Revisiting ENet for more efficient real-time embedded semantic segmentation on FPGA

Hugo Le Blevec*, Mathieu Léonardon*, Matthieu Arzel*
*IMT Atlantique, Lab-STICC, UMR CNRS 6285, Brest, France**

This work has received a French government support granted to the Labex CominLabs excellence laboratory and managed by the National Research Agency in the "Investing for the Future" program under reference ANR-10-LABX-07-01 from september 2022 to December 2025. In this program, the project associated to this work is LEASARD (Low-Energy deep neural networks for Autonomous Search-And-Rescue Drones).

Motivation

Semantic segmentation is a computer vision task aimed at identifying and separating objects in an image by assigning a label to each pixel.

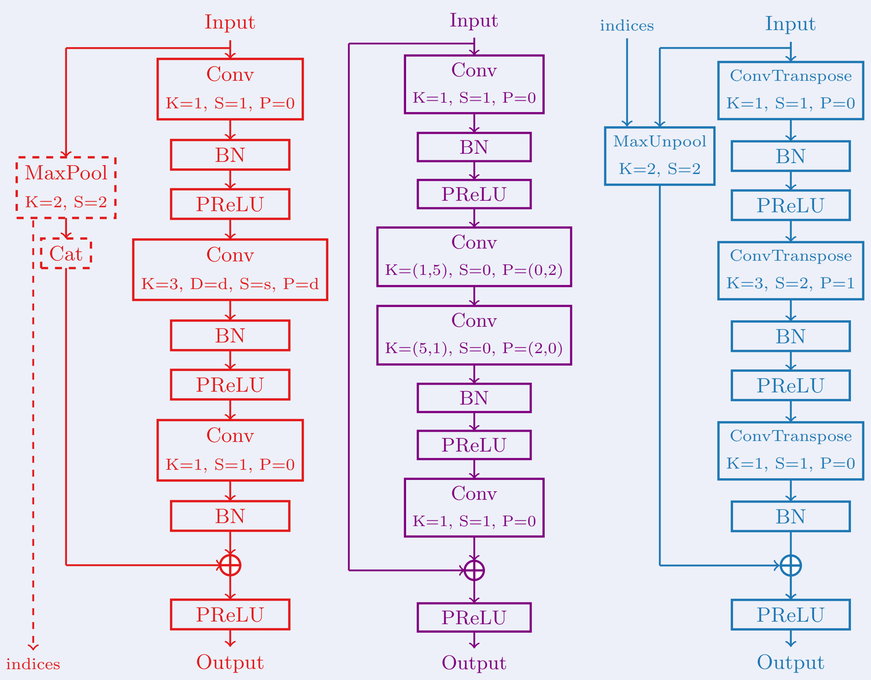
Example image of the Cityscapes dataset with fine-grained pixel annotations [1].

It generally exploits deep neural networks that require high computing power (for example, one or more power-hungry GPUs). FPGAs are very good targets for deploying energy-efficient deep neural networks, but designing the hardware architecture for these networks is a challenge. The use of high-level synthesis (HLS) in adapted *frameworks* like FINN [3] is one solution but brings with it restrictions on the design of neural networks. Among the vast list of models existing in the literature, ENet [2] seems to be a good candidate for FPGA implementation due to its low computations and memory requirements (only 0.37M parameters). However, its segmentation performances are quite low compared to recent models and its structure is not compatible with FINN out-of-the-box. **How to improve and adapt the model to produce a high-performance hardware architecture for real-time semantic segmentation?**

Contributions

We helped integrate the support for fractionally-strided convolutions in FINN, an algorithm that implements transposed convolutions as a pixel padding followed by a regular convolution. We designed a new custom operator named *FMPadding_Batch*, which inserts a number zeros between the pixels of the input depending on the stride, and a transformation that maps a transposed convolution to $\{FMPadding\_Batch \rightarrow Im2Col \rightarrow Matmul\}$, which can then be mapped to $FMPadding\_Batch + MVAU$. All these contributions are now part of FINN's GitHub main branch.

Baseline ENEt Architecture



Name	Type	Output size
initial	-	$16 \times 256 \times 128$
RDDneck	downsampling	$64 \times 128 \times 64$
4x RDDneck	-	$64 \times 128 \times 64$
RDDneck	downsampling	$128 \times 64 \times 32$
RDDneck	-	$128 \times 64 \times 32$
RDDneck	dilation 2	$128 \times 64 \times 32$
ASNeck	-	$128 \times 64 \times 32$
RDDneck	dilation 4	$128 \times 64 \times 32$
RDDneck	-	$128 \times 64 \times 32$
RDDneck	dilation 8	$128 \times 64 \times 32$
ASNeck	-	$128 \times 64 \times 32$
RDDneck	dilation 16	$128 \times 64 \times 32$
Repeat previous section, without first RDDneck		
UBNeck	-	$64 \times 128 \times 64$
RDDneck	-	$64 \times 128 \times 64$
RDDneck	-	$64 \times 128 \times 64$
UBNeck	-	$16 \times 256 \times 128$
RDDneck	-	$16 \times 256 \times 128$
Deconvolution	-	$C \times 512 \times 256$

Training Setup

In an attempt to see whether the use of a more recent and optimized training procedure would improve network performance, we trained the network using the same training settings as RegSeg [5]. We trained over 1000 epochs using SGD, poly learning rate scheduler with a linear warm-up, a various set of data augmentation techniques, and class uniform sampling. Using these settings, we were able to reach **70.5%** mIoU on the Cityscapes dataset, which is a notable increase (far from the original 58.3%) and shows the importance of having a **good training procedure**.

Training & Design Space Exploration

To make the network compatible with FINN, we performed the following modifications:

- MaxPooling-Padding → Convolution with a 1x1 kernel.
- MaxUnpooling → Transposed Convolution.
- Removing long bypasses between RDDNeck and UBNeck.
- PReLU → ReLU moved after elementwise add.

These modifications increase the number of parameters by 38.6% and the number of MACs by 136.7%, and achieve **69.75%** mIoU with INT4 quantization, obtained through QAT.

In an attempt to find a more optimized network, we explored the neural network design space through different modifications of the architecture:

- Reintroducing one or two long bypasses.
- Increasing the projection ratio, reducing the number of channels of hidden layers.
- Replacing transposed convolutions by nearest-neighbor upsampling.

The best version was obtained with nearest-neighbor upsampling, which does not change the mIoU, but only needs 0.35M parameters and 1.11G MACs, which is less than the baseline.

Implementations results

We implemented the updated model on the Xilinx ZU19EG FPGA embedded on the TySOM-3A-ZU19EG board using FINN. Our design is able to run with a 300 MHz clock and reaches **126 FPS** throughput with **8ms** latency on 512x256 inputs, and is 1.15× more power efficient than an Nvidia Jetson AGX Xavier GPU achieving 158 FPS.

Resource utilization of our design on Xilinx ZCU19EG FPGA. Negligible amounts of URAM and DSP are used for the IODMA interfaces.

Resource	Utilization	Utilization %
LUT	232,930	44.6
FF	186,646	17.85
BRAM (18K)	929	47.21

Discussions and future work

- FINN is a great tool for embedded neural networks design space exploration, giving estimations of throughput and resources that help rapidly find better design points.
- Surprisingly, transposed convolutions do not perform better than nearest-neighbor interpolation, but require more resources.
- With added support in FINN of other types of convolutions, like grouped convolutions, we could try different design points.
- The placement of FIFOs and choices of folding parameters is partly manual and probably not optimal.
- Exploring ways to work with larger image sizes, or training the network to adapt to lower input size.

References

[1] Cordts, Marius et al.: *The Cityscapes Dataset for Semantic Urban Scene Understanding*, CVPR (2016).
[2] Paszke, Adam et al.: *ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation*, arXiv (2021).
[3] Umuroglu, Yaman et al.: *FINN: A Framework for Fast, Scalable Binarized Neural Network Inference*, ACM-ISFPGA (2017).
[4] Alessandro Pappalardo.: *Xilinx/brevitas*, Zenodo (2023).
[5] Gao, Roland: *Rethinking Dilated Convolution for Real-time Semantic Segmentation*, arXiv (2023).