Practical Aspects of FPGA Implementation of Neural Network for Image Classification Based on Learned Separable Transform

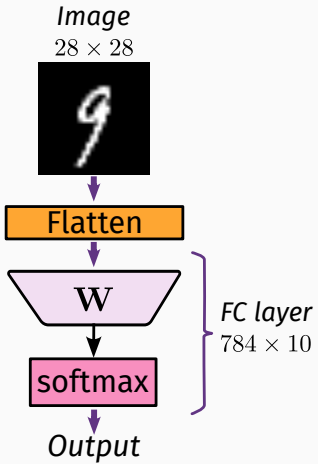
Egor Krivalcevich Maxim Vashkevich
krivalcevi4.egor@gmail.com, vashkevich@bsuir.by

Belarusian State University of Informatics and Radioelectronics
Minsk, Belarus

17th International Conference on Pattern Recognition and Information Processing

Intro

- Goal: the development of a resource-efficient hardware architecture for FPGA-based image recognition.
- A simple single-layer neural network (7850 parameters) allows achieving a relatively low accuracy of 92.5
- When adding hidden layers, the number of network parameters increases rapidly



Two-dimensional learnable separable transform (LST)

Two-dimensional separable transform

- Two-dimensional separable transforms are used in image processing to reduce the computational complexity of spatial filtering. The transform kernel has the form:

$$\mathbf{W} = \mathbf{v} \times \mathbf{h}^T,$$

  where $\mathbf{W} \in \mathbb{R}^{n \times n}$, $\mathbf{v}, \mathbf{h} \in \mathbb{R}^{n \times 1}$.
- The separable transform $\mathbf{W}$ has $2n$ independent parameters, instead of n^2 parameters that the usual transform has.
- An example of a separable transform is the Sobel filter:

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}.$$

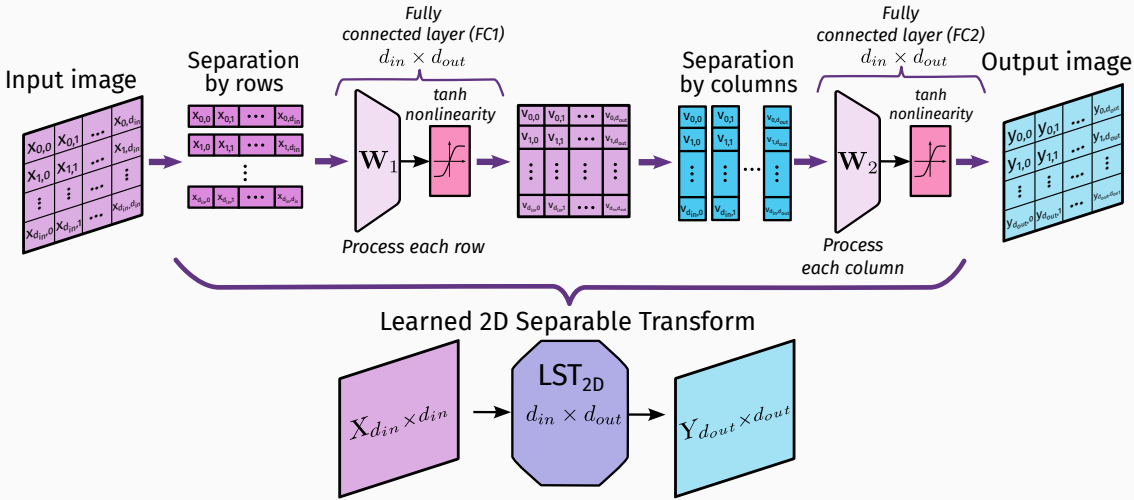
Two-dimensional learnable separable transform (LST)

- The proposed learned separable transform (LST) processes an image first row-wise and then column-wise.
- The LST processes an image $\mathbf{X}$ of size $d_{in} \times d_{in}$ and outputs an image $\mathbf{Y}$ of size $d_{out} \times d_{out}$:

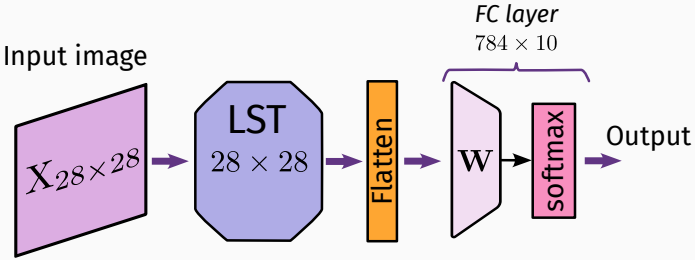
$$\mathbf{Y} = \mathrm{LST}_{d_{in} \times d_{out}}(\mathbf{X}) = \tanh(\mathbf{W}_2 \tanh(\mathbf{W}_1 \mathbf{X}^T)),$$

  where $\mathbf{W}_1$, $\mathbf{W}_2$ are the weight matrices of fullyconnected layers (FC1 and FC2), d_{out} is the hyperparameter of the transform that determine the total number of learnable parameters $N_{params} = 2 \times (d_{in} + 1) \times d_{out}$.

Two-dimensional learnable separable transform (LST)



Model LST-1



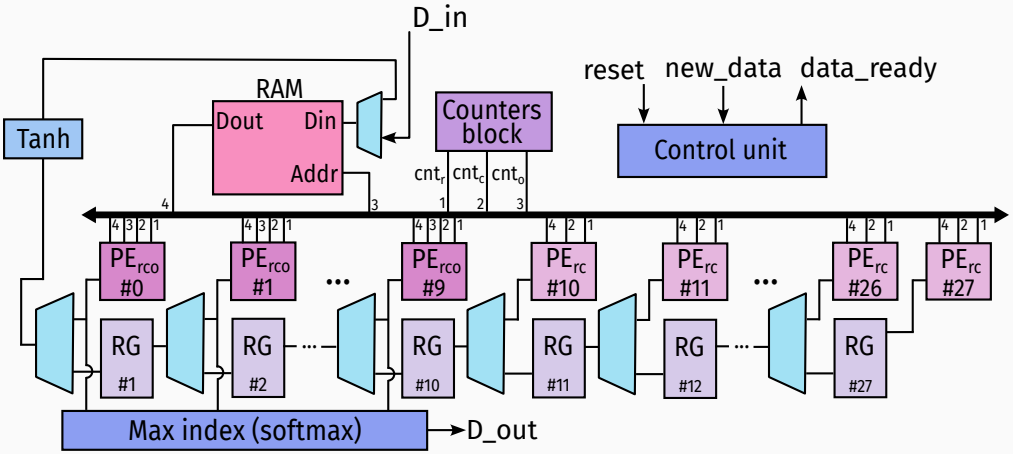
- LST can be viewed as a basic building block for constructing compact neural networks for image recognition.
- Number of parameters of LST-1 model:

$$N_{params} = 2(d_{in} + 1) \times d_{out} + (d_{out}^2 + 1) \times 10$$

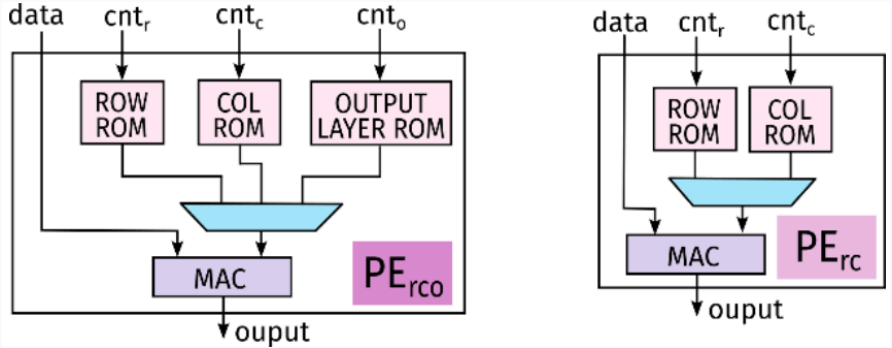
- For $d_{in} = d_{out}$ = 28 number of model parameters N_{params} = 9 474.

Implementing the LST-1 neural network on FPGA

Implementation of LST-1 on FPGA



Implementing the LST-1 neural network on FPGA



- The LST-1 implementation includes 10 PE_{rco} blocks and 18 PE_{rc} blocks
- At the first stage of the LST calculation, all PE processing elements are used. In each PE, one column of the $\mathbf{W}_1$ matrix is stored in the “ROW ROM” memory, and the columns of the $\mathbf{W}_2$ matrix are stored in the “COL ROM” memory.

Implementing the LST-1 neural network on FPGA

- The LST-1 processor is described in the SystemVerilog language and implemented on the Xilinx Zybo Z7 board (FPGA XC7Z010)
- The Linux PYNQ distribution was used to organize the testing process, which was launched on the ARM core of the XC7Z010 crystal.
- The LST-1 processor was implemented as an IP core using a 12-bit representation of numbers.

Block type	Used	Available	Usage, %
LUT as logic	1302	17600	7.4
Flip Flop	1461	35200	4.15
BRAM	33.5	120	55.83
DSP	57	80	71.25

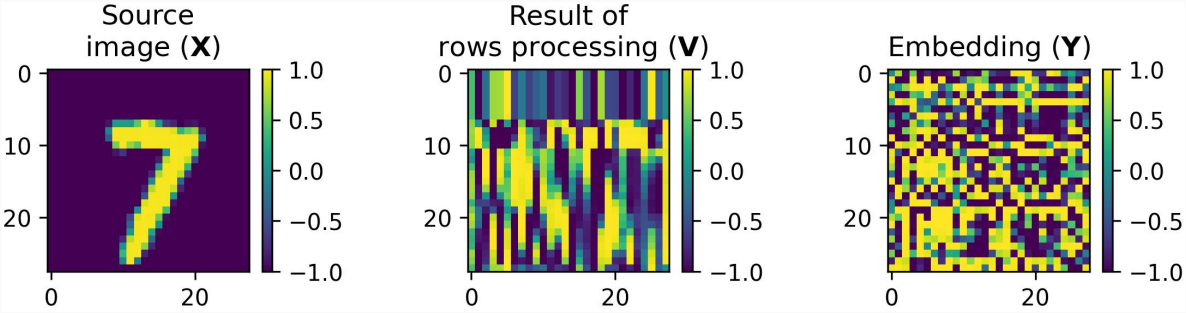
Experiment and results

Experiment and results

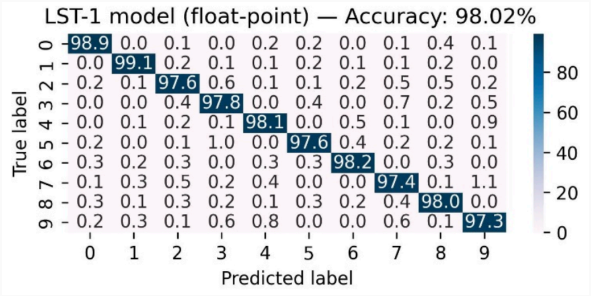
- MNIST dataset (70k images of handwritten digits of size 28 × 28)
- Initialization of model weights was performed using the Xavier method
- Objective function – negative logarithmic likelihood (`torch.nn.NLLLoss`)
- Training was performed using the Adam algorithm (learning rate $\eta = 2 \times 10^{-3}$, number of epochs – 300, batch size – 1000)
- The metric – accuracy was used to evaluate the quality of recognition

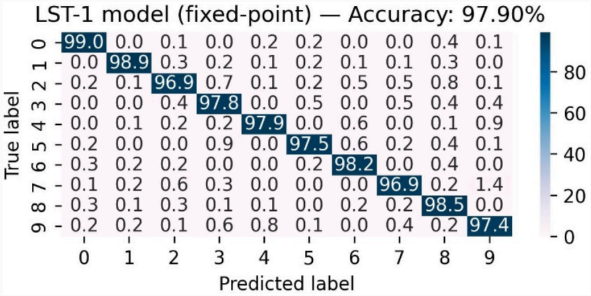
Experiment and results

- The LST-1 model encodes an image as an irregular QR-code-like pattern.



Experiment and results





- The confusion matrices obtained from the floating-point model and FPGA implementation, which aligns with the matrix generated by the Python-based fixed-point model. The overall accuracy of the LST-1 model with weights quantized into Q6.7 format is 97.9

Experiment and results

Authors & ref.	DNN architecture	Model size	Accuracy, %
Medus [1]	784-600-600-10	891 610	98.63%
Samragh [2]	784-512-512-10	670 208	98.40%
Huynh [3]	784-126-126-10	115 920	98.16%
Huynh [3]	784-40-40-40-10	34 960	97.20%
Westby [4]	784-12-10	9 550	93.25%
proposed	LST-1	9 474	98.02%

[1] M. Samragh, “Customizing neural networks for efficient FPGA implementation”, 2017
[2] L.D. Medus, “A novel systolic parallel hardware architecture for the FPGA acceleration of feedforward neural networks”, 2019.
[3] T.V. Huynh, “Deep neural network accelerator based on FPGA”, 2017
[4] I. Westby, et al. “FPGA acceleration on a multilayer perceptron neural network for digit recognition”, 2021

Conclusions

- A two-dimensional trainable separable transform is proposed, which can be used as a basic block for constructing compact neural networks for image recognition
- The LST-1 model attains a recognition accuracy exceeding 98% for handwritten digits using only 9.4 thousand parameters, demonstrating a highly efficient architecture
- A hardware architecture for implementing the LST-1 model based on FPGA is proposed