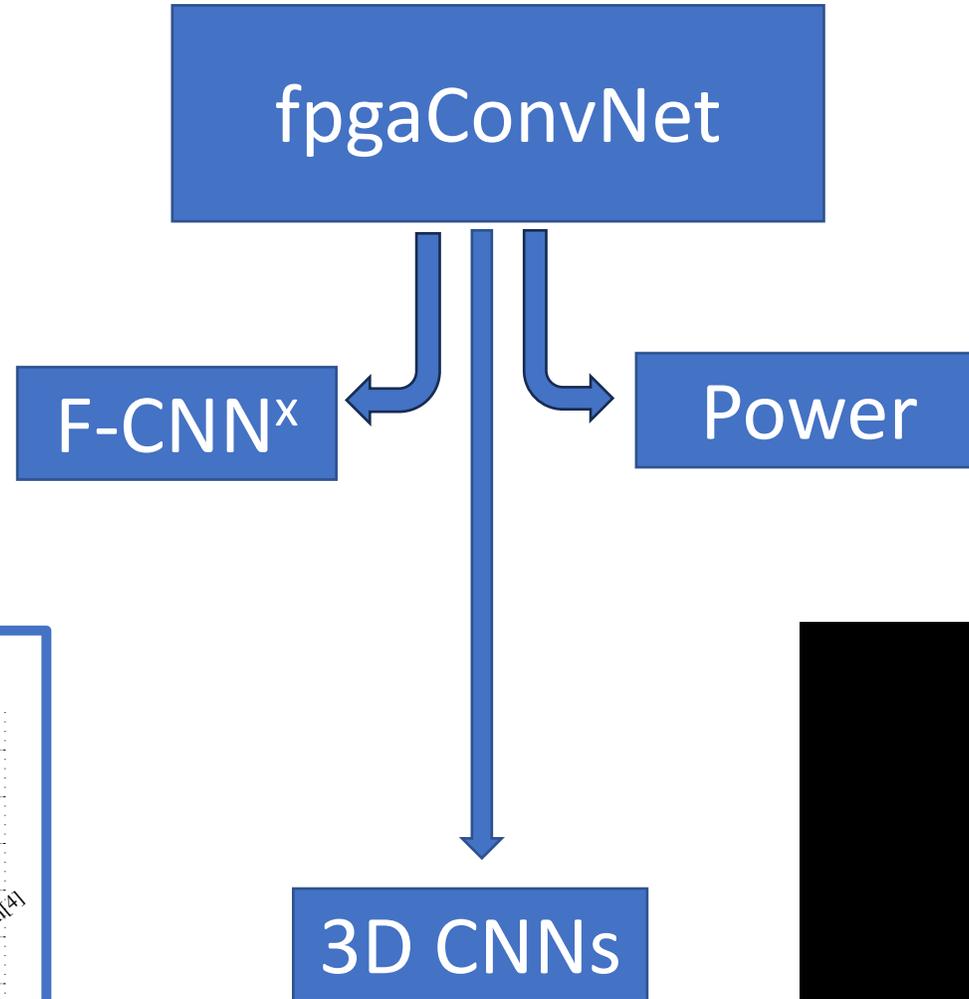


Performance-per-Watt: f-CNN^x vs. TX1
at 5W



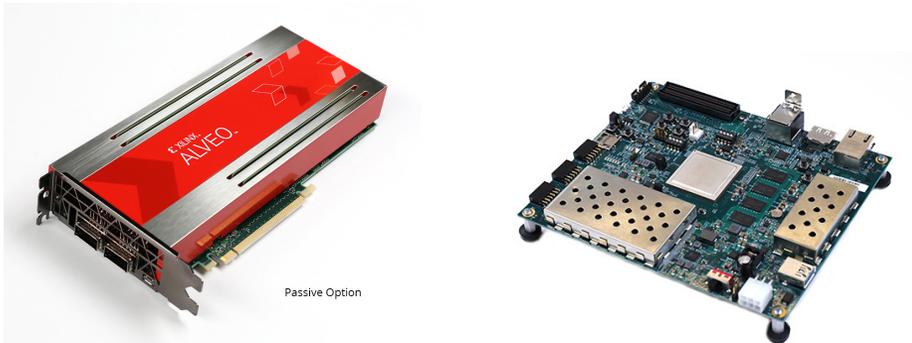
same target platform. Moreover, the latency-sensitive nature of modern applications prohibits the use of batch processing. As a result, in both emerging embedded and cloud applications there is a requirement for the latency-driven mapping of multiple CNNs on the computing platform of the target system.

II. MULTIPLE CNNs ON RECONFIGURABLE LOGIC
A. Background on Multi-CNN Systems
Multi-CNN systems employ a number of models, with each one trained for a different task. In the embedded case, devices are often required to run multiple CNNs on a single platform, as navigation and obstacle avoidance [9]. In the cloud, ser-



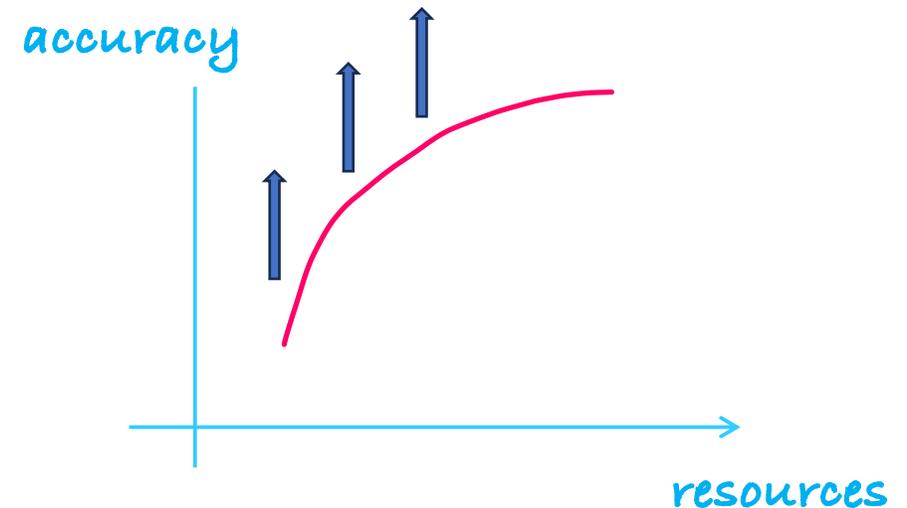
Embedded Space: Hardware Limitations and Challenges

- Small devices



| | U250 | ZCU104 |
|-----------------|------|---------|
| INT8 TOPs | 33.3 | 1.2 |
| SRAM memory | 54MB | 4.75MB |
| Off-chip memory | 64GB | Limited |

- Approximations for performance



Limited resources
DSPs

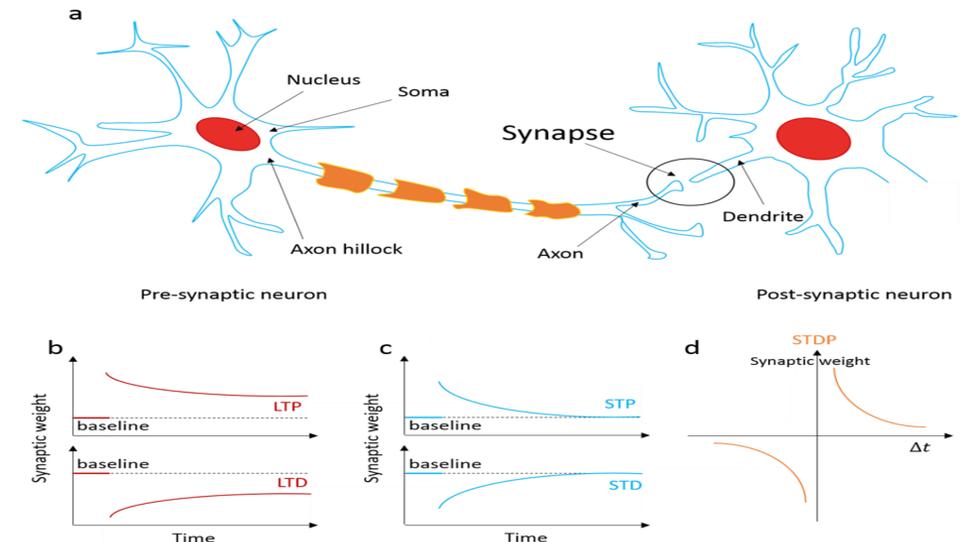
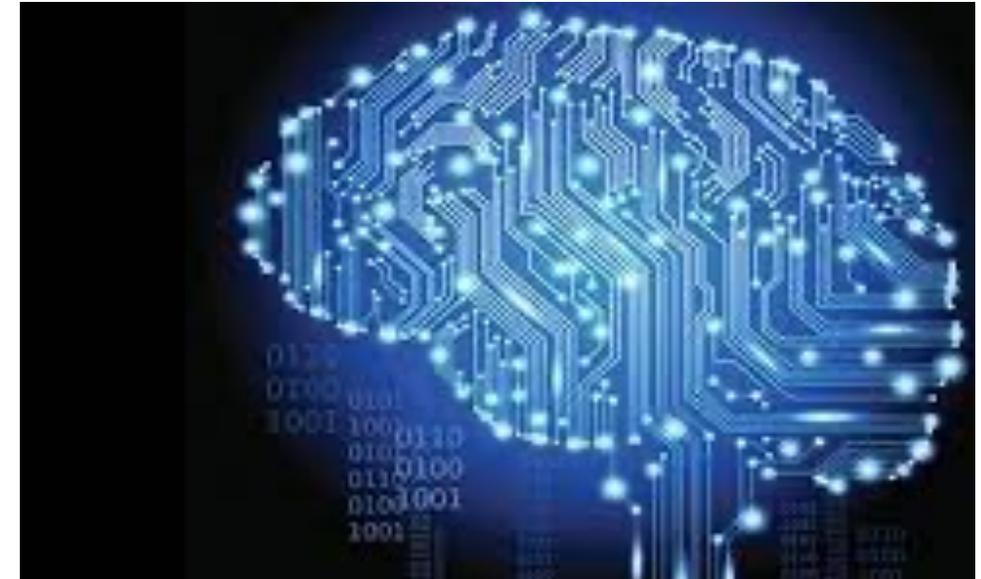
Targeting compute resources

Neuromorphic Computing

- Event driven computation => Explore Sparsity

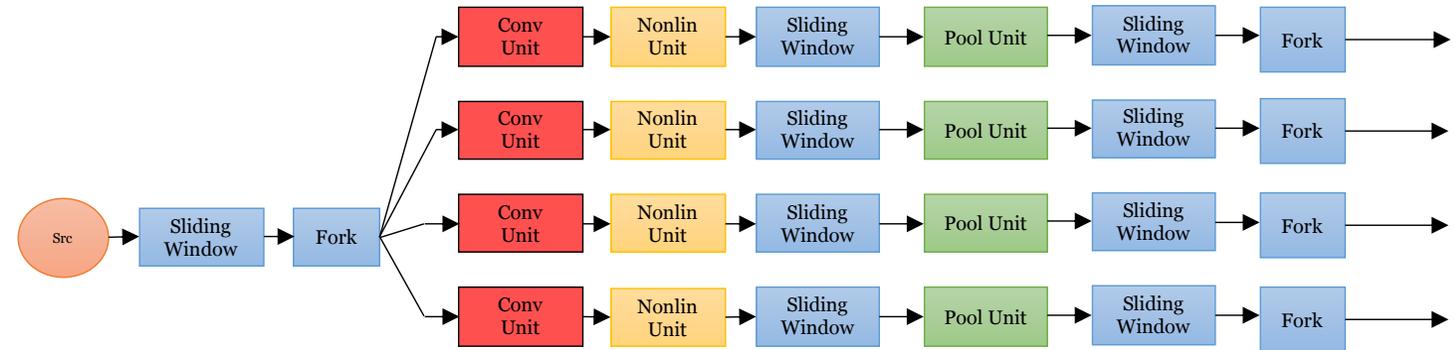
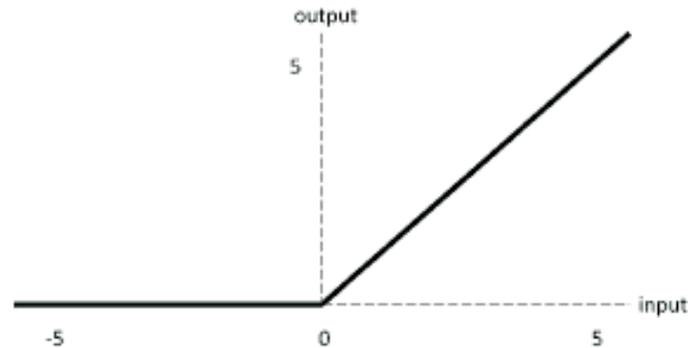
Activation
Sparsity

Weights
Sparsity



Post-activation sparsity

ReLU activation function



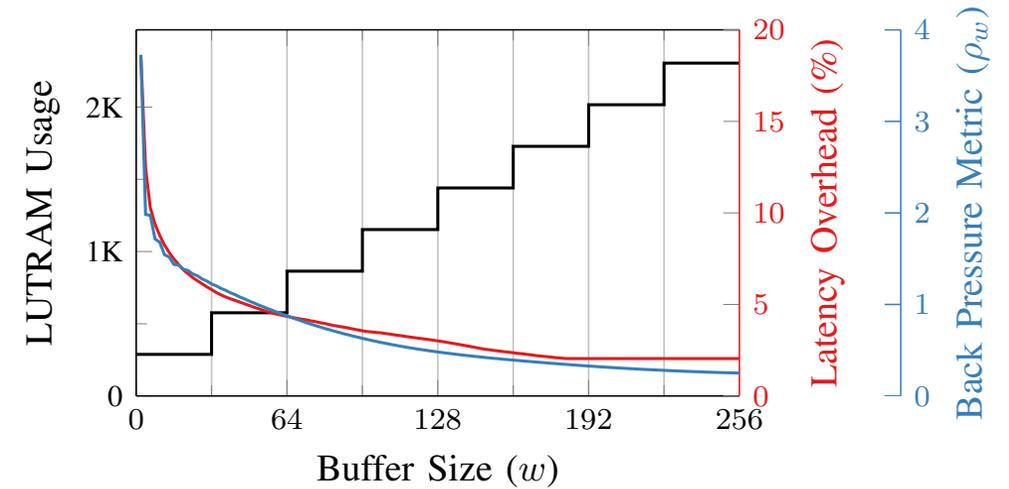
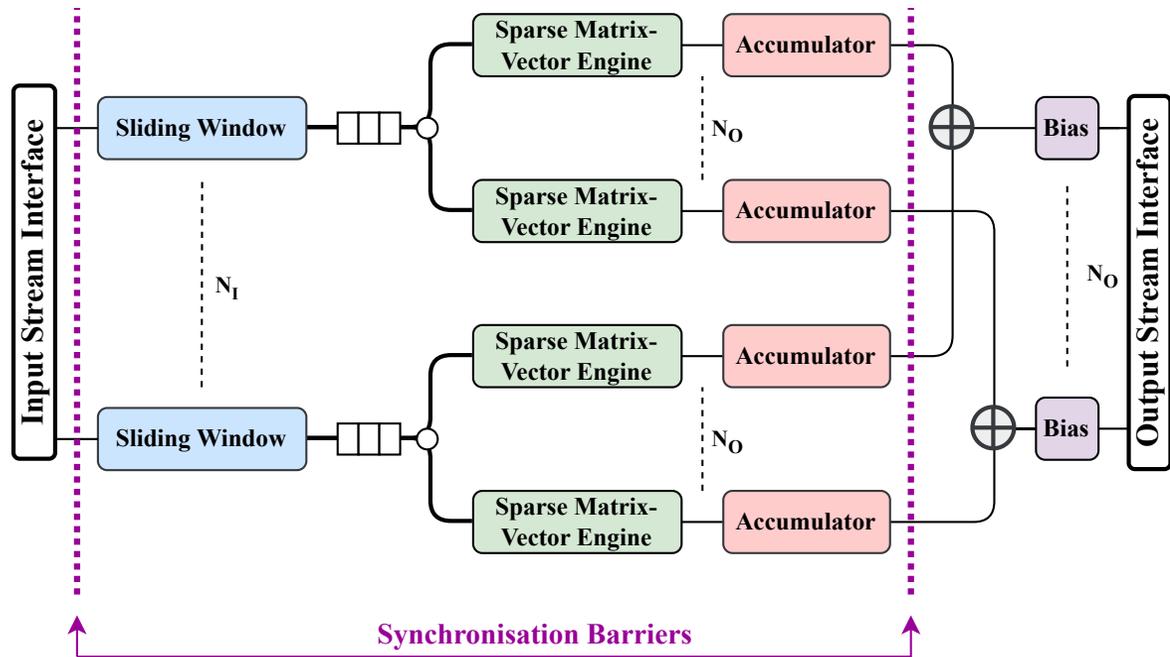
Opportunities

- Many CNNs produces sparse activations
- ImageNet validation set:
 - VGG16: 65% sparsity
 - ResNet-18: 57% sparsity
- Reduce the resources allocated per compute node
- Tune the architecture to specific sparsity pattern
 - CNN specific
 - Layer specific (Localized)

Challenges

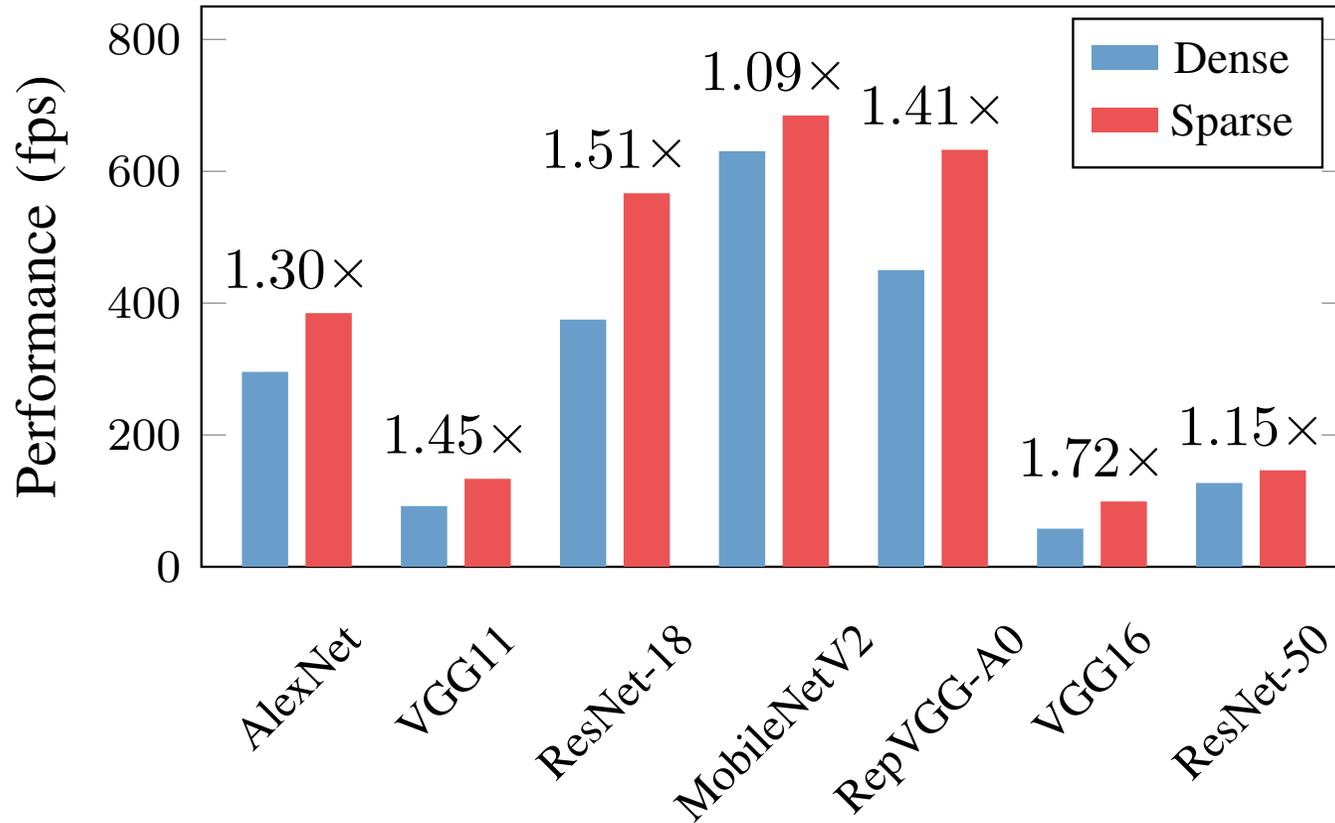
- **Statistical information available only**
- Streaming architecture
- Dynamic scheduling
- Data Stream synchronization
- Resource allocation (Global optimum)

Conv layer with Sparse Matrix-Vector Engines



Tuning:

- Statistical information gathering
- Resource allocation per engine based on extracted information
- Buffer size to alleviate back pressure



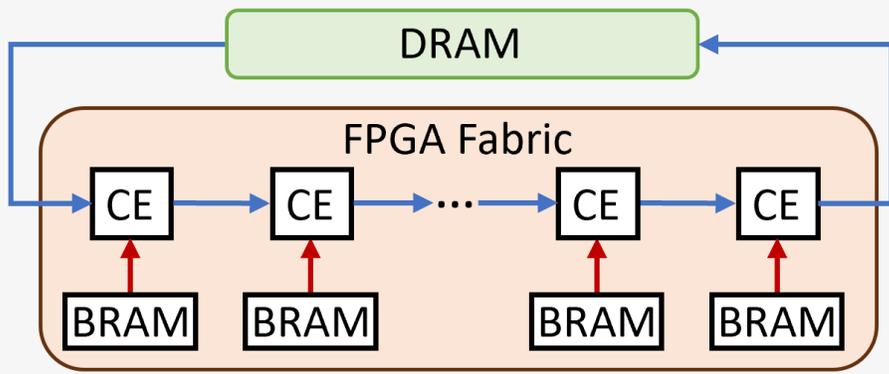
VGG16 – 3rd Conv layer

| Design | LUT | FF | BRAM | DSP | Freq. (MHz) | Lat. (ms) |
|---------------|-------------|-------------|-------------|-------------|-------------|-------------|
| <i>Dense</i> | 26,046 | 41,211 | 272 | 192 | 223 | 44.5 |
| <i>Sparse</i> | 38,112 | 48,895 | 272 | 192 | 200 | 17.8 |
| | 1.5x | 1.2x | 1.0x | 1.0x | 0.9x | 0.4x |

Limited resources
on-chip memory

Low-power devices: Weights Streaming

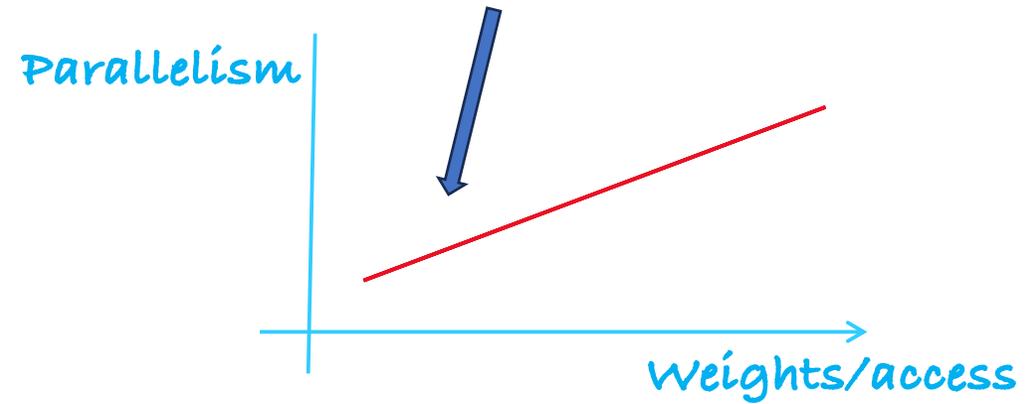
Latency oriented applications
Targeting limited on-chip memory



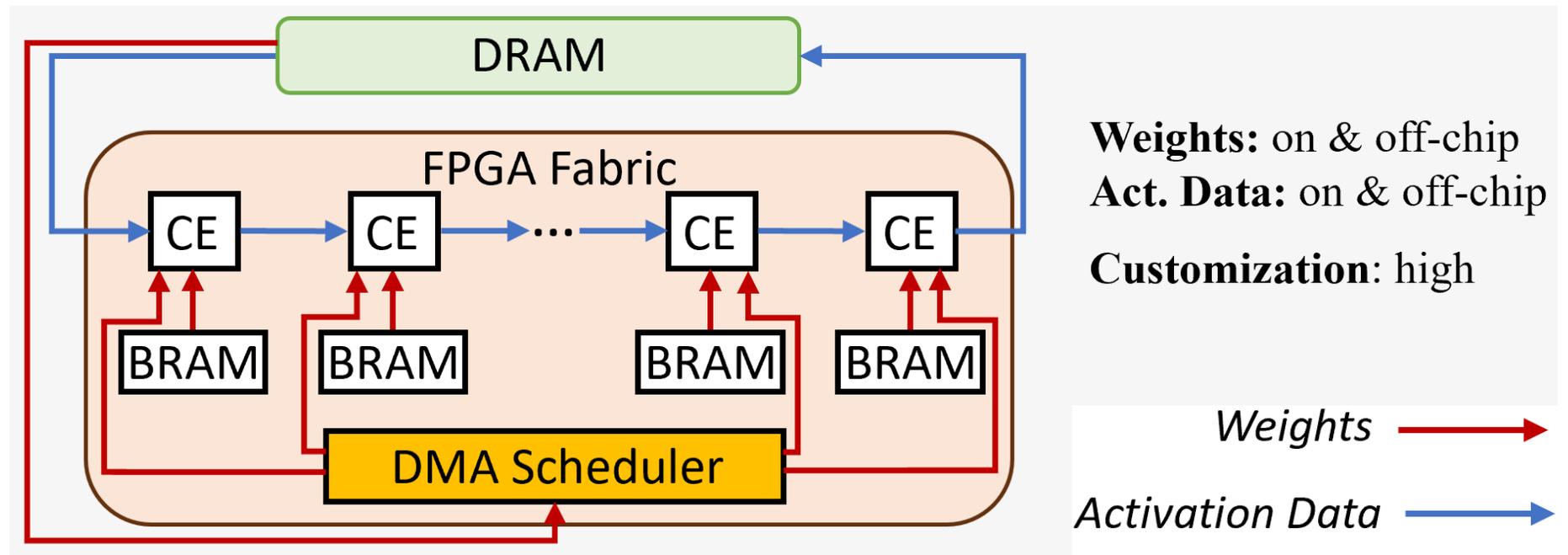
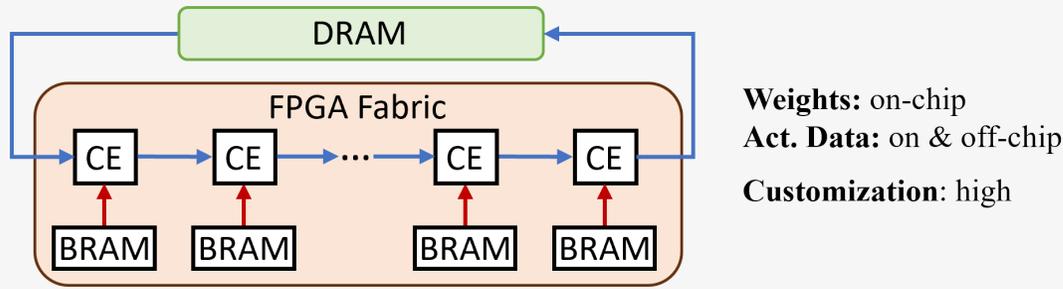
Weights: on-chip
Act. Data: on & off-chip
Customization: high

Weights →

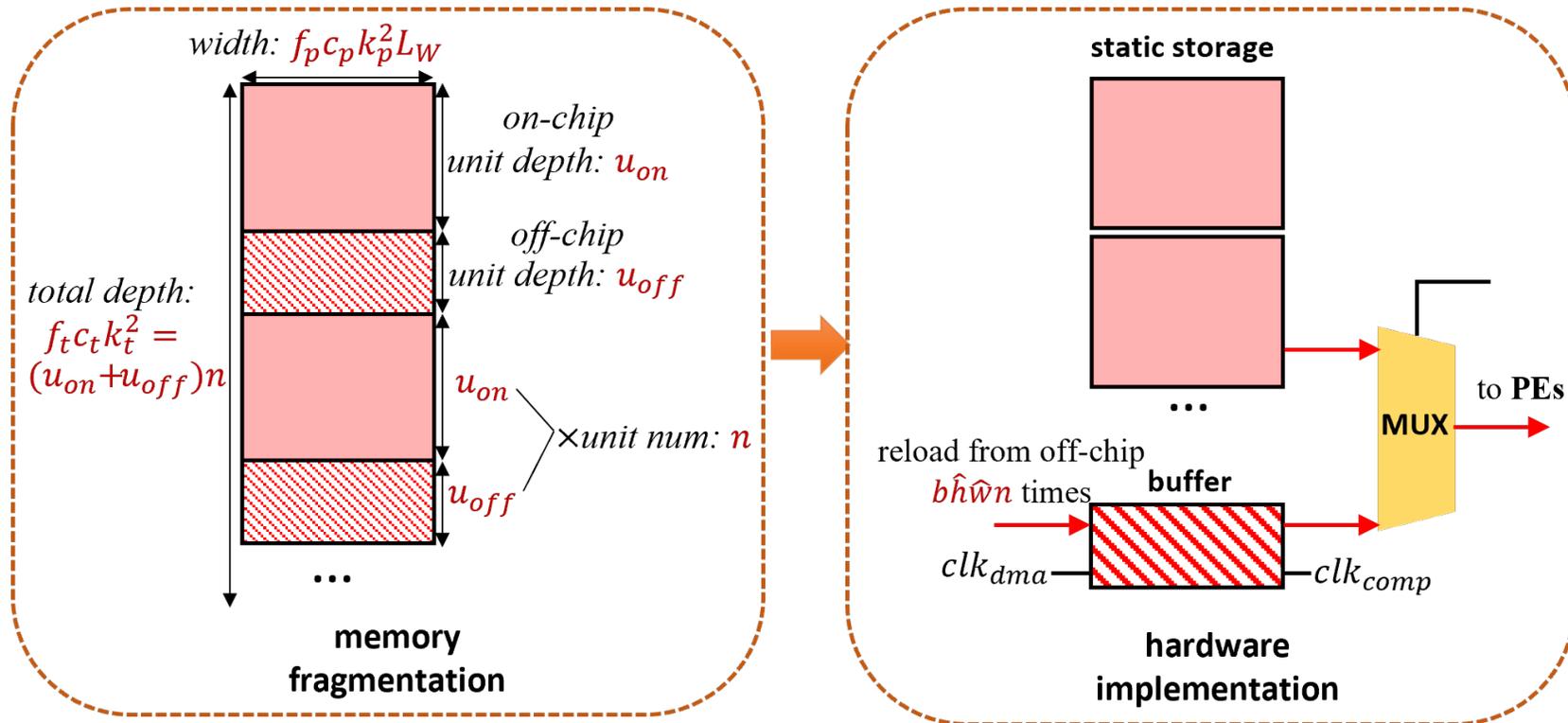
Activation Data →



Low-power devices: Weights Streaming



Low-power devices: Weights Streaming

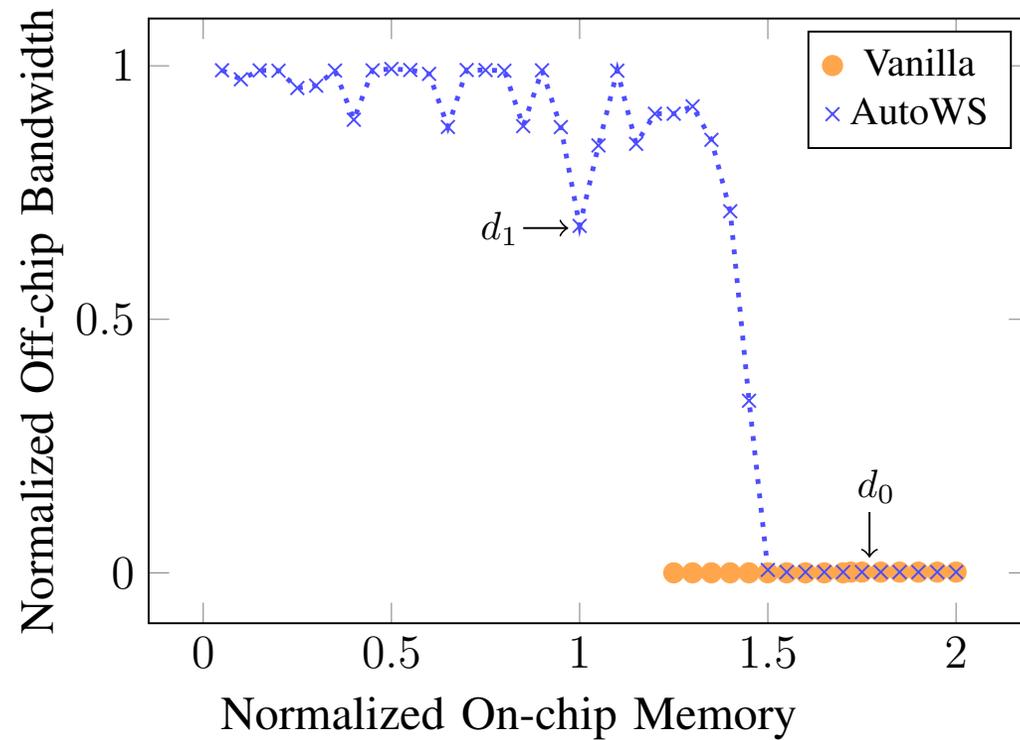
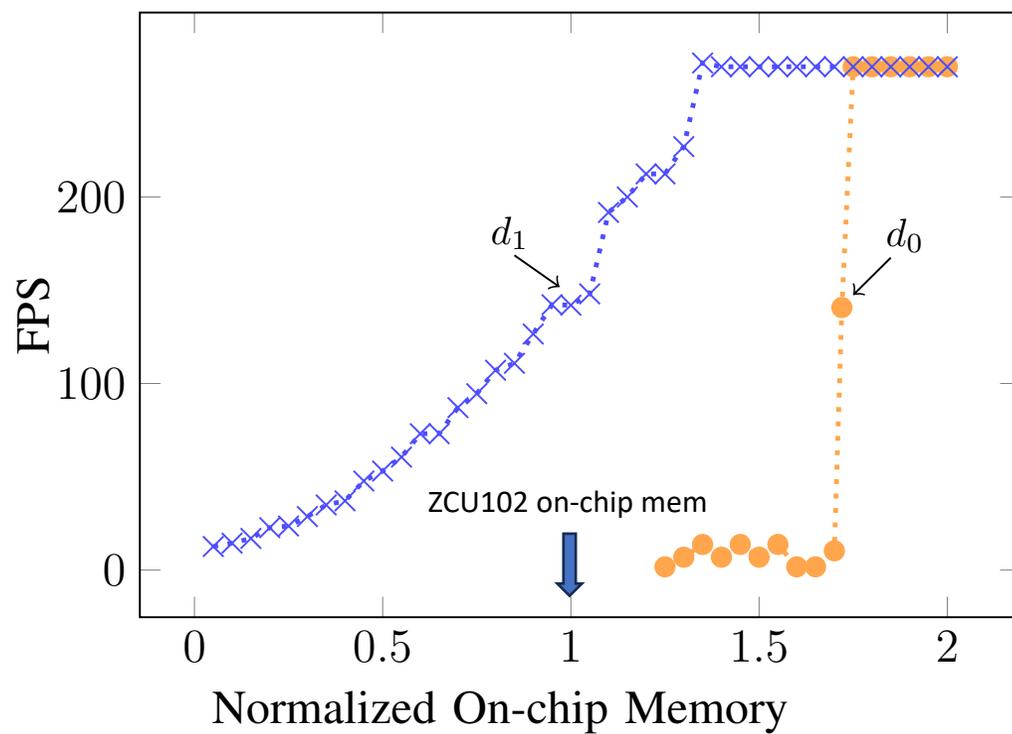


Symbols Definitions

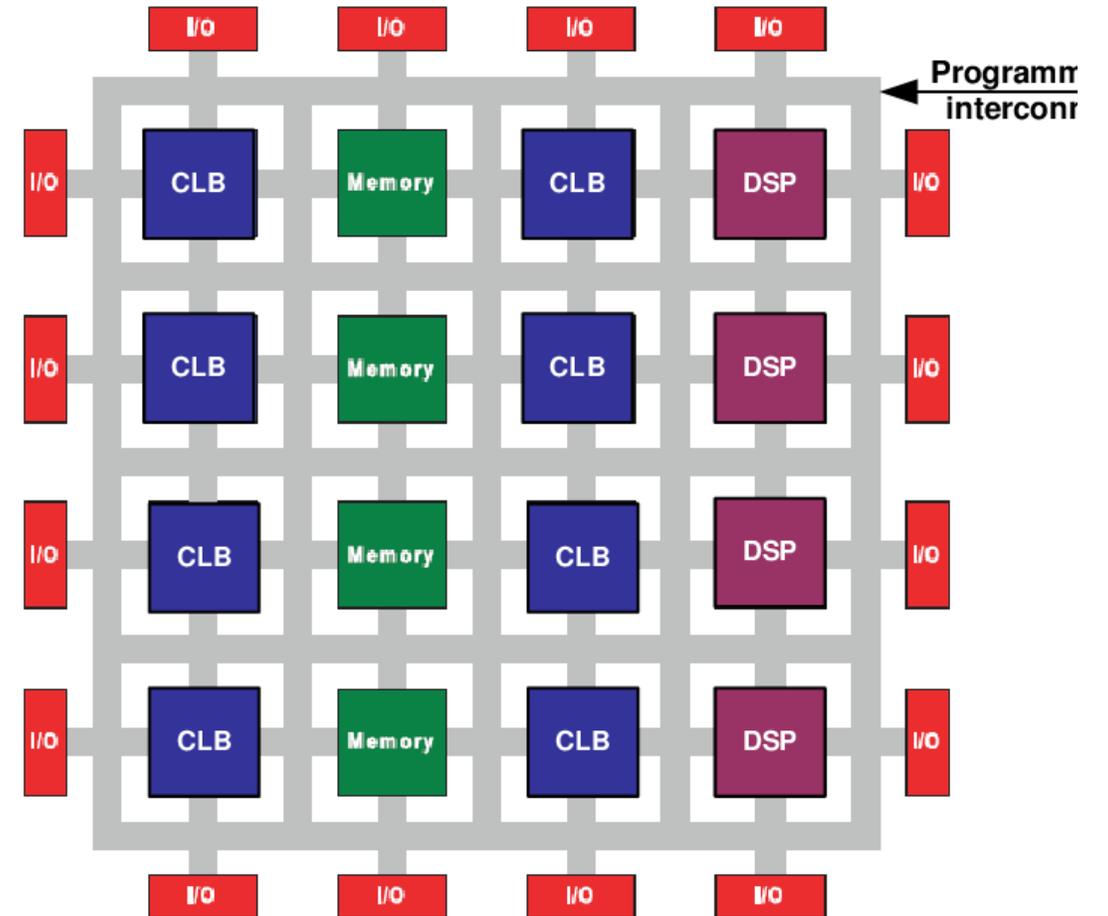
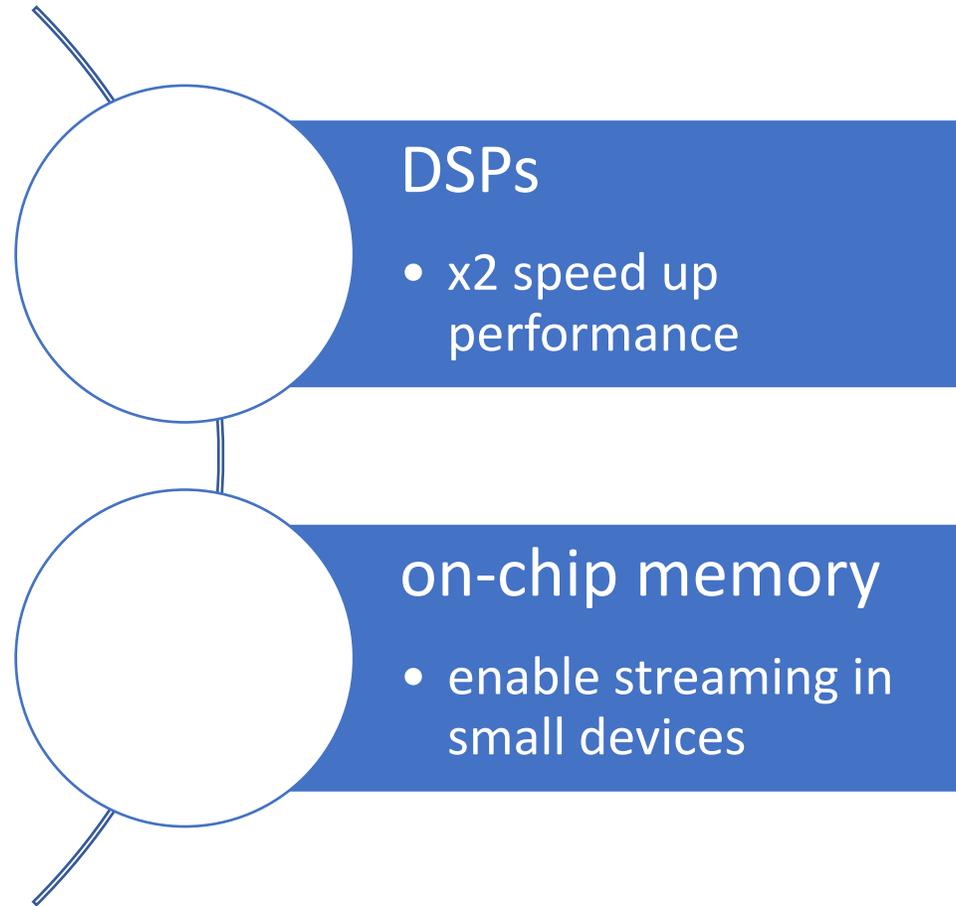
| Symbols | Definitions |
|--------------------|----------------------|
| b | batch size |
| c | input channel number |
| h, w | input height/width |
| k | kernel size |
| \hat{h}, \hat{w} | output height/width |
| f | filter number |
| L_W | weights bitwidth |
| L_A | activations bitwidth |

Low-power devices: Weights Streaming

Model: ResNet-18
Device: ZCU102



Low-power devices: Limited resources



Challenge 2

(unachievable) Performance

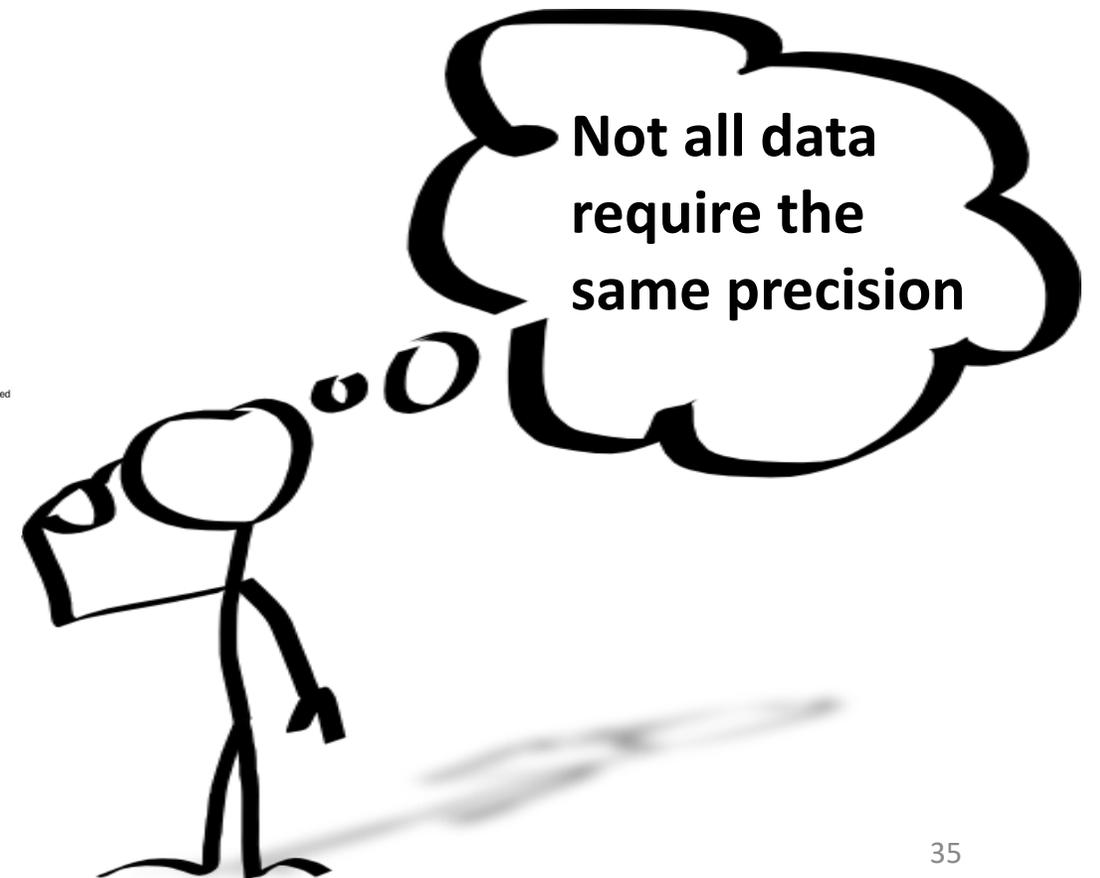
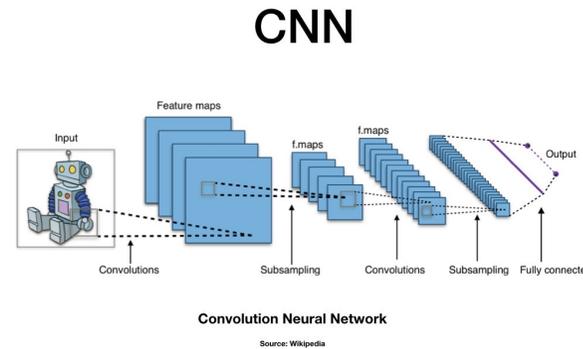
Approximations in DNN - Overview



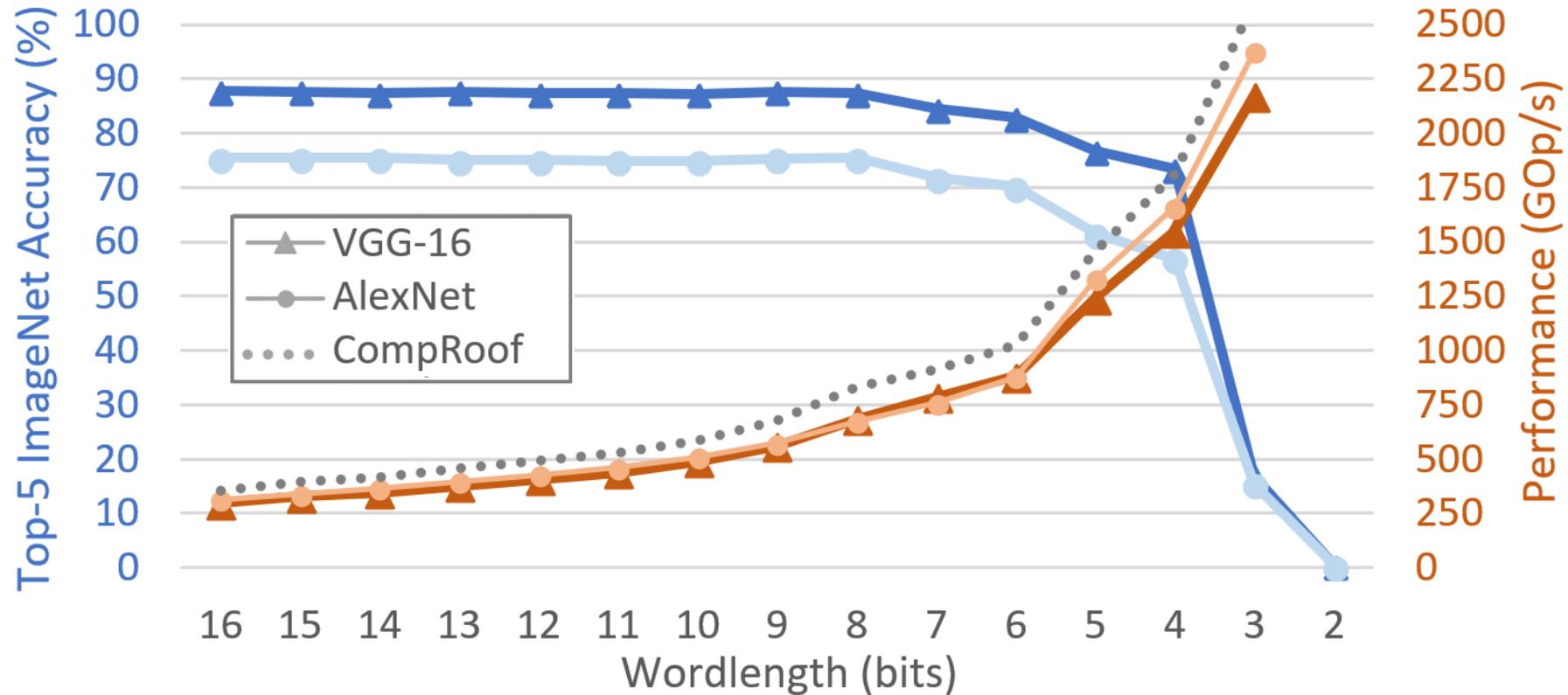
| CNN architecture | Compression Approach | Data Type | Original → Compressed Model Size | Reduction in Model Size vs. AlexNet | Top-1 ImageNet Accuracy | Top-5 ImageNet Accuracy |
|-------------------|--------------------------------------|-----------|----------------------------------|-------------------------------------|-------------------------|-------------------------|
| AlexNet | None (baseline) | 32 bit | 240MB | 1x | 57.2% | 80.3% |
| AlexNet | SVD (Denton et al., 2014) | 32 bit | 240MB → 48MB | 5x | 56.0% | 79.4% |
| AlexNet | Network Pruning (Han et al., 2015b) | 32 bit | 240MB → 27MB | 9x | 57.2% | 80.3% |
| AlexNet | Deep Compression (Han et al., 2015a) | 5-8 bit | 240MB → 6.9MB | 35x | 57.2% | 80.3% |
| SqueezeNet (ours) | None | 32 bit | 4.8MB | 50x | 57.5% | 80.3% |
| SqueezeNet (ours) | Deep Compression | 8 bit | 4.8MB → 0.66MB | 363x | 57.5% | 80.3% |
| SqueezeNet (ours) | Deep Compression | 6 bit | 4.8MB → 0.47MB | 510x | 57.5% | 80.3% |

92.4% (OmniVec)

Given a target FPGA board and a CNN model, design a system, optimising throughput/latency, without access to training data



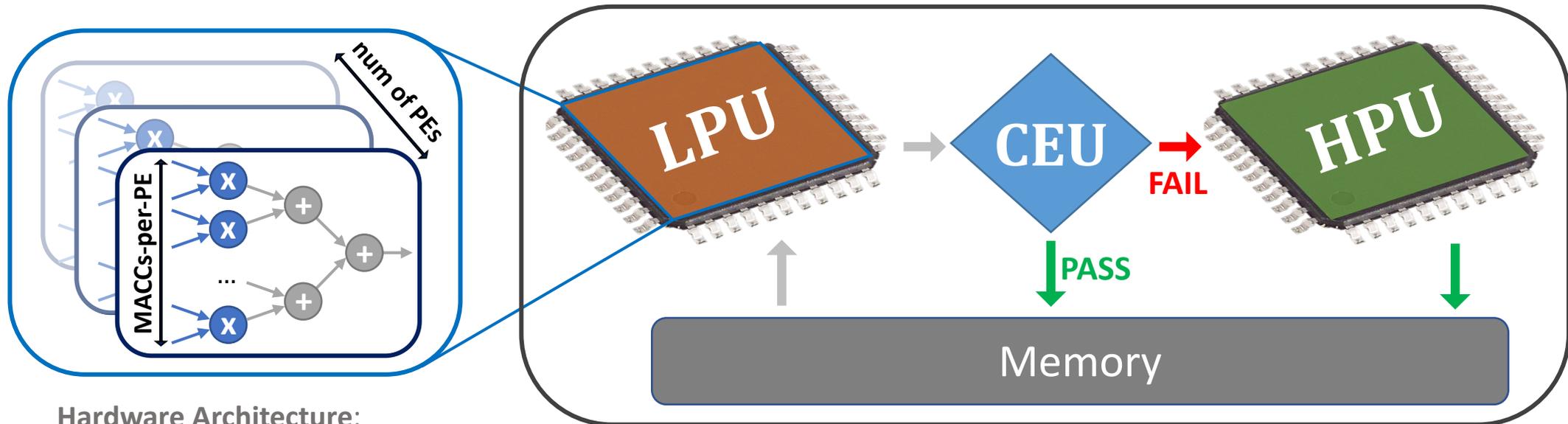
Approximations in DNN – Post-training



CascadeCNN Overview

Exploit the fact that not all inputs require the same level of precision to obtain a confident prediction

- **Key idea: Generate a Cascade of CNN Classifiers, tailored to CNN-FPGA pair**



Hardware Architecture:

Tailored for any given CNN_FPGA pair, optimized for high-throughput inference

Low-Precision Unit:

Degraded accuracy classification with high performance

Confidence

Evaluation Unit:

Identify misclassified cases

High-Precision Unit:

Correct detected misclassified samples, to restore accuracy

Prediction Confidence Evaluation

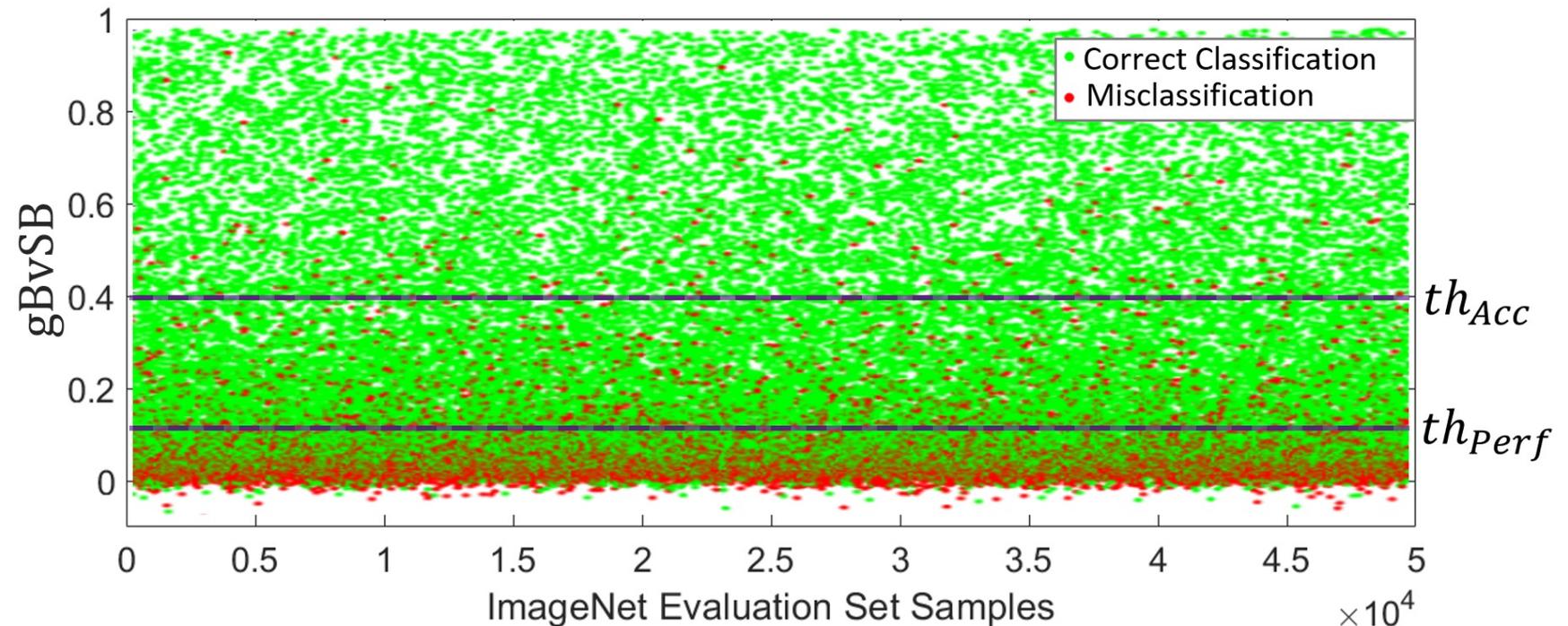
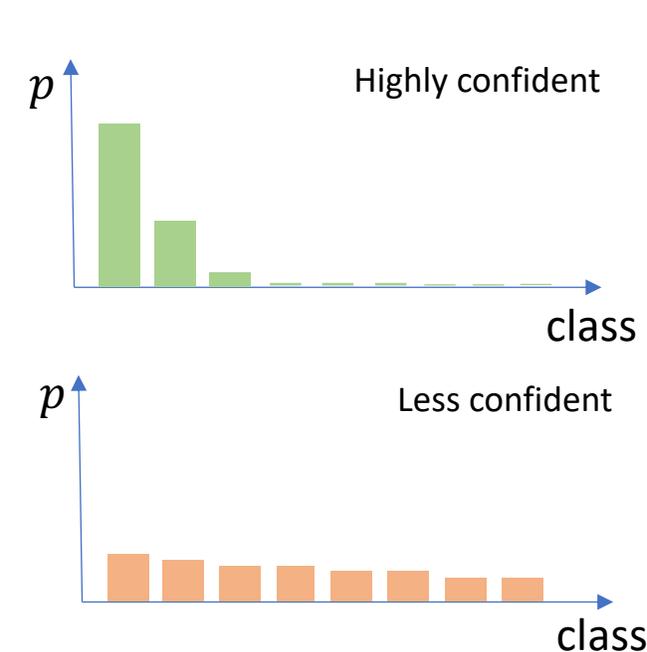
Confidence Estimation of LPU classifications prediction, at run-time.

generalised Best-vs-Second-Best:

$$gBvSB_{\langle M, N \rangle}(\mathbf{p}) = \sum_{i=1}^M p_i - \sum_{j=M+1}^N p_j$$

Evaluator:

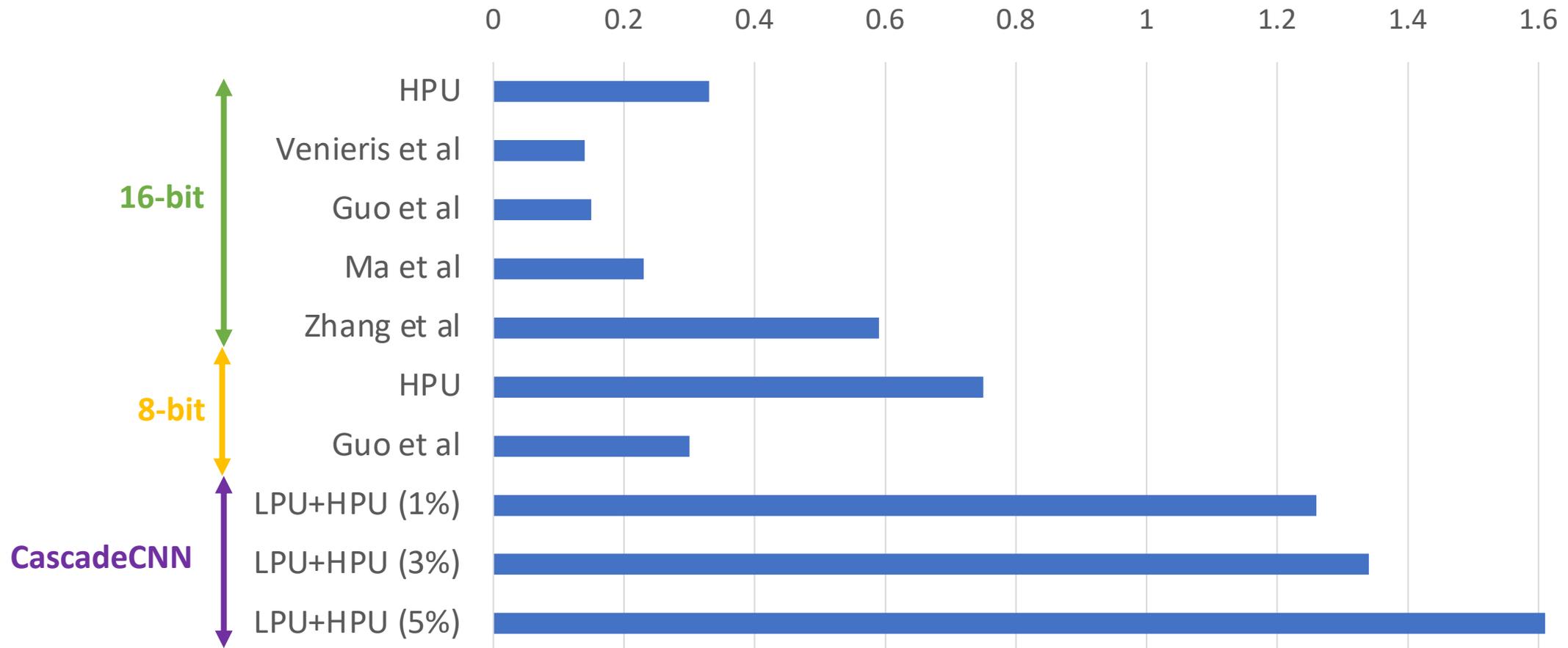
$$gBvSB_{\langle M, N \rangle}(\mathbf{p}) \geq th$$



Comparison with Existing FPGA Work

VGG-16

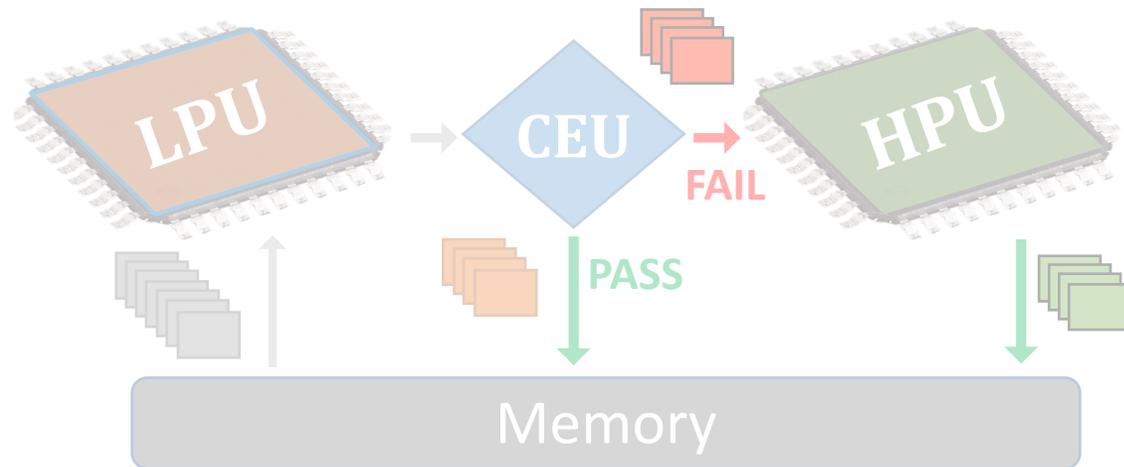
Normalised Throughput (GOps/s/DSP)



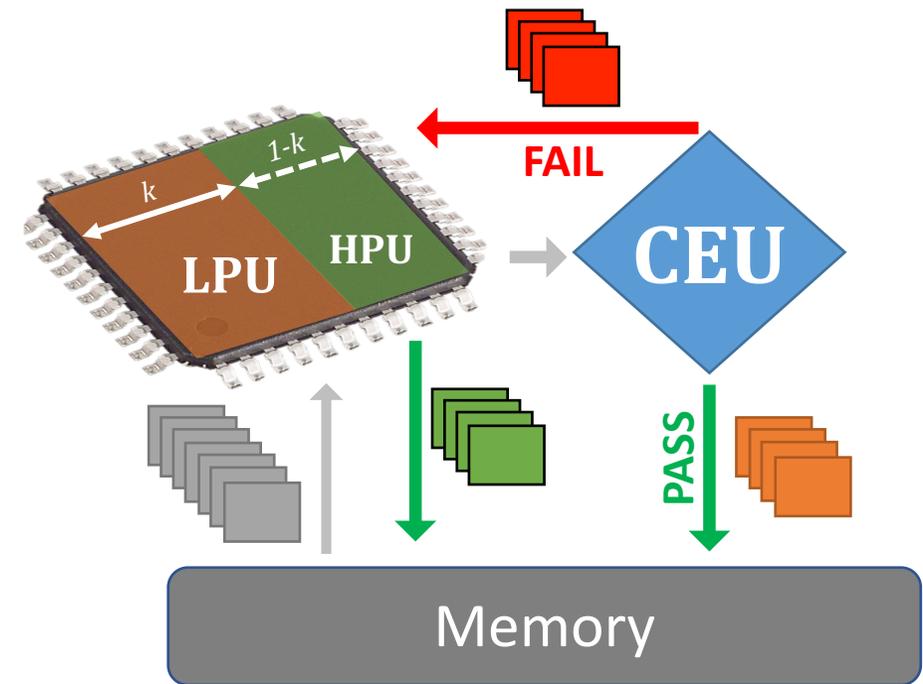
A Latency-Driven CNN Cascade Architecture

- Sharing the device resources eliminates the need for reconfiguration and batching.

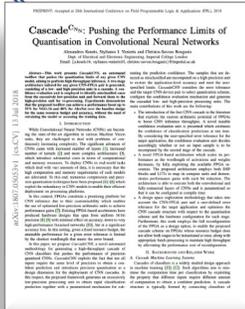
Throughput-Optimised



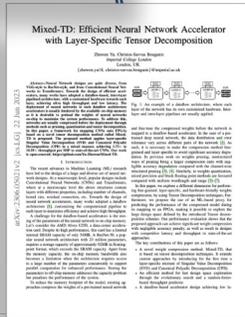
Multi-objective



Low-power devices: Approximations

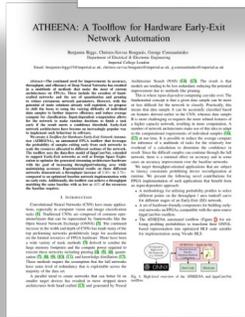


Cascade CNN



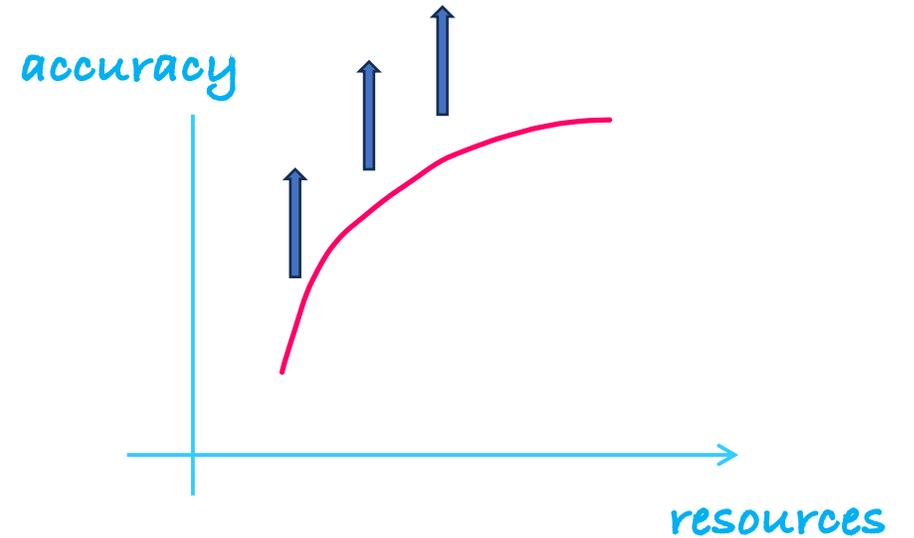
Tensor Decomposition

- Mixed-TD



Early Exit Network

- ATHEENA

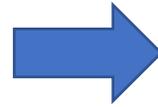


Are we done?

- More models:
 - Transformers
 - Diffusion models
 - Consistency models
- More applications:
 - on-device training
 - multiple DNNs
- New technology
 - in-memory compute



customisation



...but expensive

We need investment in methods to support deployment of DNNs on the embedded space.

Intelligent Digital Systems Lab (IDSL)

Home About us Research Group members Publications Work with us Contact

Welcome to the Intelligent Digital Systems Lab at Imperial College

TOP LINKS

Our research

Dr. Christos Bouganis

Join our lab

CNN-to-FPGA Benchmark Suite

fpgaConvNet



The IDSL lab is part of the Electrical and Electronic Engineering Department of Imperial College London.

1 of 11

Cascade^{CNN}: Pushing the performance limits of quantisation

Alexandros Kouris Dept. of Electrical and Electronic Eng. Imperial College London a.kouris16@ic.ac.uk
 Stylianos I. Venieris Dept. of Electrical and Electronic Eng. Imperial College London stylianos.venieris10@ic.ac.uk
 Christos-Savvas Bouganis Dept. of Electrical and Electronic Eng. Imperial College London christos-savvas.bouganis@ic.ac.uk

ABSTRACT
 This work presents Cascade^{CNN}, an automated toolflow that pushes the quantisation limits of any given CNN model, to perform high-throughput inference by exploiting the computation time-accuracy trade-off. Without the need for retraining, a two-stage architecture tailored for any given FPGA device is generated, consisting of a low- and a high-precision unit. A confidence evaluation unit is employed between them to identify misclassified cases at run-time and forward them to the high-precision unit or terminate computation. Experiments demonstrate that Cascade^{CNN} achieves a performance boost of up to 55% for VGG-16 and 48% for AlexNet over the baseline design for the same resource budget and accuracy.

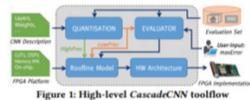


Figure 1: High-level Cascade^{CNN} toolflow

CNN-device pair to select quantisation scheme, configure the confidence evaluation mechanism and generate the cascaded low- and high-precision processing units.

1 INTRODUCTION

While Convolutional Neural Networks are becoming the state-of-

2 CASCADE^{CNN}

Research

In the Intelligent Digital Systems Lab, we perform research towards high-performance (embedded) digital systems spanning several topic areas, including machine learning, computer vision, and robotics.

MORE DETAILS

@CBouganis

Tweets by @CBouganis

Christos @CBouganis
 We are recruiting for an exciting post on Machine Learning and FPGAs. Please see details here: tinyurl.com/yadhvwnd

Research Assistant/Associat...
 The Intelligent Digital Systems ...
 imperial.ac.uk

Aug 4, 2018

Christos @CBouganis
 Excited to announce the 2nd EDI

