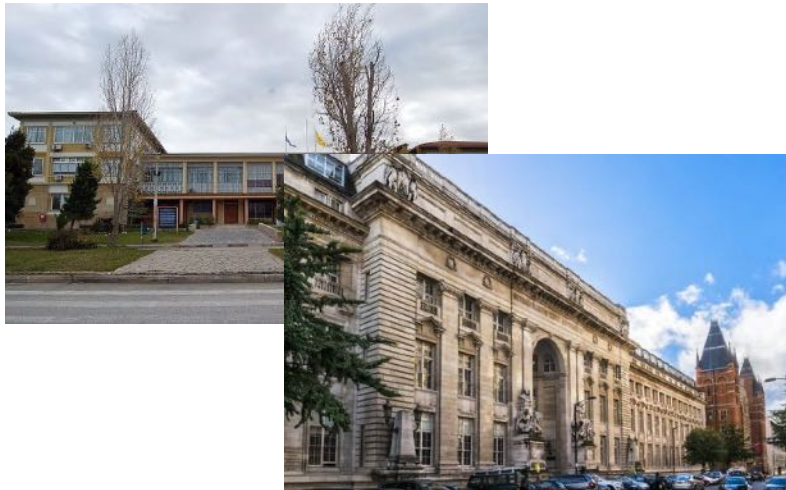
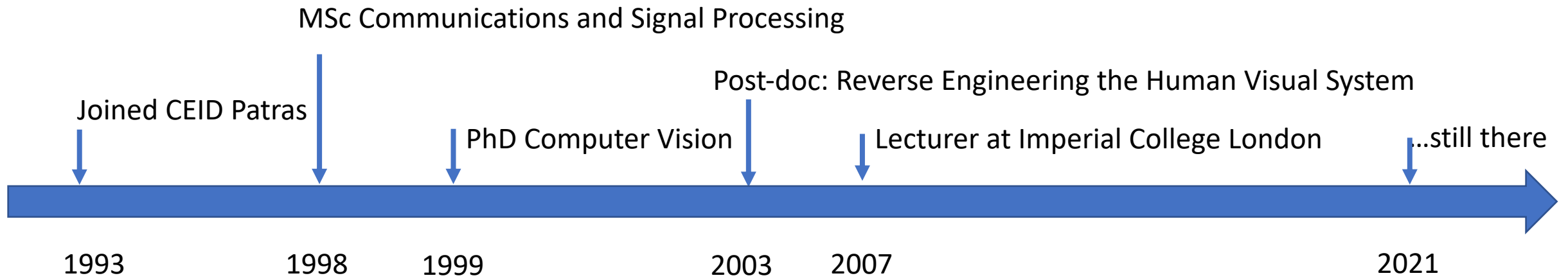


Deploying DNNs in the Embedded Space: Challenges and Opportunities

Christos-Savvas Bouganis

About myself



The team



Aditya Rajagopal
Machine Learning



Alexandros Kouris
Machine Learning,
Robotics



Diederik Vink
Machine Learning



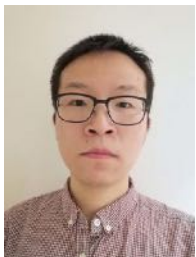
Alexander Montgomerie
Hardware Acceleration
for Machine Learning



Mudhar Bin Rabieah
Machine Learning



Giorog Zampokas
Computer Vision,
Machine Learning



Zhewen Yu
Machine Learning



Petros Toupas
Machine Learning

Our vision

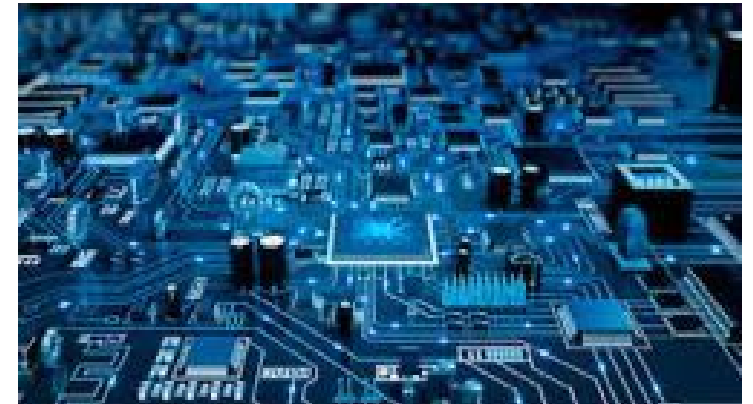
To research and develop intelligent autonomous systems



+



+



“see”

“understand”

“process”

Some of our work

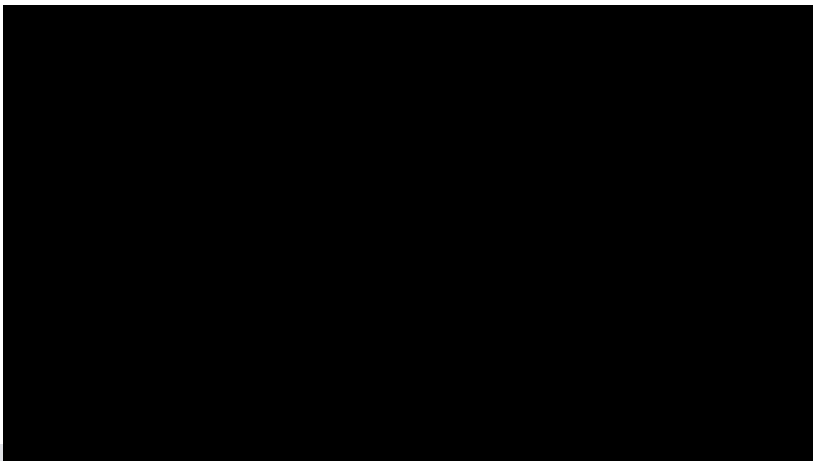
Autonomous Navigation



Human Pose Estimation



Traffic Detection



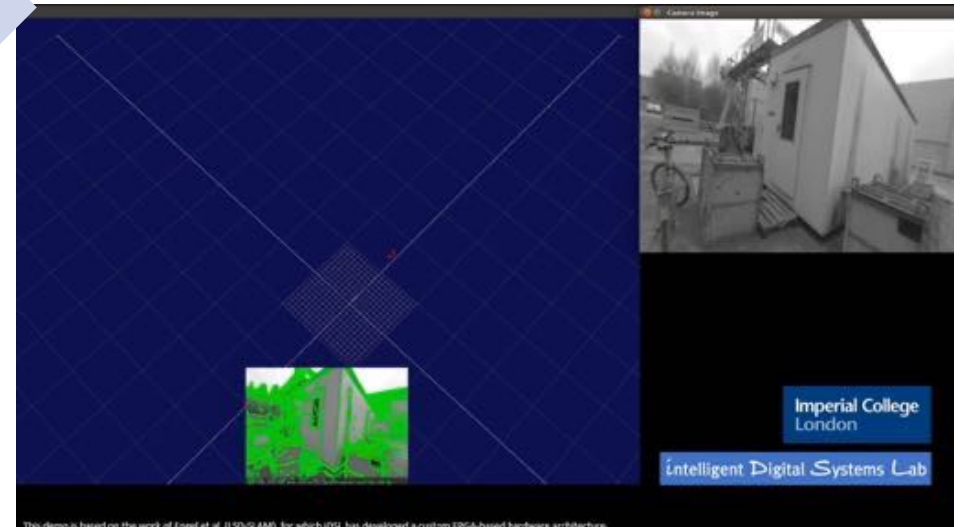
fpgaConvNet

Multi-CNN
Deployment

Time-
constrained
LSTM
Inference

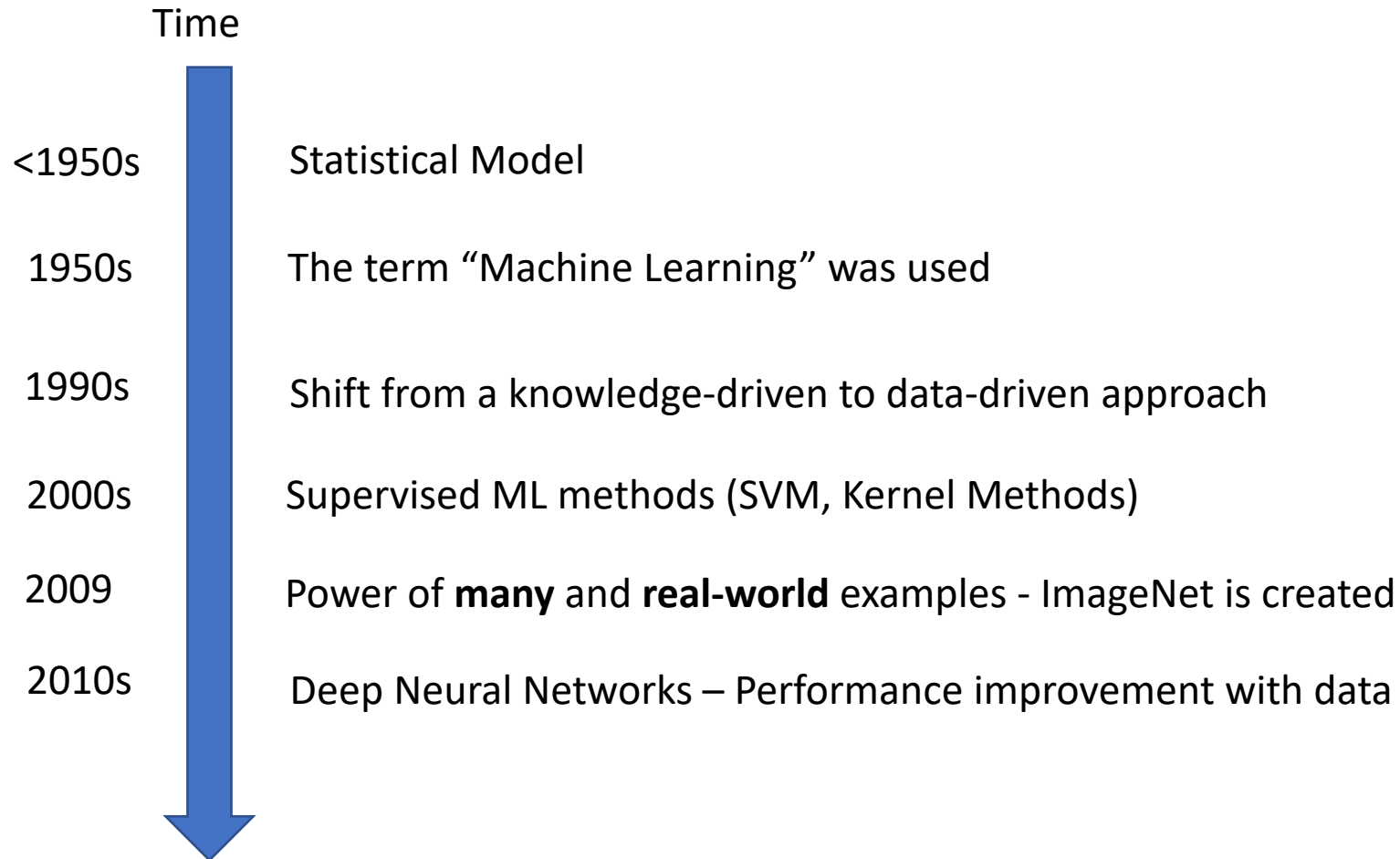
Data-Driven
CNN
Inference

Localisation and Mapping



This demo is based on the work of Engel et al. (1509.0457), for which IDS1 has developed a custom FPGA-based hardware architecture.

A bit of history: Artificial Intelligence - Machine Learning – Deep Neural Networks



Artificial Intelligence

Machine Learning

Deep Neural
Networks

CNNs

ImageNet Challenge

IMAGENET

- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.



ImageNet Challenge

IMAGENET



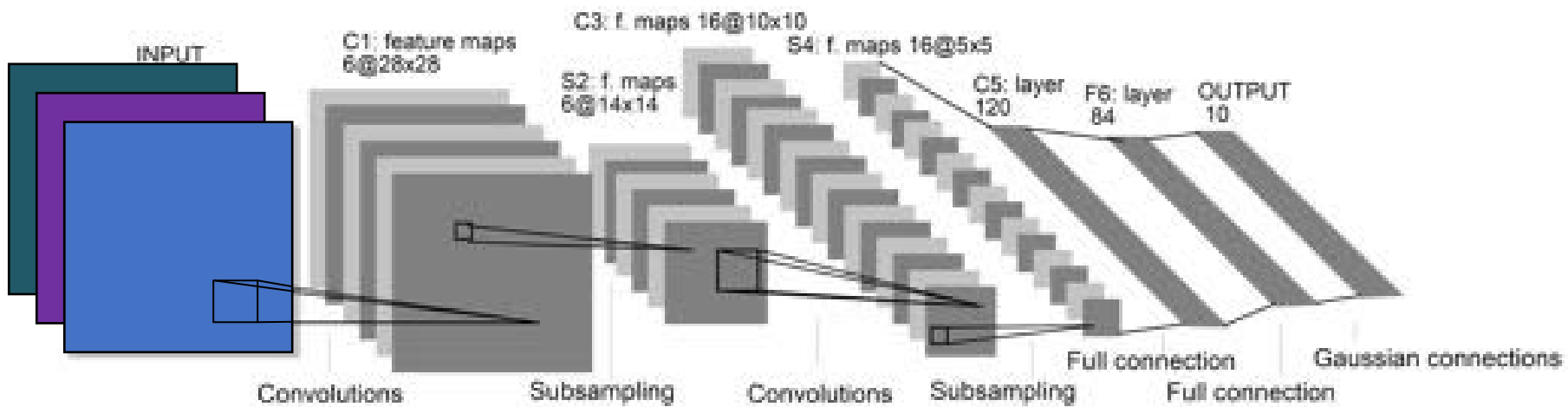
- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.

convolutional
+ nonlinearity

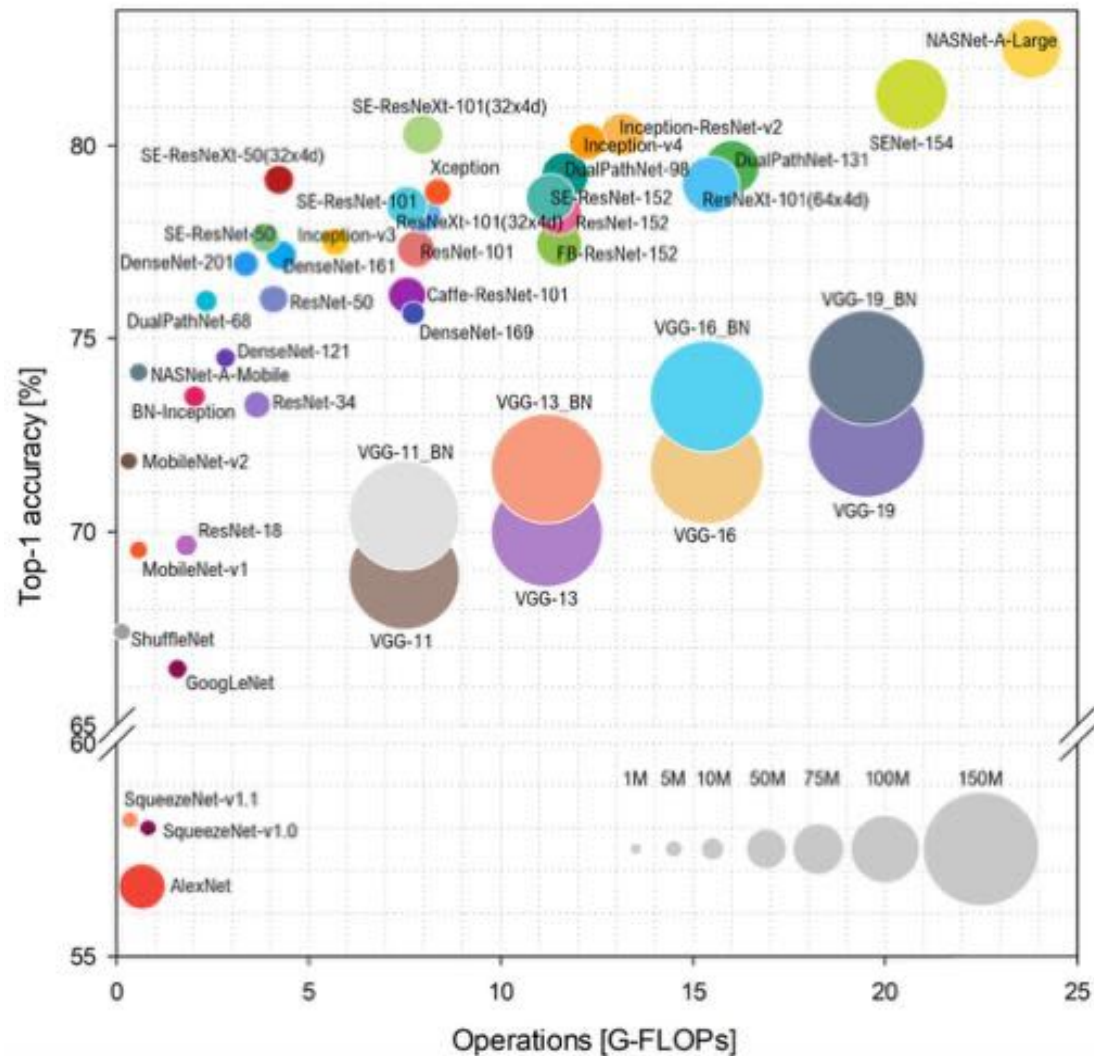
pooling

convolutional
+ nonlinearity

pooling



Models – Where we are today



- Number of models trading-off complexity vs accuracy
- Top-1 accuracy 82% (increase of 30 pp)
- 20x higher computational complexity

DNNs in the Embedded Space – Variability in Performance Requirements

surveillance



Aerial Monitoring



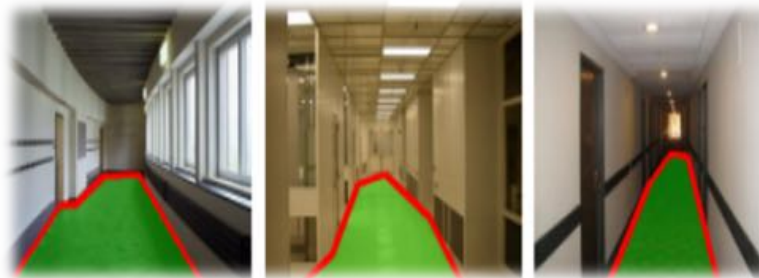
Autonomous Driving



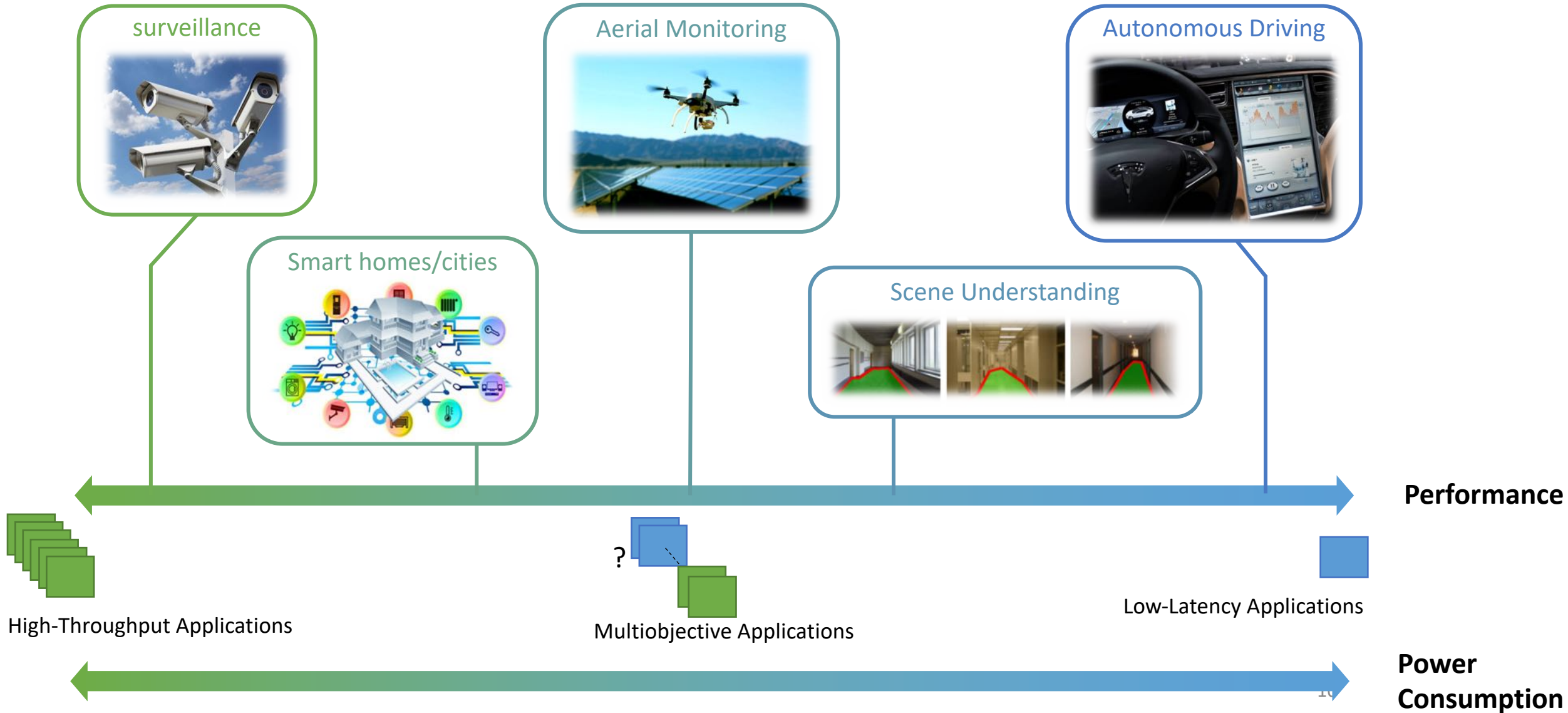
Smart homes/cities



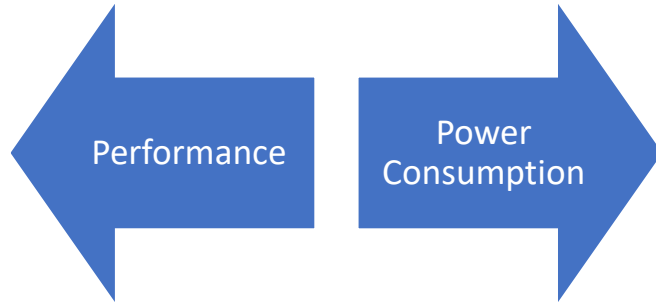
Scene Understanding



DNNs in the Embedded Space – Variability in Performance Requirements



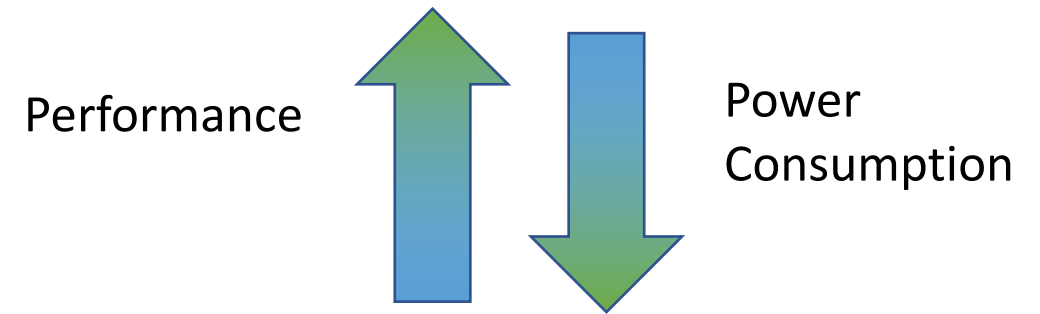
Challenge: Performance vs Power



Integer	
Add	
8 bit	0.03pJ
32 bit	0.1pJ
Mult	
8 bit	0.2pJ
32 bit	3.1pJ

FP	
FAdd	
16 bit	0.4pJ
32 bit	0.9pJ
FMult	
16 bit	1.1pJ
32 bit	3.7pJ

Memory	
Cache	(64bit)
8KB	10pJ
32KB	20pJ
1MB	100pJ
DRAM	1.3-2.6nJ



Efficient utilisation of the resources

- Compute resources
- Memory resources

Have the Right Data in the Right Place at the Right Time

Efficiency comes from customisation

Platform Layer

Generic

Application Specific

DSPs
Qualcomm Hexagon,
Apple Neural Engine,

GPUs
Tegra K1, X1 and X2

FPGAs
Custom datapath
Custom memory subsystem

ASICs
TPU

MAXIMIZE



EFFICIENCY

Customisation



Algorithm

Implementation
(GEMM, Winograd)

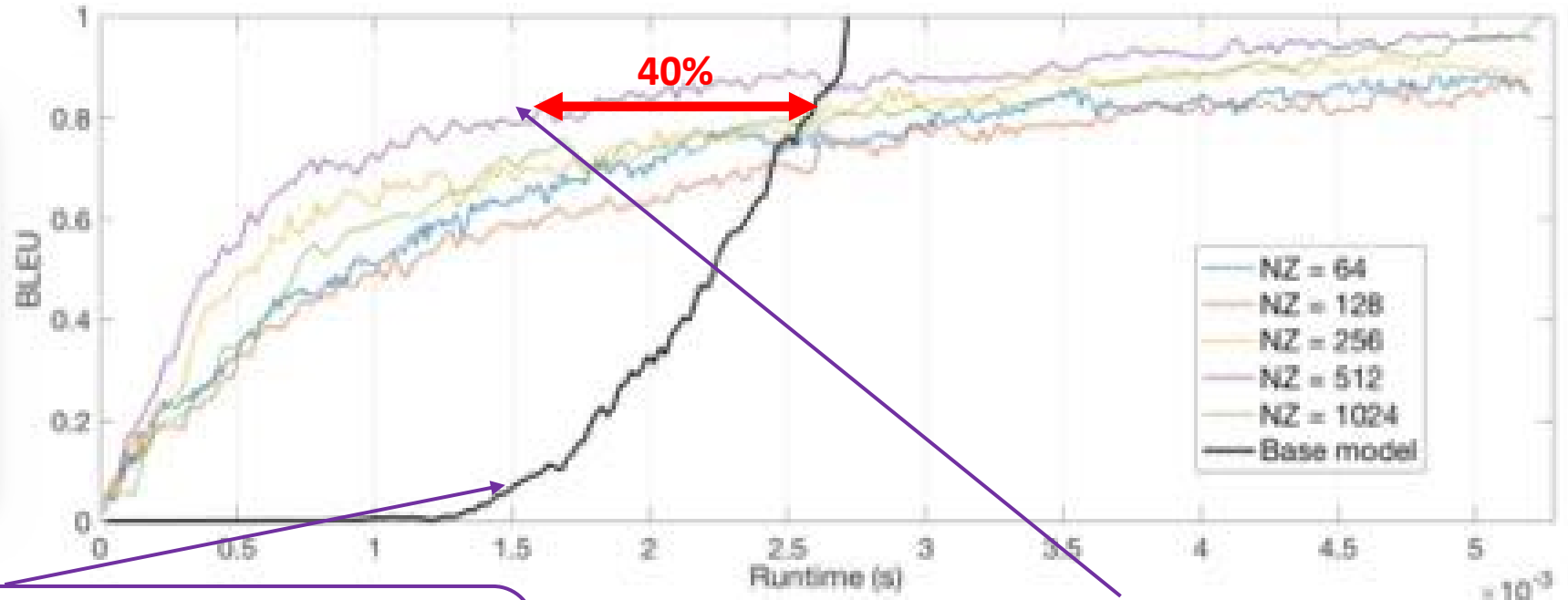
Approximations

Algorithmic Layer

Putting things in perspective – What customization buys you

Impact on LSTM-based Image Captioning – Computations tailored to the architecture

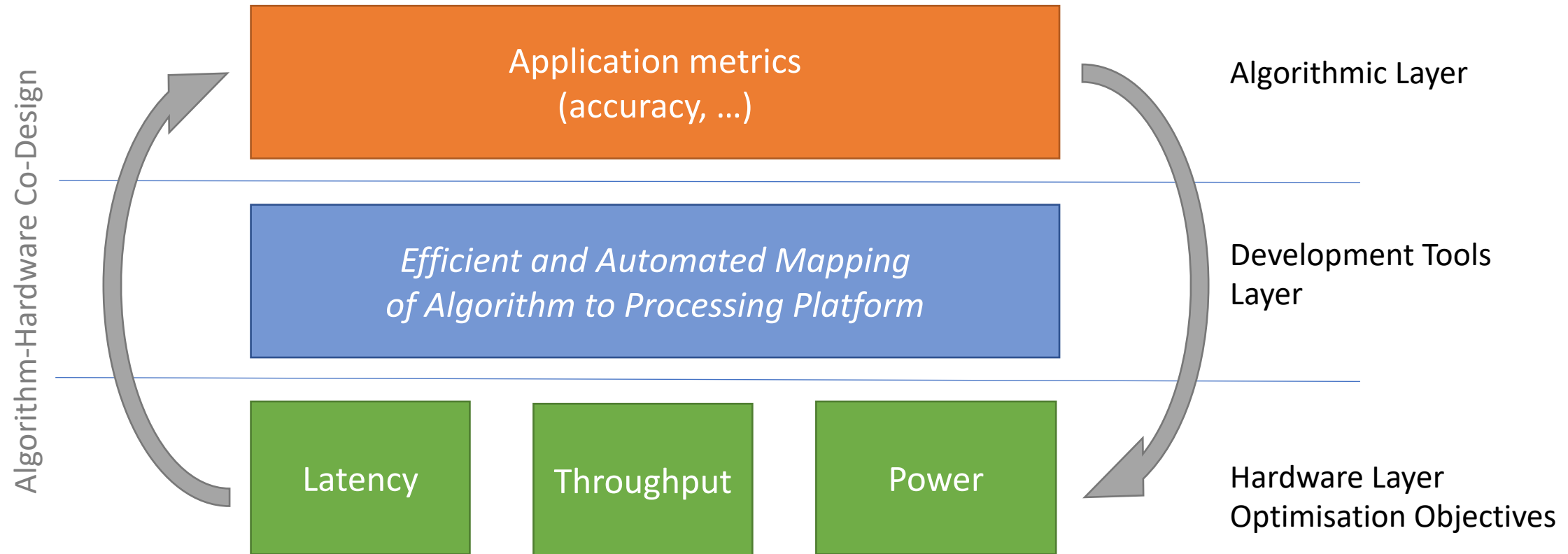
Input Image



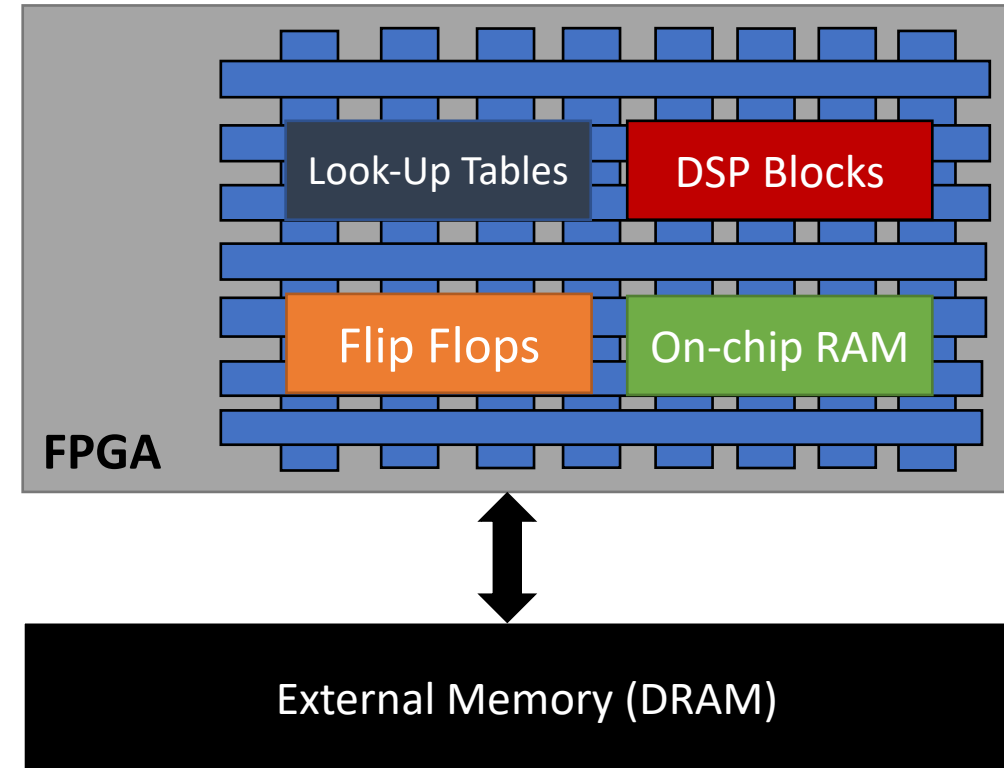
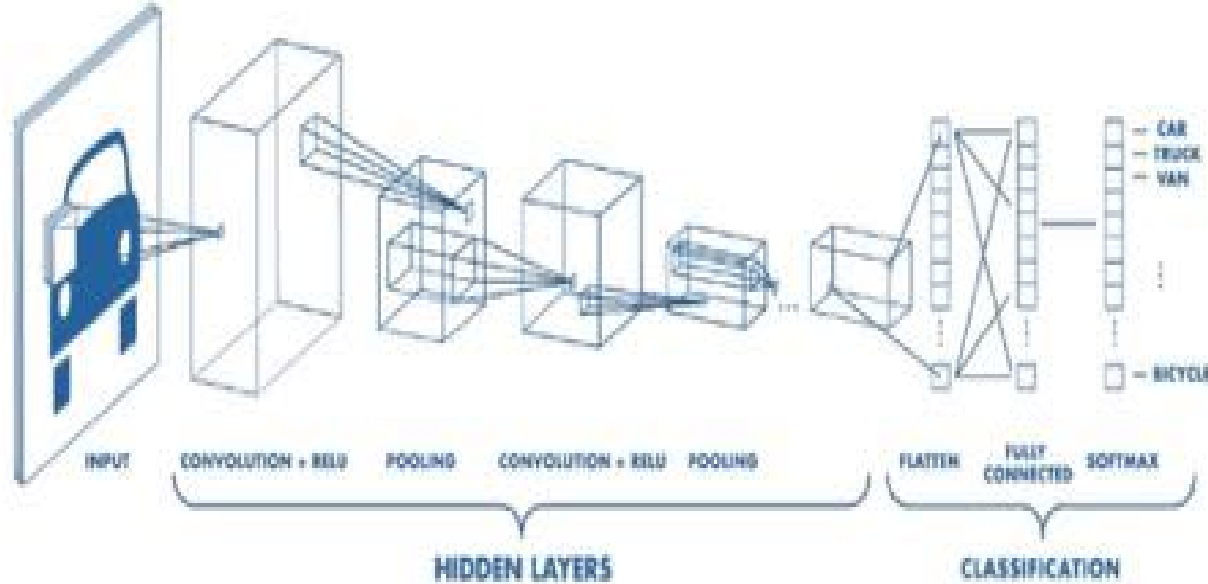
0) a man is sitting on a <UNK> with a <UNK> . (p=0.000000)
 1) a man is sitting on a <UNK> with a <UNK> (p=0.000000)
 2) a man is sitting on a <UNK> with a small dog . (p=0.000000)
 3) a man is sitting on a <UNK> with a small dog (p=0.000000)
 4) a man is sitting on a <UNK> with a <UNK> on the ground . (p=0.000000)

0) a brown dog laying on top of a pile of luggage . (p=0.000031)
 1) a brown dog laying on top of a pile of shoes . (p=0.000016)
 2) a brown dog laying on top of a rug . (p=0.000015)
 3) a brown dog laying on top of a pile of clothes . (p=0.000010)
 4) a dog is laying on the floor next to a stuffed animal . (p=0.000007)

Algorithm-Hardware Co-design



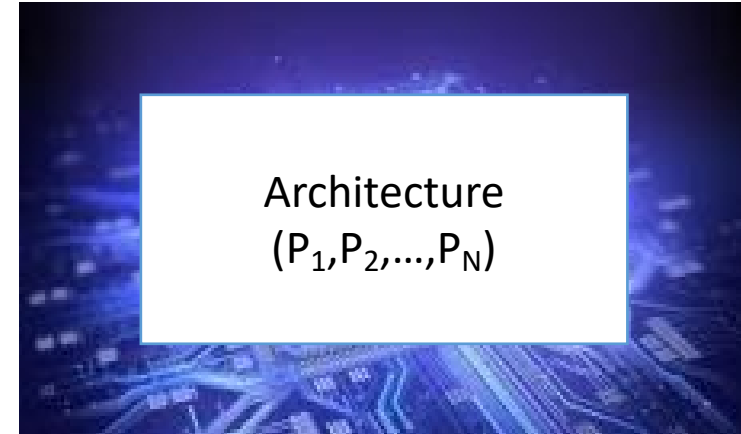
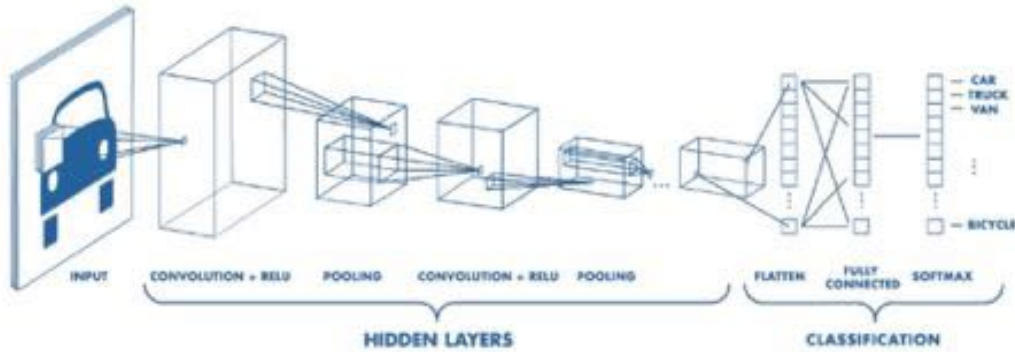
CNN acceleration through an FPGA



Characteristics

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

The Challenge of the Mapping Problem



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

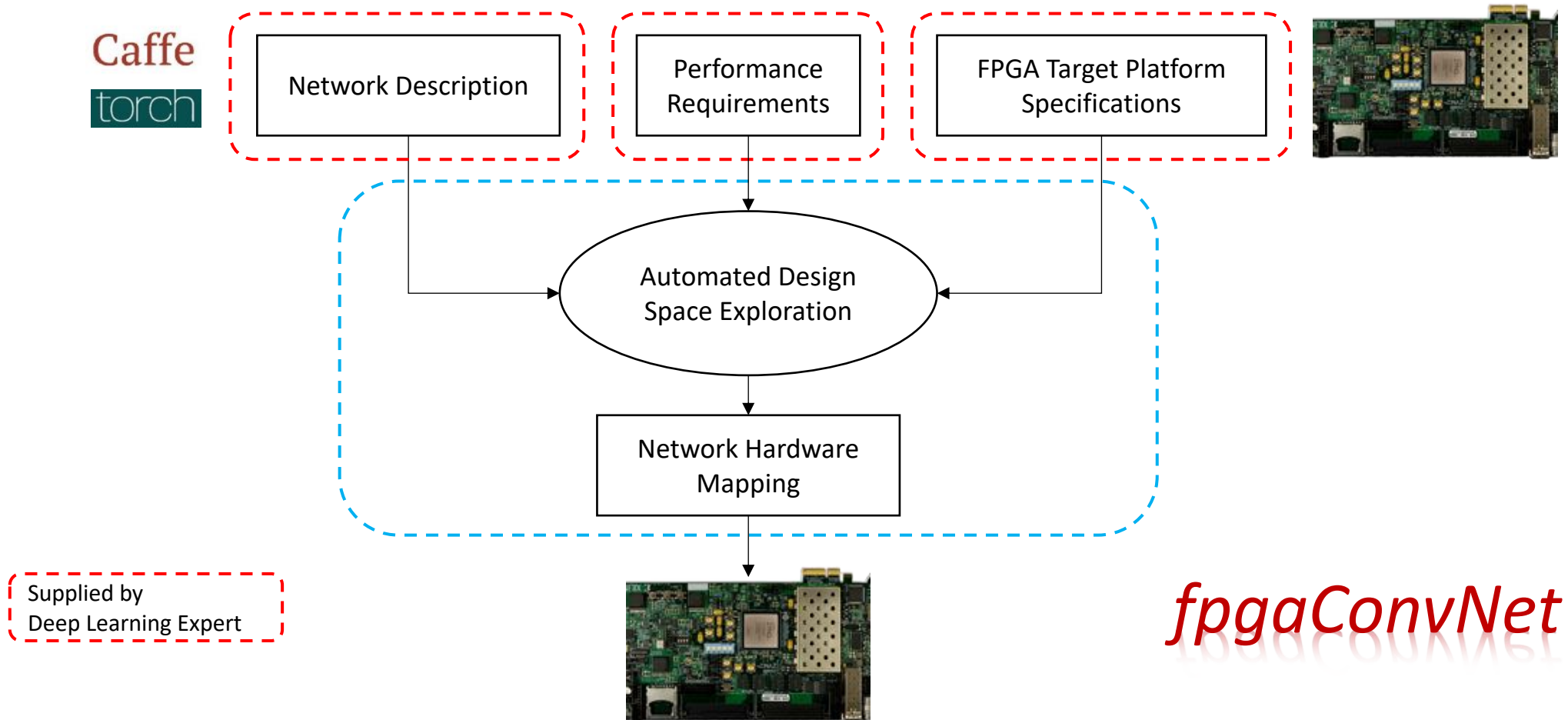
Specifications

- Latency
- Throughput
- Power consumption

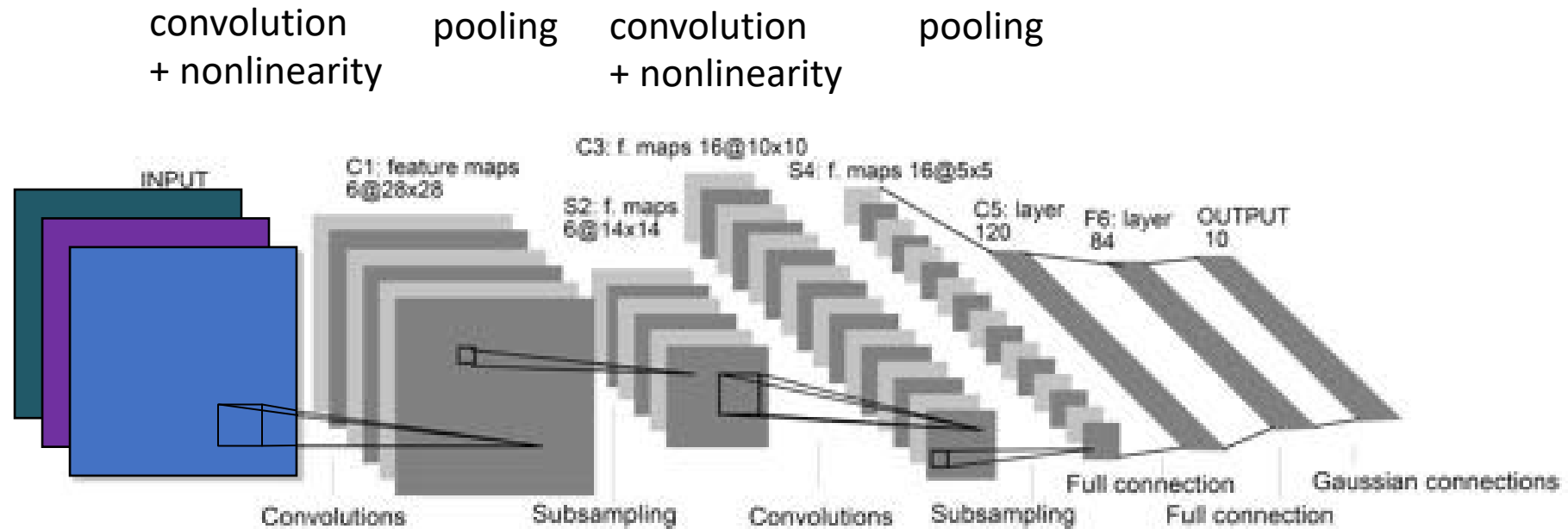
Challenges:

- Diversity of operations in modern NN
- Diversity and resources of modern FPGAs
- Competition (or need for performance)
- Large number of parameters in the target architecture

Challenge #1: Automated CNN-to-FPGA Toolflow

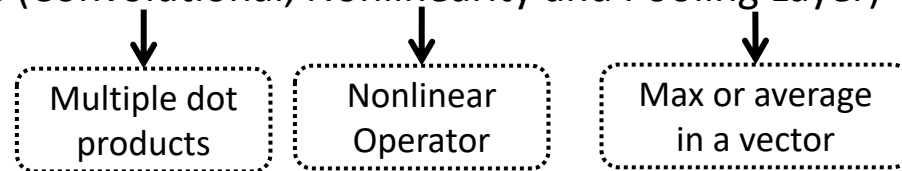


Under the hood: Convolutional Neural Networks (ConvNets)

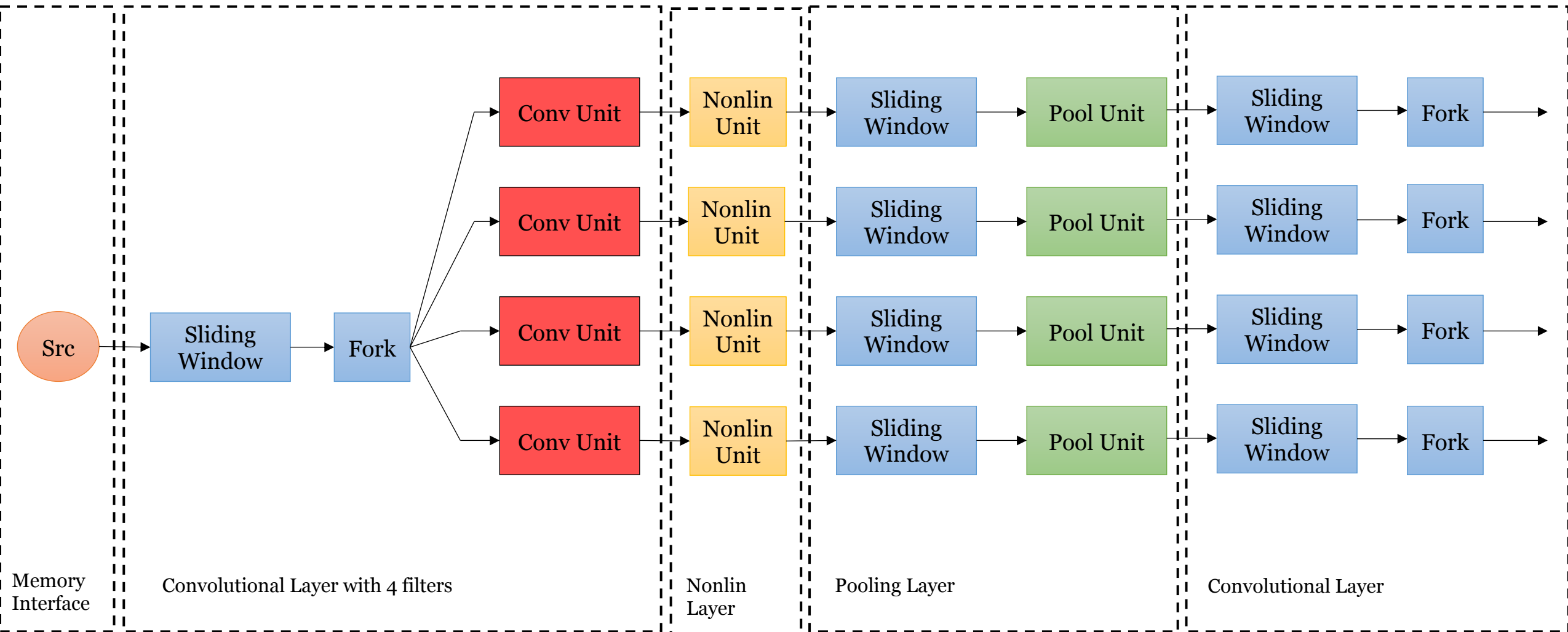


- ConvNet Inference

- Tailored to images and data with spatial patterns
- Built as a sequence of layers (Convolutional, Nonlinearity and Pooling Layer)
- Feedforward operation
- Inherently streaming

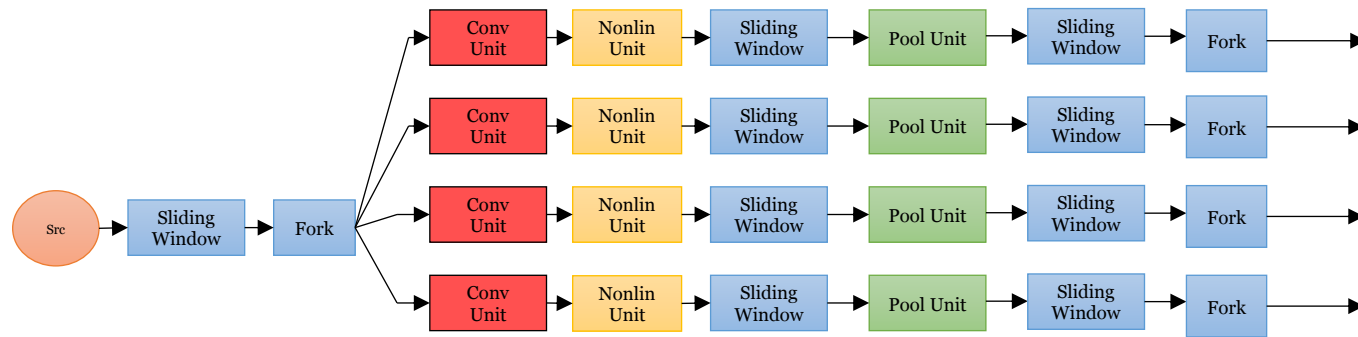


fpgaConvNet – Streaming Architecture for CNNs



fpgaConvNet – Streaming Architecture for CNNs

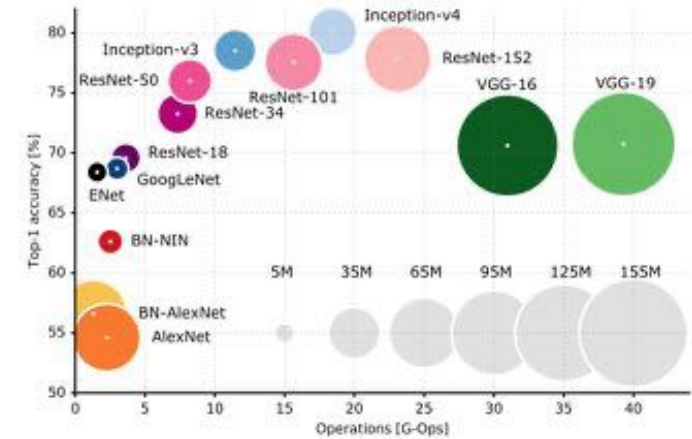
CNN Hardware SDF Graph



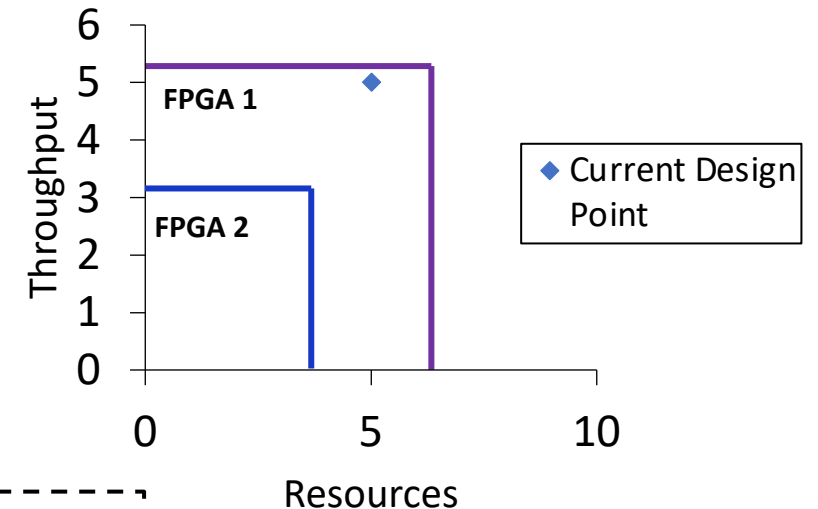
Complex Model → Bottlenecks:

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

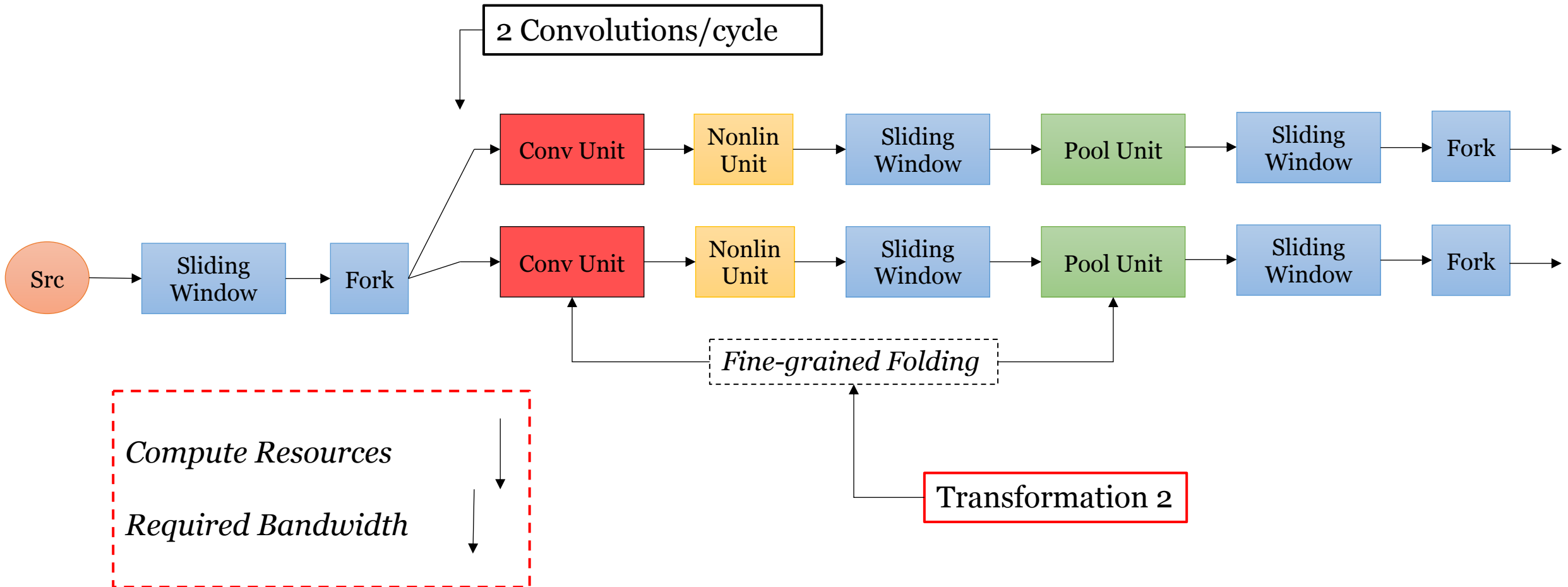
Define a set of **graph transformations** to traverse the design space in **fast** and **principled** way



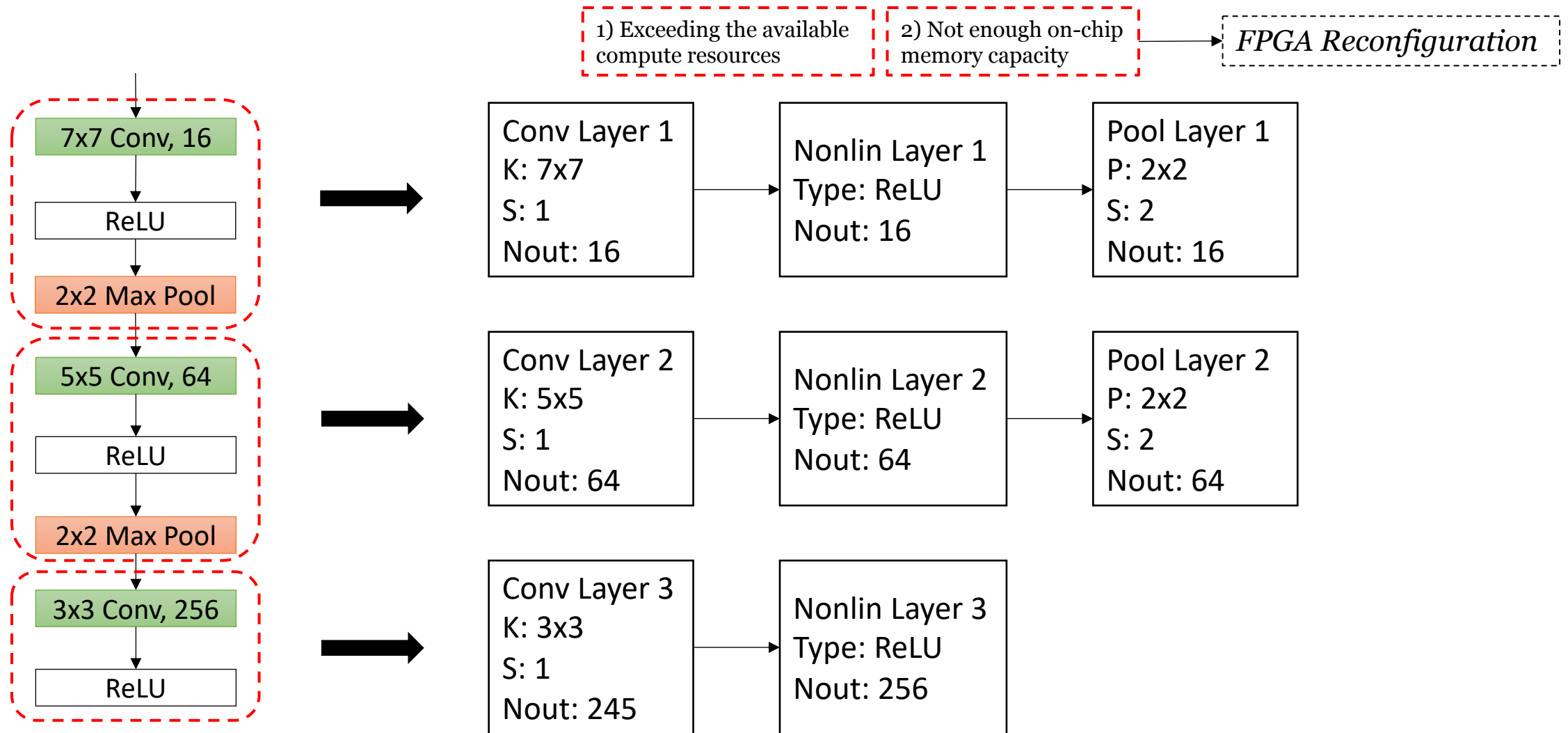
Design Space



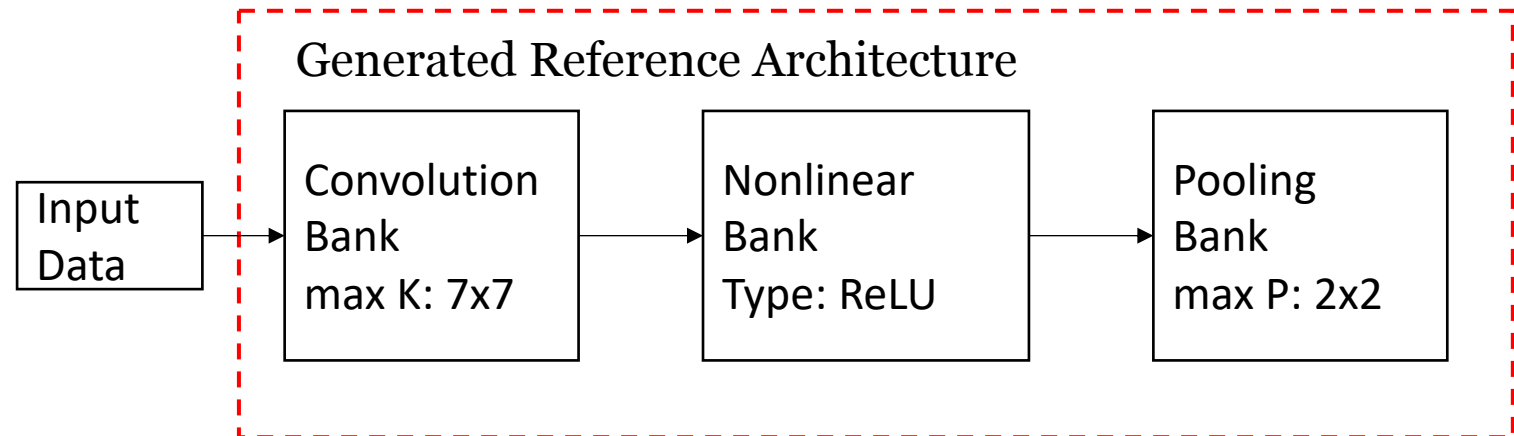
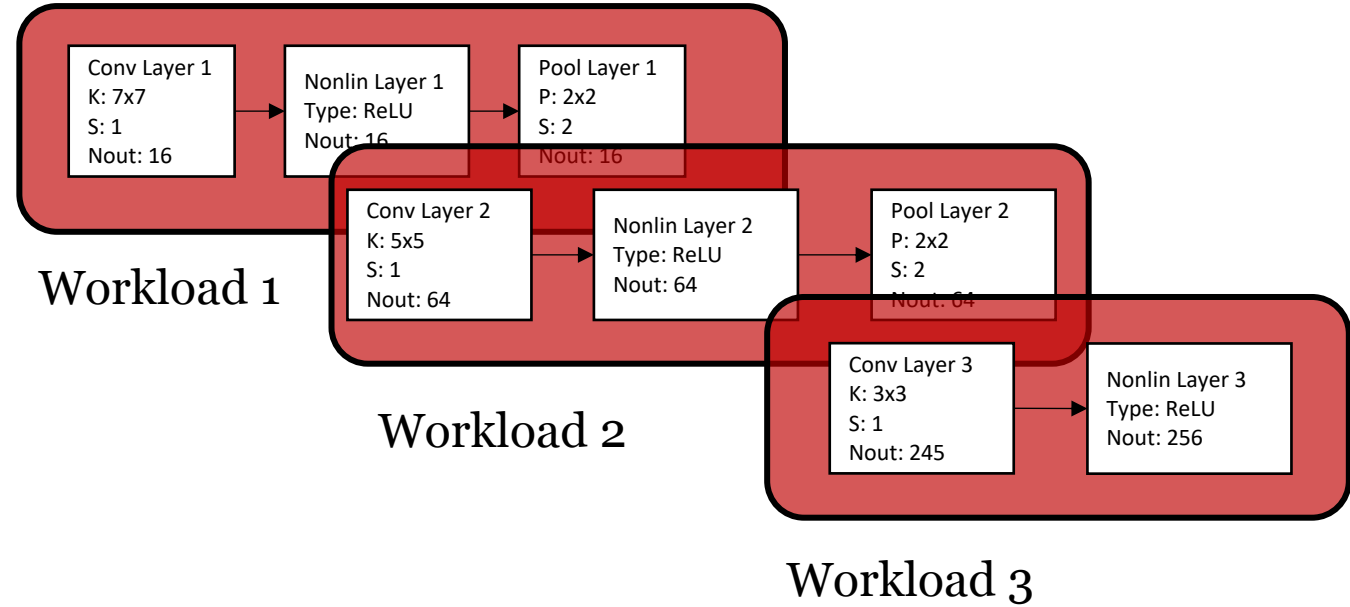
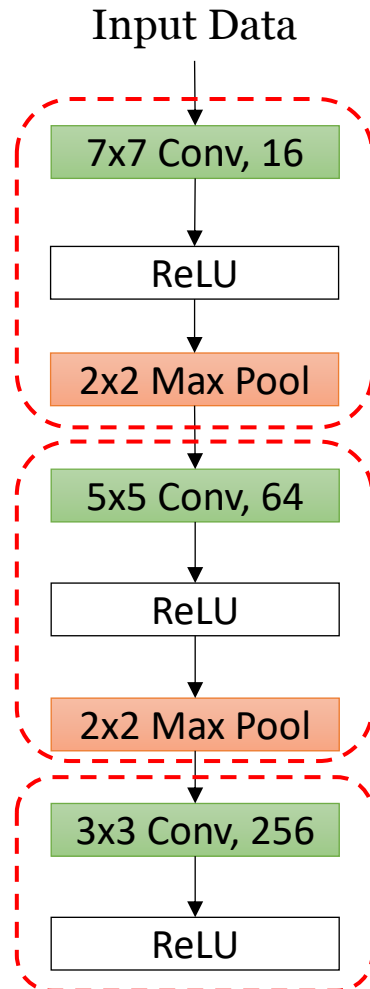
Transformations 1 & 2: Coarse- and fine-grained Folding



Transformation 3: Graph Partitioning with Reconfiguration

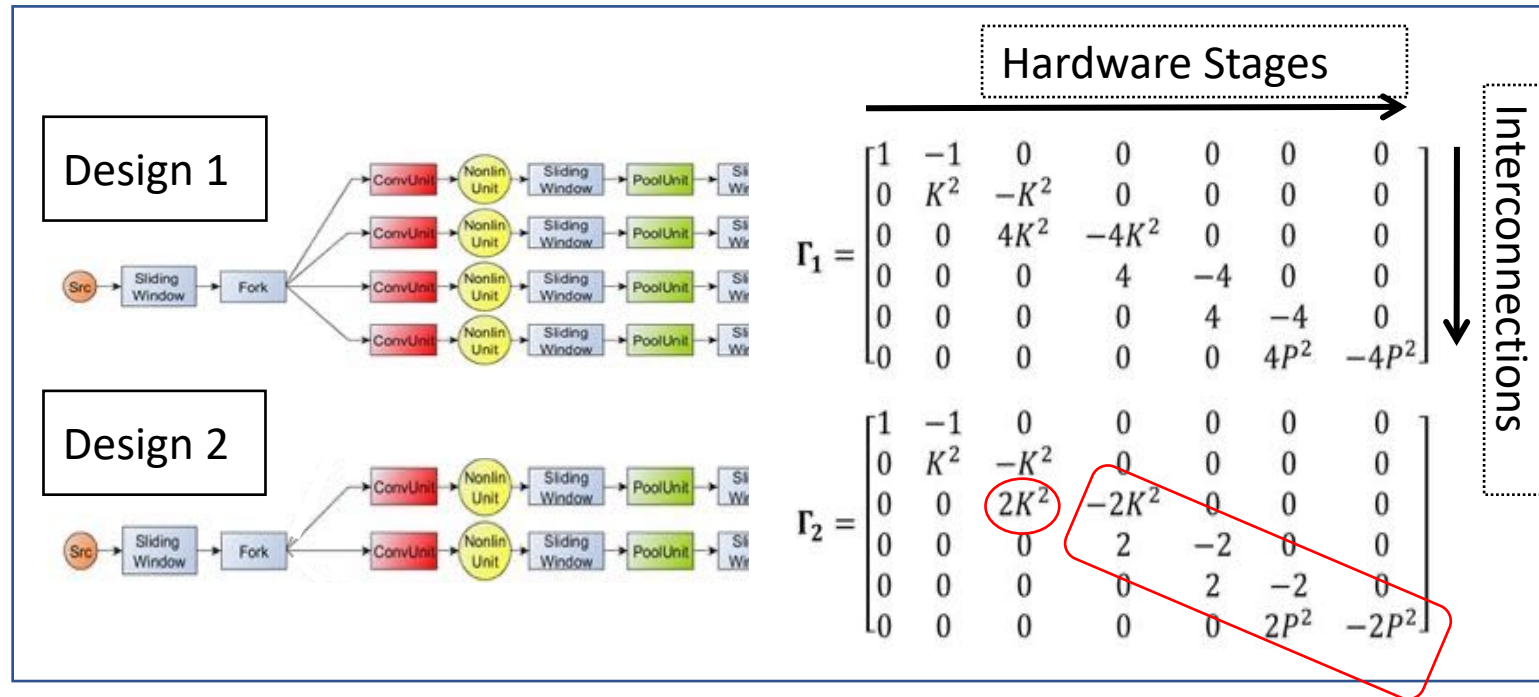


Transformation 4: Weights Reloading



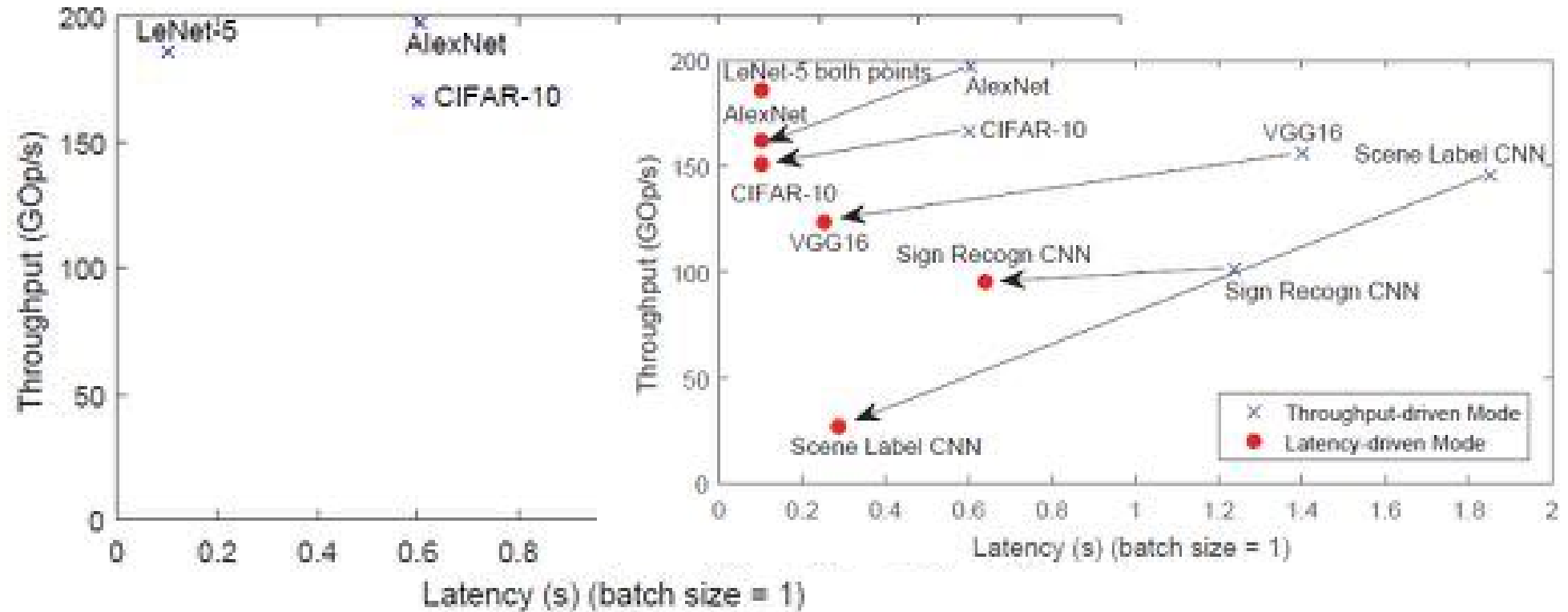
fpgaConvNet – Design Space Exploration and Optimisation

- Synchronous Dataflow Modelling
 - Capture hardware mappings as matrices
 - Transformations as *algebraic operations*
 - Analytical *performance model*
 - Cast design space exploration as a mathematical optimisation problem

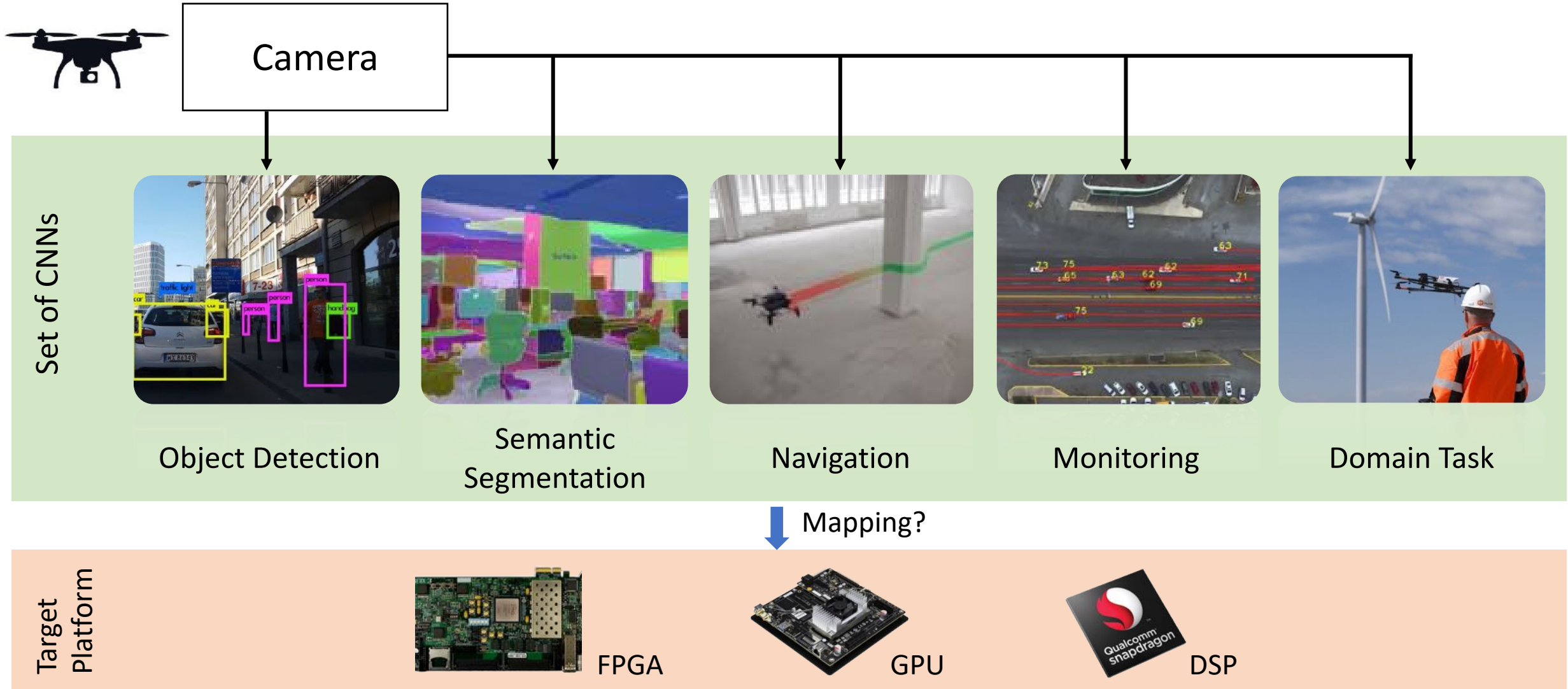


$$t_{total}(B, N_P, \Gamma) = \sum_{i=1}^{N_P} t_i(B, \Gamma_i) + (N_P - 1) \cdot t_{reconfig}.$$

Meeting the performance requirements



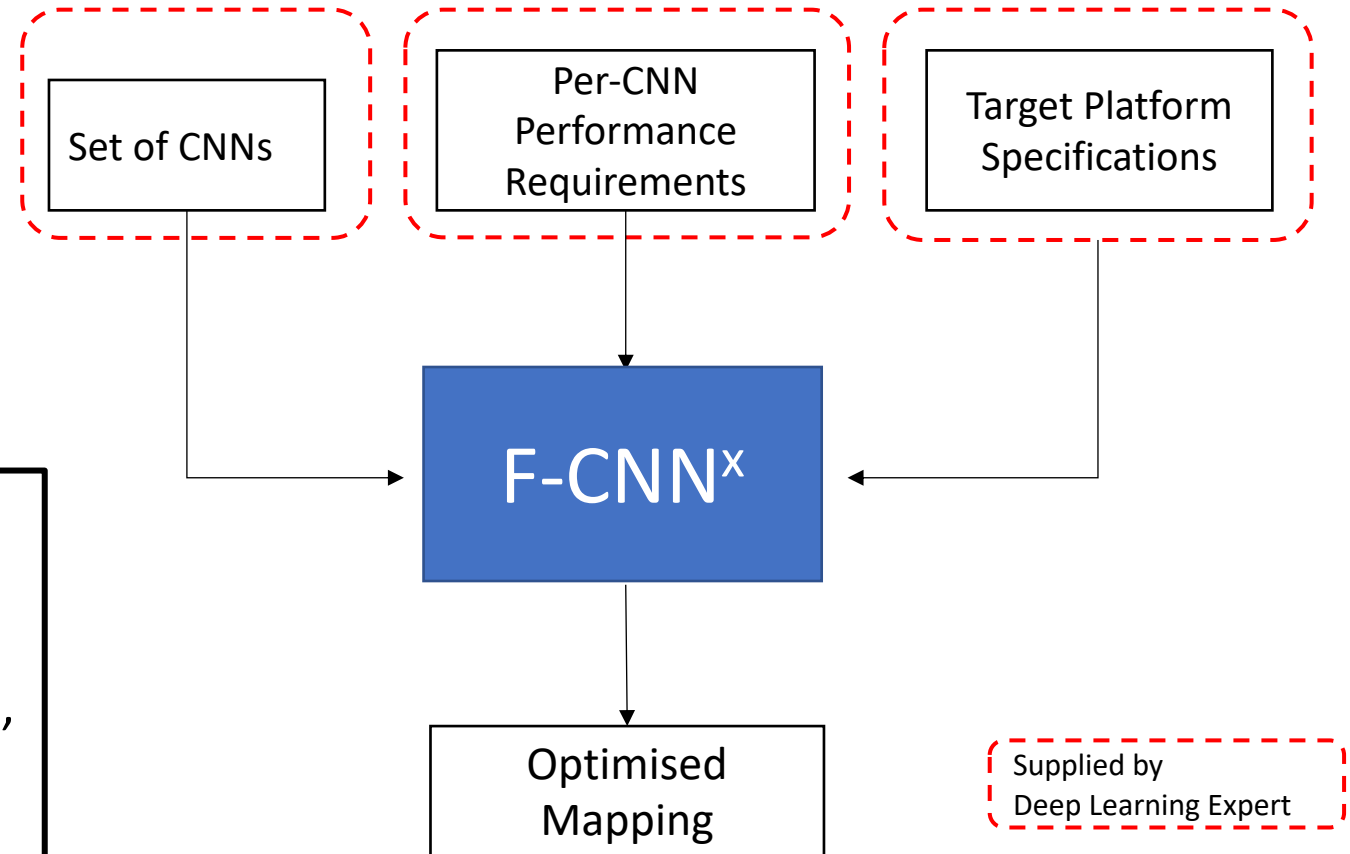
Challenge #2: Multi-CNN Systems – Autonomous Drones



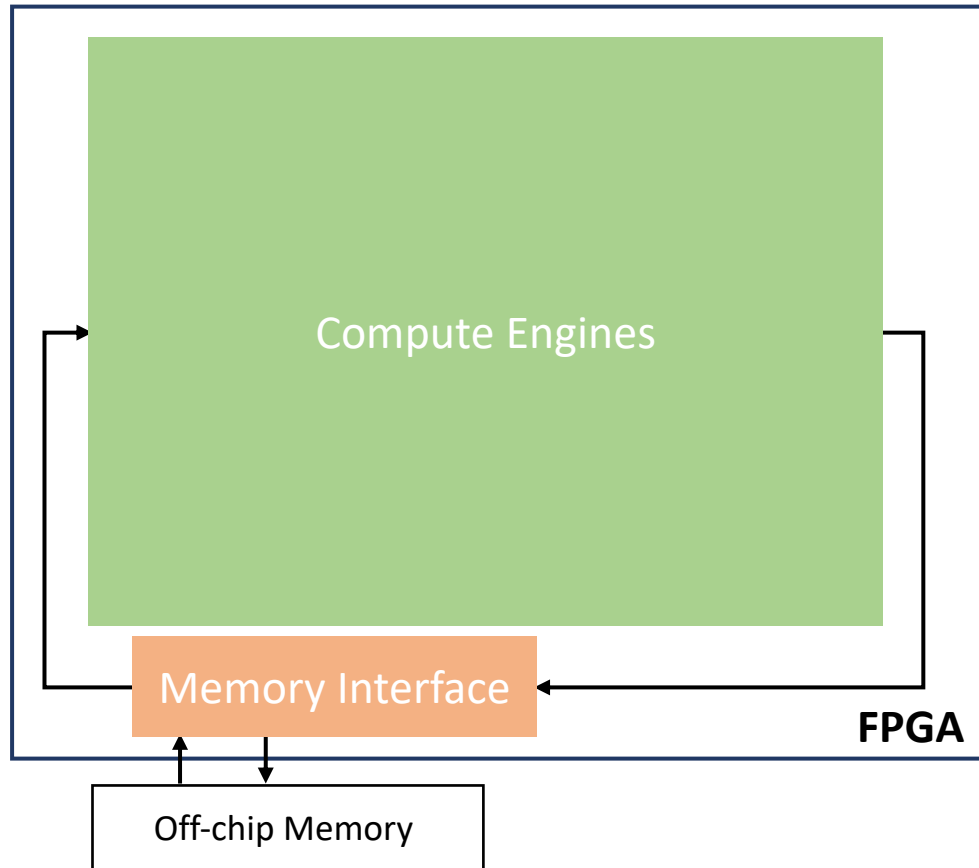
Challenge #2: Multi-DNN System

Challenges:

- Resource allocation among CNNs
- Design automation
- Models with different performance constraints, e.g. required throughput and latency
- Competing for the same pool of resources
- High-dimensional design space



Multi-CNN Hardware Architecture



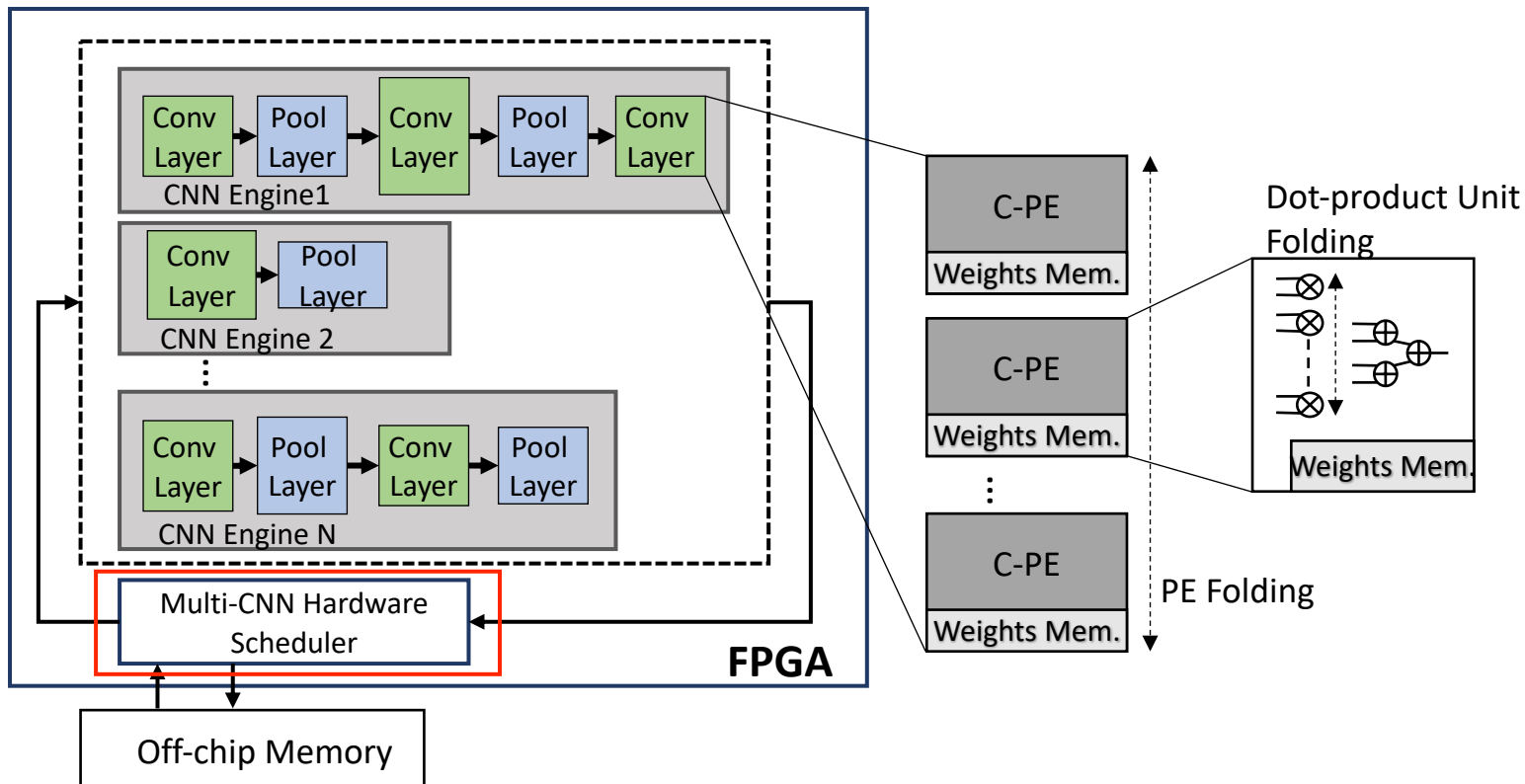
Key characteristics

- Latency is relevant: Reconfiguration is not an option
- One hardware engine per CNN – highly customisable
- Hardware scheduler to control memory access schedule

Multi-CNN Hardware Architecture

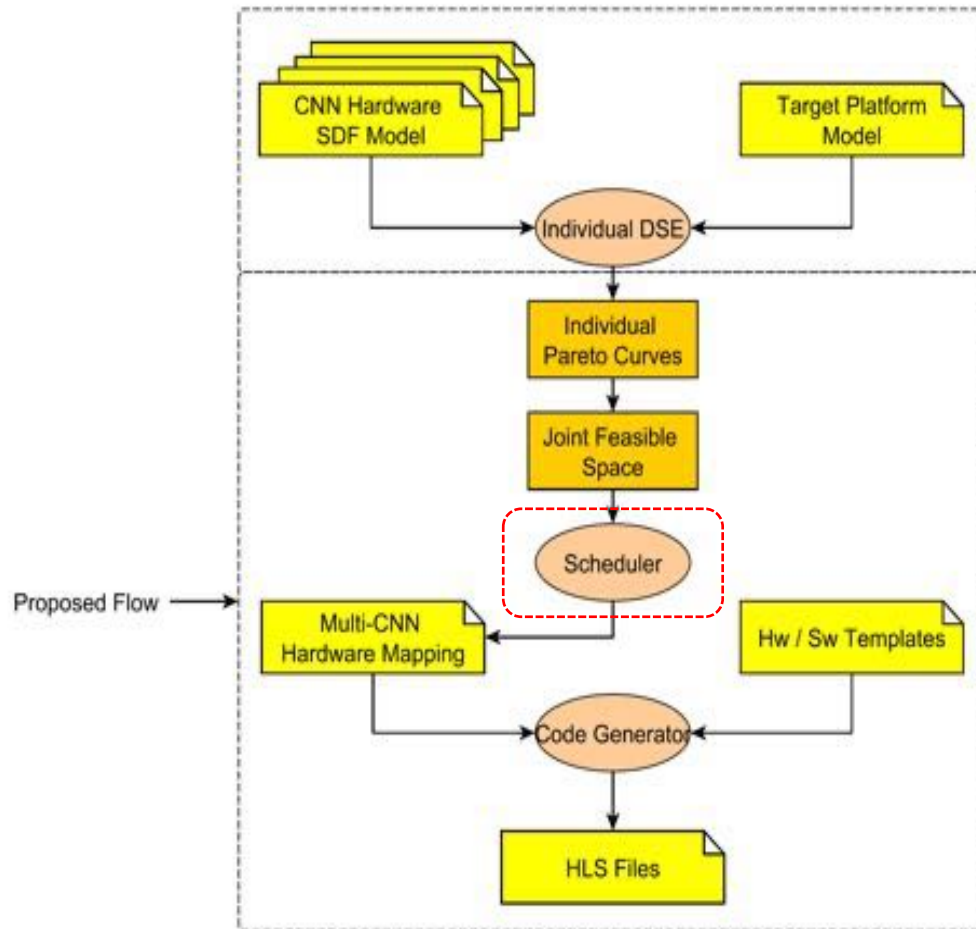
Key characteristics

- One hardware engine per CNN – highly customisable
- Hardware scheduler to control memory access schedule



Parameter	Symbol
Pipeline structure	Γ_i
No. of PEs in each stage	$N_{PE,ij}$
No of MAC operators within each PE	$N_{op,ij}$
Schedule	S

Proposed Design Space Exploration Method



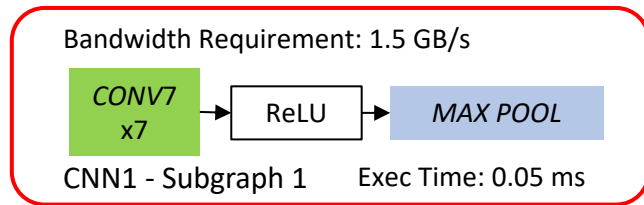
- Memory contention
 - Problem 1: Performance model \neq Actual performance (scheduler)
 - Problem 2: Not full utilization of the memory bandwidth
- CNN inference over a stream of inputs
 - Cast to a **cyclic scheduling problem**
 - Search for a periodic solution
- Optimal ILP scheduler has very high runtimes for large-sized problems
- Develop a heuristic Resource Constrained List Scheduler (RCLS).
- Key points:
 - Scheduler exposed in the engine design optimization process
 - Introduce slow-down => fine control over bandwidth

The effect of slow-downs

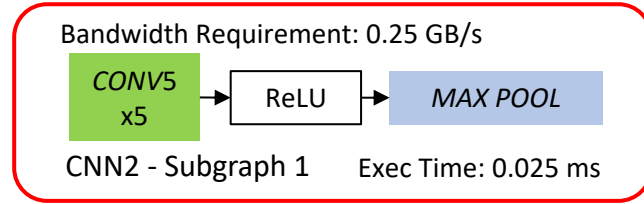
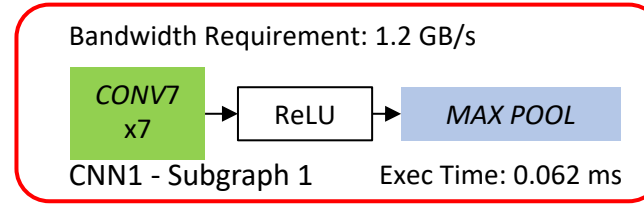
Scheduler

Scheduler + slow downs

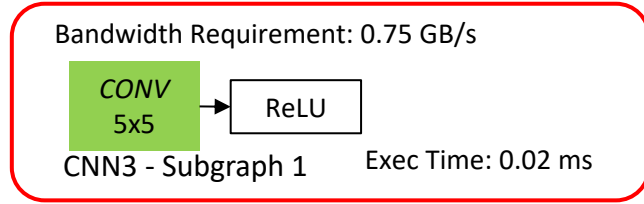
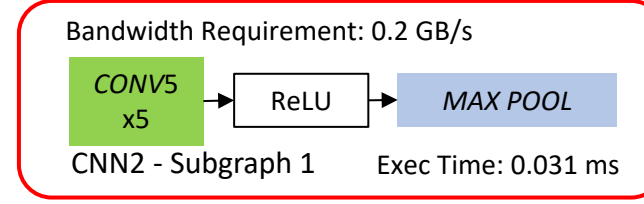
Available Memory Bandwidth: 2 GB/s



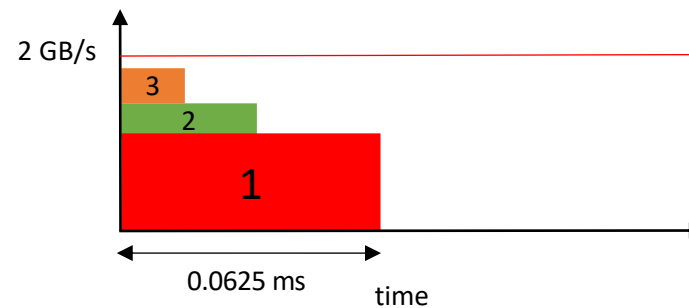
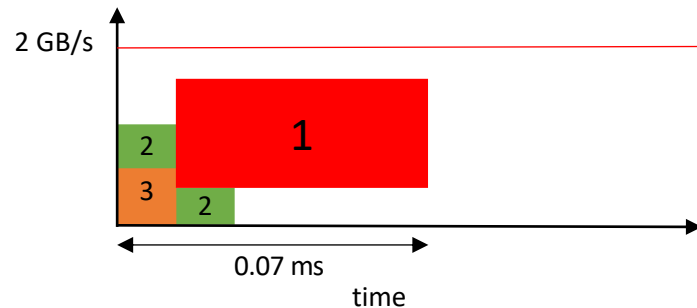
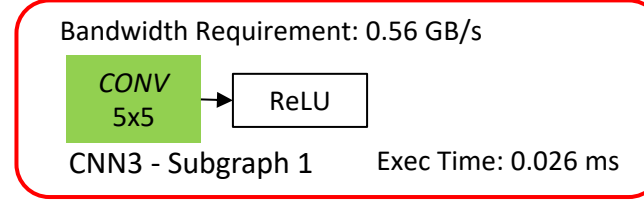
Slowdown1_1:
0.8x



Slowdown2_1:
0.8x

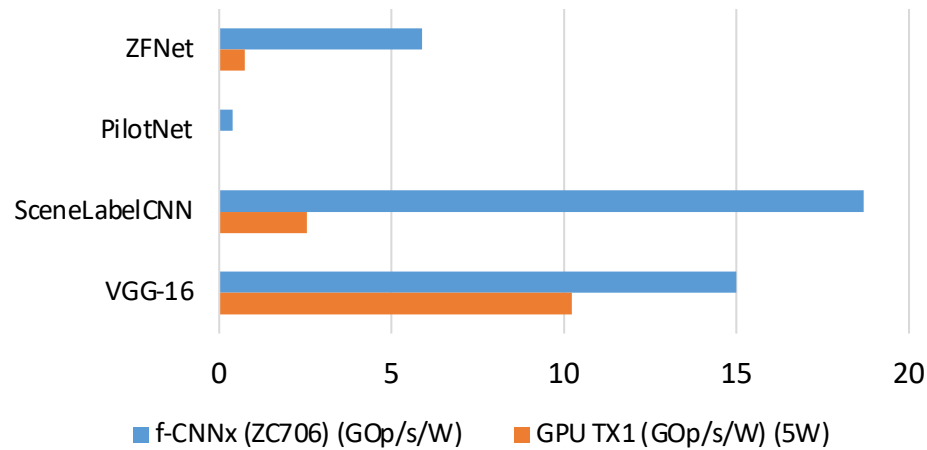


Slowdown3_1:
0.75x



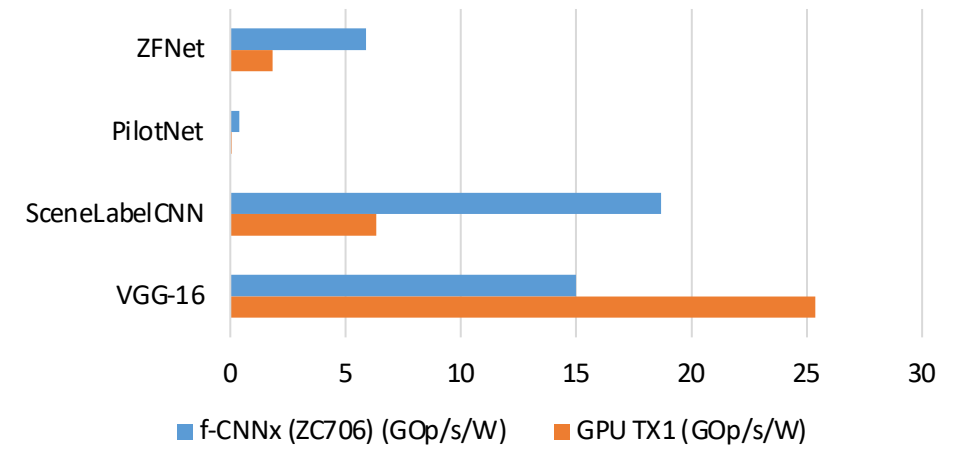
Comparison with Embedded GPUs

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



- Latency-driven scenario → batch size of 1
- Up to 19.09× speedup with an average of 6.85× (geo. mean)

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



- Latency-driven scenario → batch size of 1
- Up to 9.61× speedup with an average of 2.76× (geo. mean)

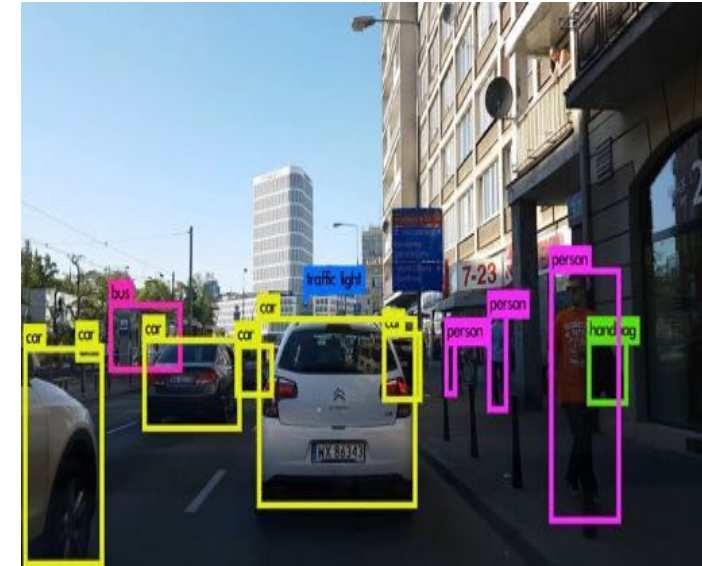
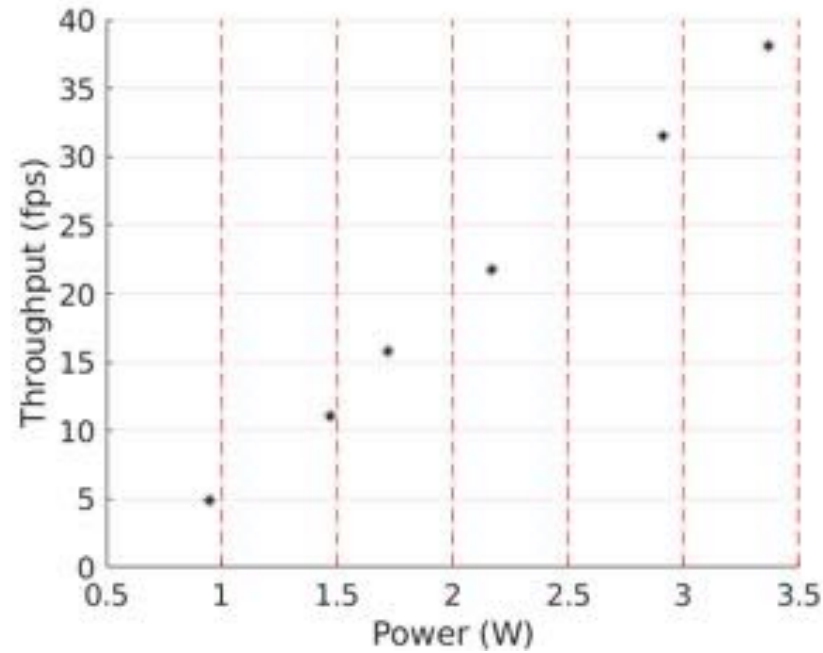
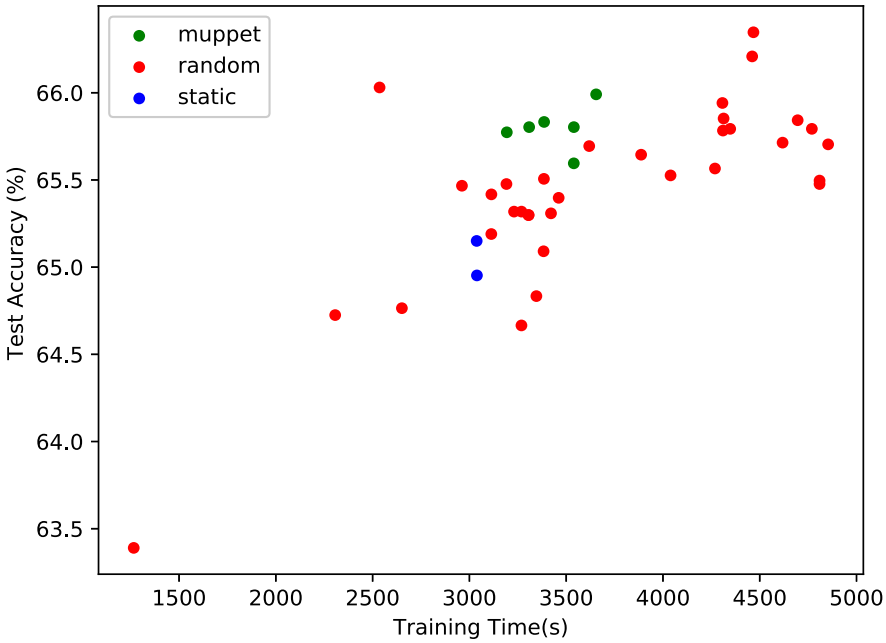
Summary

- Customisation is **key**, but also a **challenge** in the design of DNN systems
- We need toolflows to support deployment of DNN on the embedded space
 - Many choices, high-dimensional space
- Exposing the hardware capabilities to the algorithm can lead to performance gains
 - Challenging task
 - Rethink current approaches to fully utilise the underlying hardware



What we are looking into...

Accuracy-Time Trade-off for Resnet20 on CIFAR100



DNN Training -
MuPPET

Power-aware
CNN mapping

Object
Detection to
FPGA
mapping

Homomorphic
Encryption ML
loads

On-device
adaptation

What we are looking into...

Co-optimize **topology** and **hardware architecture**



*Model
(accuracy)*



*HW architecture
(latency, throughput, resources)*

What we are looking into...

Adversarial attacks to DNNs and how to prevent them



Tesla “sees” 85

Opportunities at Imperial

- MSc Programmes
 - Analogue and Digital Integrated Circuit Design
 - Applied Machine Learning
 - Communications and Signal Processing
 - Control and Optimisations
 - Future Power Networks
- PhD Programme
 - Scholarships available for top students




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
- fpgaConvert



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

1 of 11

Cascade²: Pushing the performance limits of quantisation

Alexandros Kourtellis
Dept. of Electrical and Electronic Eng.
Imperial College London
a.kourtellis@ic.ac.uk

Stylianos I. Venetis
Dept. of Electrical and Electronic Eng.
Imperial College London
stylianos.venetis@ic.ac.uk

Christos-Servos Bouganis
Dept. of Electrical and Electronic Eng.
Imperial College London
christos-servos.bouganis@ic.ac.uk

ABSTRACT
The work presents Cascade/CN, an automated workflow that pushes the quantisation limits of any given CNN model, to produce high-throughput solutions by exploring the competitive trade-off between accuracy and hardware cost. Without the need for re-training, cross-stage architectures tailored for any given FPGA device is generated, consisting of a low- and a high-precision path. A candidate evaluation and re-optimization process then iteratively re-quantizes paths at run-time and forward them to the high-precision and as separate computations. Experiments demonstrate that Cascade/CN achieves a performance level of up to 110% for YOLO-v3 and 40% for ResNet over the baseline design for the same resource budget and accuracy.

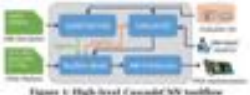


Figure 1: High-level Cascade/CN workflow

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

1 INTRODUCTION
While Convolutional Neural Networks are becoming the de facto

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/yadhwand

Research Assistant/Associa...
The Intelligent Digital Systems ...
Imperial.ac.uk

Aug 4, 2018

Christos @CBouganis
Excited to be presenting at the #ICDL

