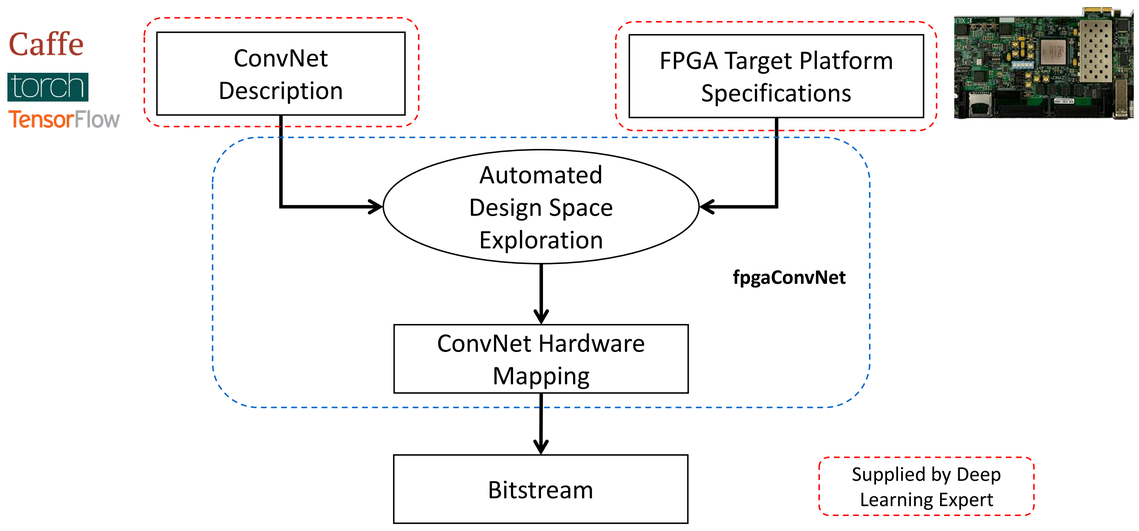
fpgaConvNet: Automated Mapping of CNNs on FPGAs

Stylianos I. Venieris and **Christos-Savvas Bouganis**
Department of Electrical and Electronic Engineering, Imperial College London
{stylianos.venieris10, christos-savvas-bouganis}@imperial.ac.uk

Introduction

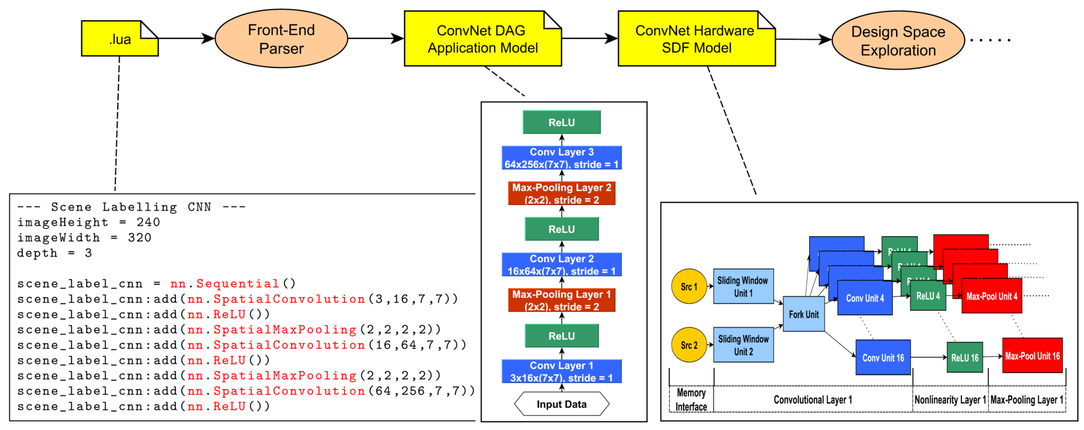
- Convolutional Neural Networks (ConvNets) are state-of-the-art in AI tasks, from object recognition to natural language processing.
- Several frameworks have been released which enable faster experimentation and development of ConvNet models by targeting powerful GPUs.
- FPGAs provide a potential alternative in the form of a low-power, reconfigurable platform with tunable trade-offs between throughput, latency and resource cost.
- fpgaConvNet is an end-to-end tool that aims to bridge the gap between existing Deep Learning toolchains and FPGAs, targeting both high-throughput and low-latency applications [1].

Tool Flow



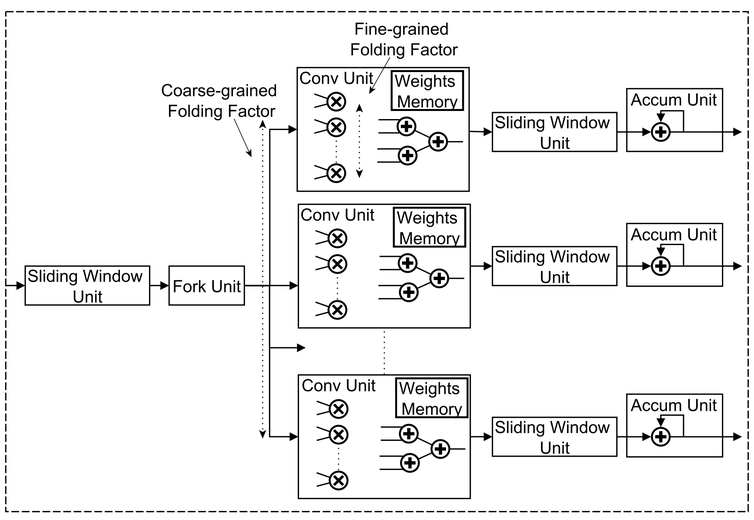
Synchronous Dataflow for ConvNets

- SDF captures computing systems as an SDF graph (SDFG) with nodes representing computations and with arcs in place of data streams between them [2].
- Streaming operation: each node fires whenever data are available at its incoming arcs.
- We developed an SDF-based model to represent ConvNet workloads and hardware design points.
- A hardware design is represented in a compact form by a topology matrix, $\mathbf{\Gamma}$, with one column for each node and one row for each arc.
- Performance and resource models have been developed to estimate the performance-resource characterisitcs given the $\mathbf{\Gamma}$ matrix of a design point.



Mapping Layers to Building Blocks

- Each ConvNet layer is mapped to a sequence of *hardware building blocks*.
- Each block is tunable with respect to its performance-resource characteristics by means of:
  - → *Coarse-grained folding* and
  - → *fine-grained folding* of its units.
- SDF modelling enables the efficient tuning of folding by means of algebraic operations.
- Below, the convolution building block is shown, with its configurable parameters.



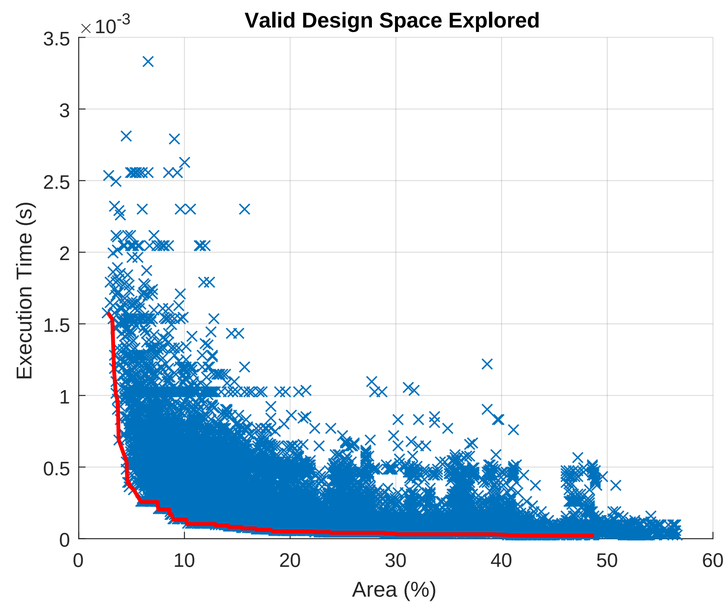
Design Space Exploration and Optimisation

- The design space exploration method uses performance and resource models to traverse the design points described by the tunable parameters of the building blocks.
- A set of four transformations over the SDF model are defined:
  - → Graph Partitioning with Reconfiguration
  - → Coarse-Grained Folding
  - → Fine-Grained Folding
  - → Weights Reloading
- The optimiser operates in two modes by selecting objective function based on the performance metric of interest: throughput or latency. We pose two combinatorial constrained optimisation problems for high-throughput and low-latency applications respectively:

$$\max_{\mathbf{\Gamma}} T(B, \mathbf{\Gamma}), \text{ s.t. } rsc(B, \mathbf{\Gamma}) \leq \boldsymbol{rsc}_{Avail.}$$
$$\min_{\mathbf{\Gamma}} L(1, \mathbf{\Gamma}), \text{ s.t. } rsc(B, \mathbf{\Gamma}) \leq \boldsymbol{rsc}_{Avail.}$$

where T, L and rsc return the throughput in GOp/s, the latency in s/input and the resouce consumption of the current design point, B is the batch size and $\boldsymbol{rsc}_{Avail.}$ is the resource budget of the target FPGA.

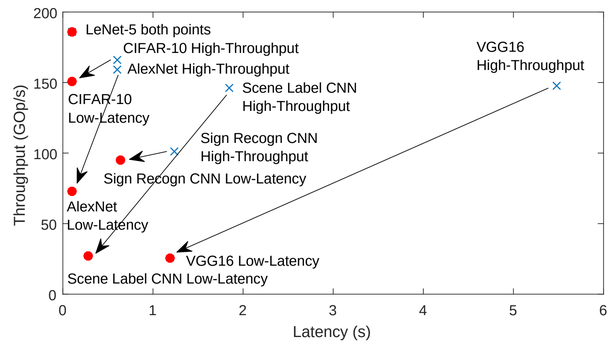
- The optimiser employs all four SDF transformations to traverse the architectural design space and aims to find a design point that approximately optimises the objective function.

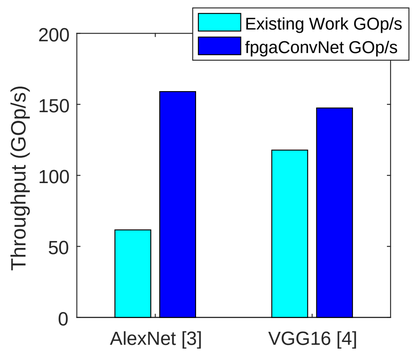


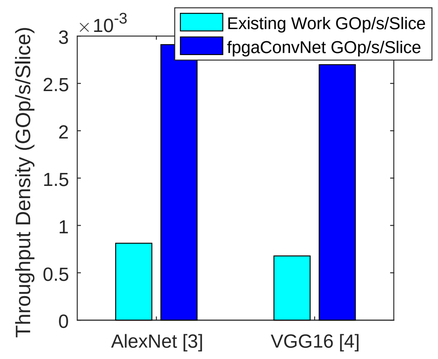
Evaluation

- Throughput-driven optimisation employs batch processing for:
  - → weights reuse across inputs in the batch,
  - → amortisation of the FPGA reconfiguration and reloading of the weights.
- Latency-driven design shifts the generated design points to low-latency regions, with an average latency improvement of 25.41×.
- fpgaConvNet achieves 2.58× and 3.58× higher throughput and throughput density compared to the AlexNet design by Zhang et al. [3]. Moreover, our framework achieves 1.25× and 3.98× higher throughput and throughput density compared to the OpenCL-based VGG16 design by Suda et al. [4].

Model Name	Latency Improvement	Throughput Decrease
LeNet-5	No Change	No Change
CIFAR-10	105.92 ×	1.10 ×
AlexNet	30.95 ×	2.18 ×
Sign Recognition CNN	1.93 ×	1.06 ×
Scene Labelling CNN	6.46 ×	5.47 ×
VGG16	4.62 ×	5.69 ×







References

[1] S. I. Venieris and C.-S. Bouganis, "fpgaConvNet: A Framework for Mapping Convolutional Neural Networks on FPGAs," in *FCCM*. IEEE, 2016, pp. 40–47.

[2] E. A. Lee and D. G. Messerschmitt, "Synchronous Data Flow," *Proceedings of the IEEE*, Sept 1987.

[3] C. Zhang *et al.*, "Optimizing FPGA-based Accelerator Design for Deep Convolutional Neural Networks," in *FPGA*. ACM, 2015, pp. 161–170.

[4] N. Suda *et al.*, "Throughput-Optimized OpenCL-based FPGA Accelerator for Large-Scale Convolutional Neural Networks," in *FPGA*. ACM, 2016, pp. 16–25.