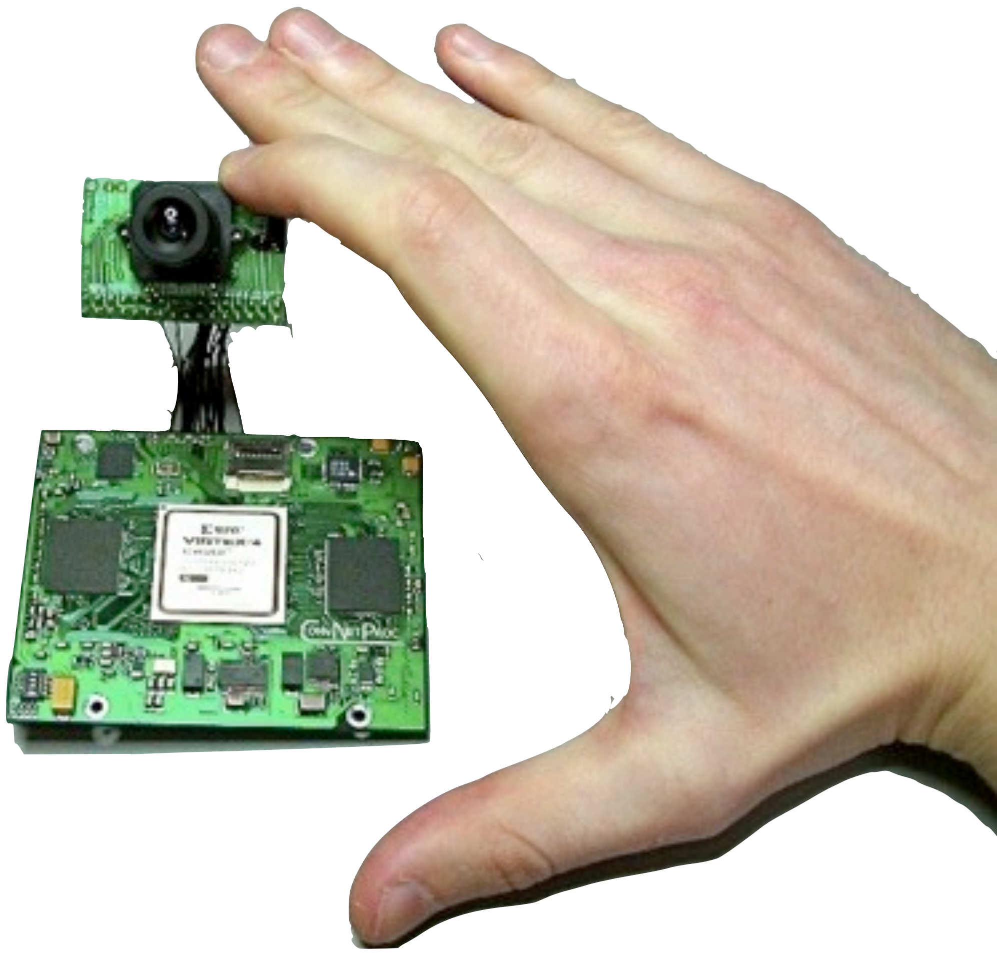
neuFlow: A Dataflow Architecture for Vision

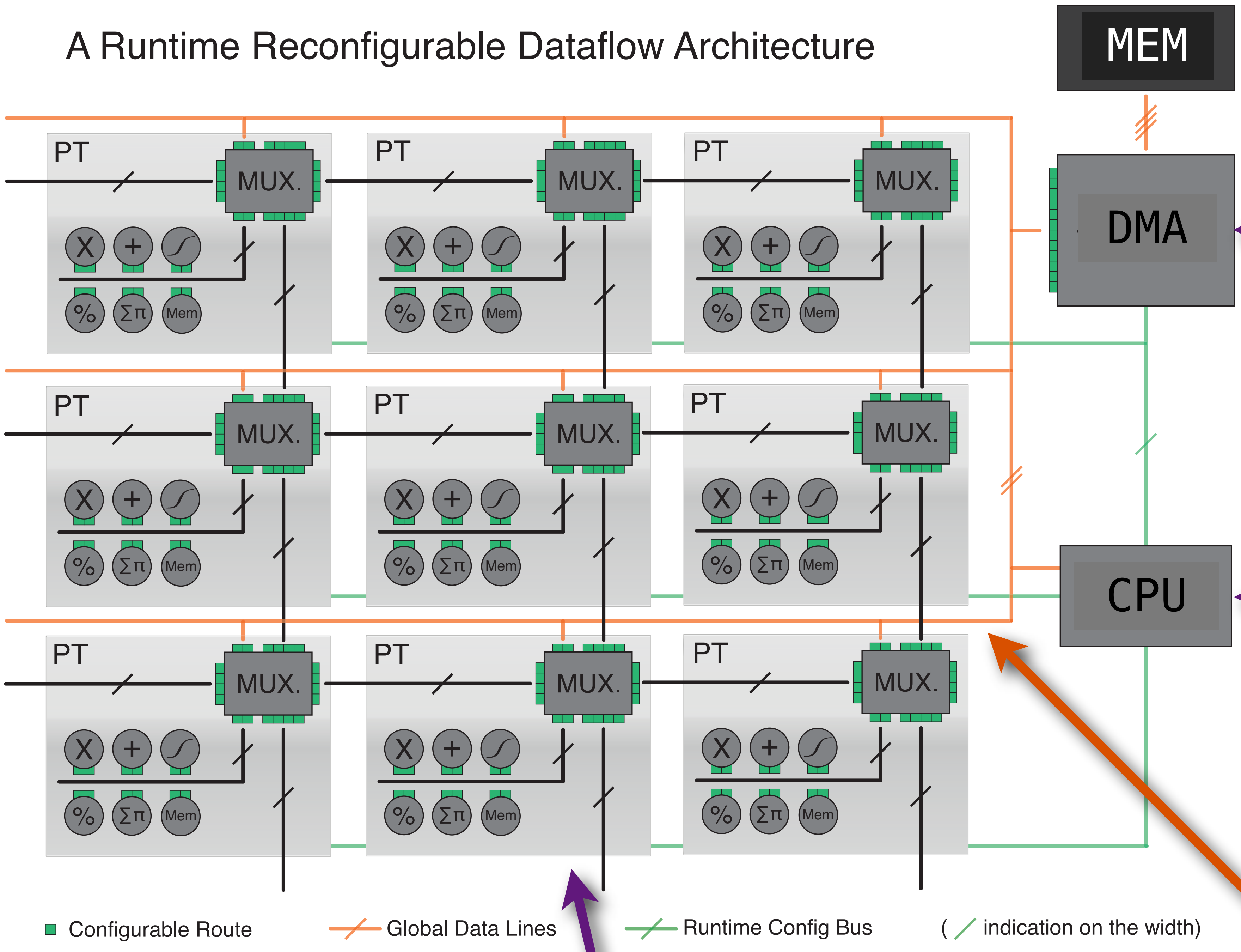
Clément Farabet, Yann LeCun



joint work with:
Yann LeCun, Laurent Najman, Marco Scoffier, Srinivas Turaga
Eugenio Culurciello, Berin Martini, Polina Akselrod, Darko Jelaca,

neuFlow: Architecture

A Runtime Reconfigurable Dataflow Architecture



multi-port memory controller (DMA)

[x12 on a V6 LX240T]

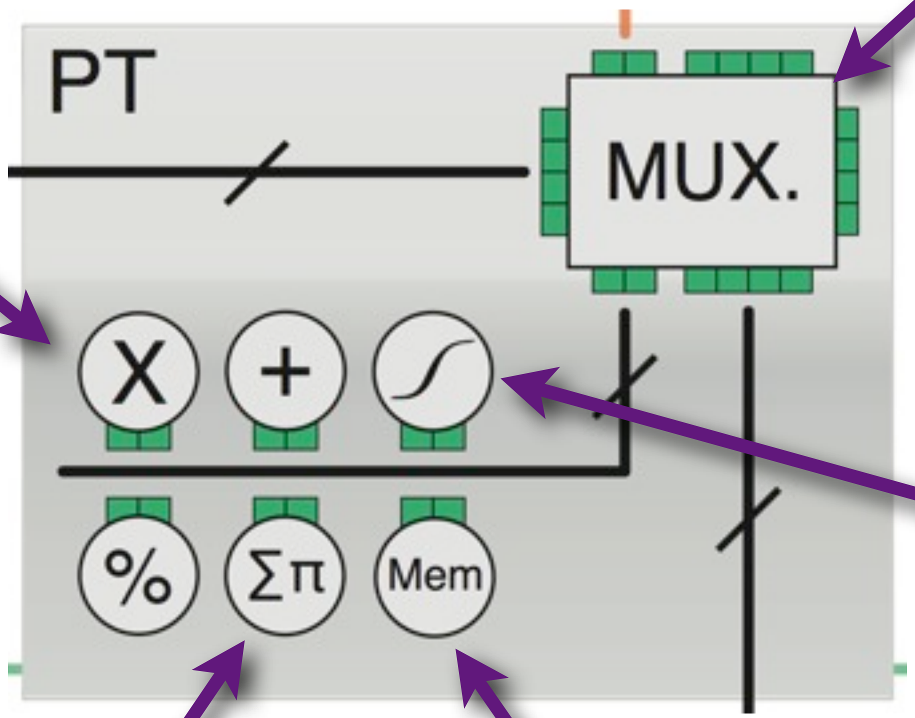
RISC CPU, to reconfigure tiles and data paths, at runtime

grid of passive processing tiles (PTs)

[x20 on a Virtex6 LX240T]

global network-on-chip to allow fast reconfiguration

neuFlow: Processing Tile (PT) Structure



term-by-term streaming operators (MUL,DIV,ADD, SUB,MAX)

[x8,2 per tile]

configurable router, to stream data in and out of the tile, to neighbors or DMA ports

[x20]

configurable piece-wise linear or quadratic mapper

[x4]

full 1/2D parallel convolver with 100 MAC units

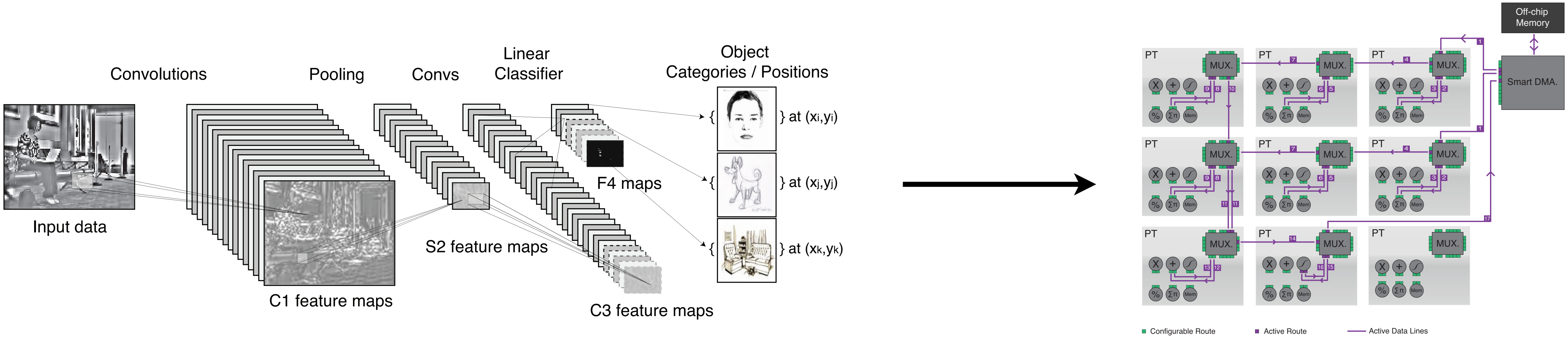
[x4]

configurable bank of FIFOs , for stream buffering, up to 10kB per PT

[x8]

[Virtex6 LX240T]

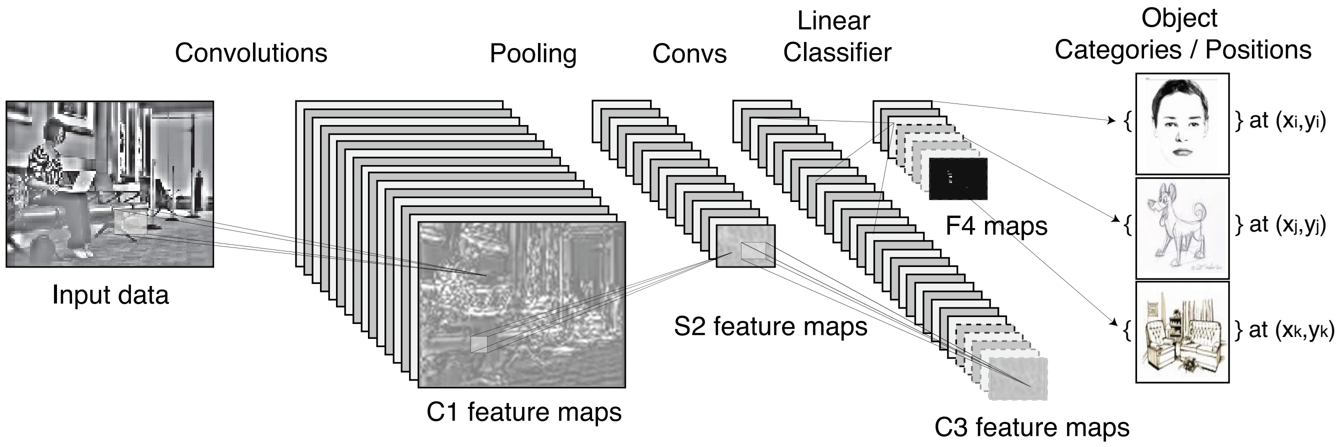
luaFlow: A Dataflow Compiler



a home-grown compiler that compiles ConvNets and the likes to sequences of grid reconfigurations
(e.g. neuFlow bytecode)

luaFlow: A Dataflow Compiler

1/5

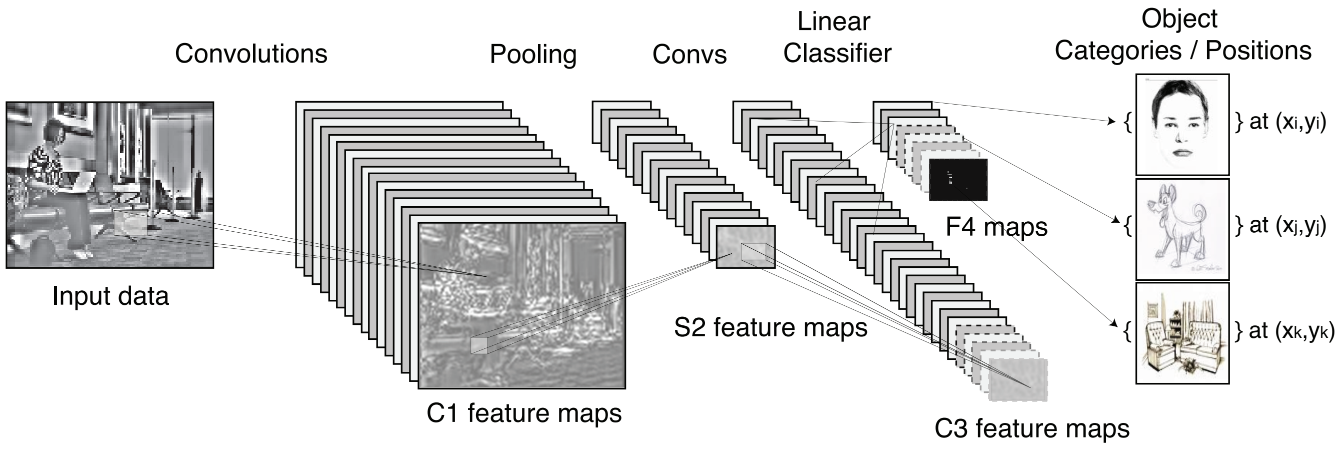


high-level (functional) description

```
net = nn.Sequential()
net:add(nn.SpatialConvolution(3,6,9,9))
net:add(nn.Tanh())
net:add(nn.SpatialSubSampling(6,4,4))
net:add(nn.SpatialConvolution(6,12,9,9))
net:add(nn.SpatialLinear(12,6))
```

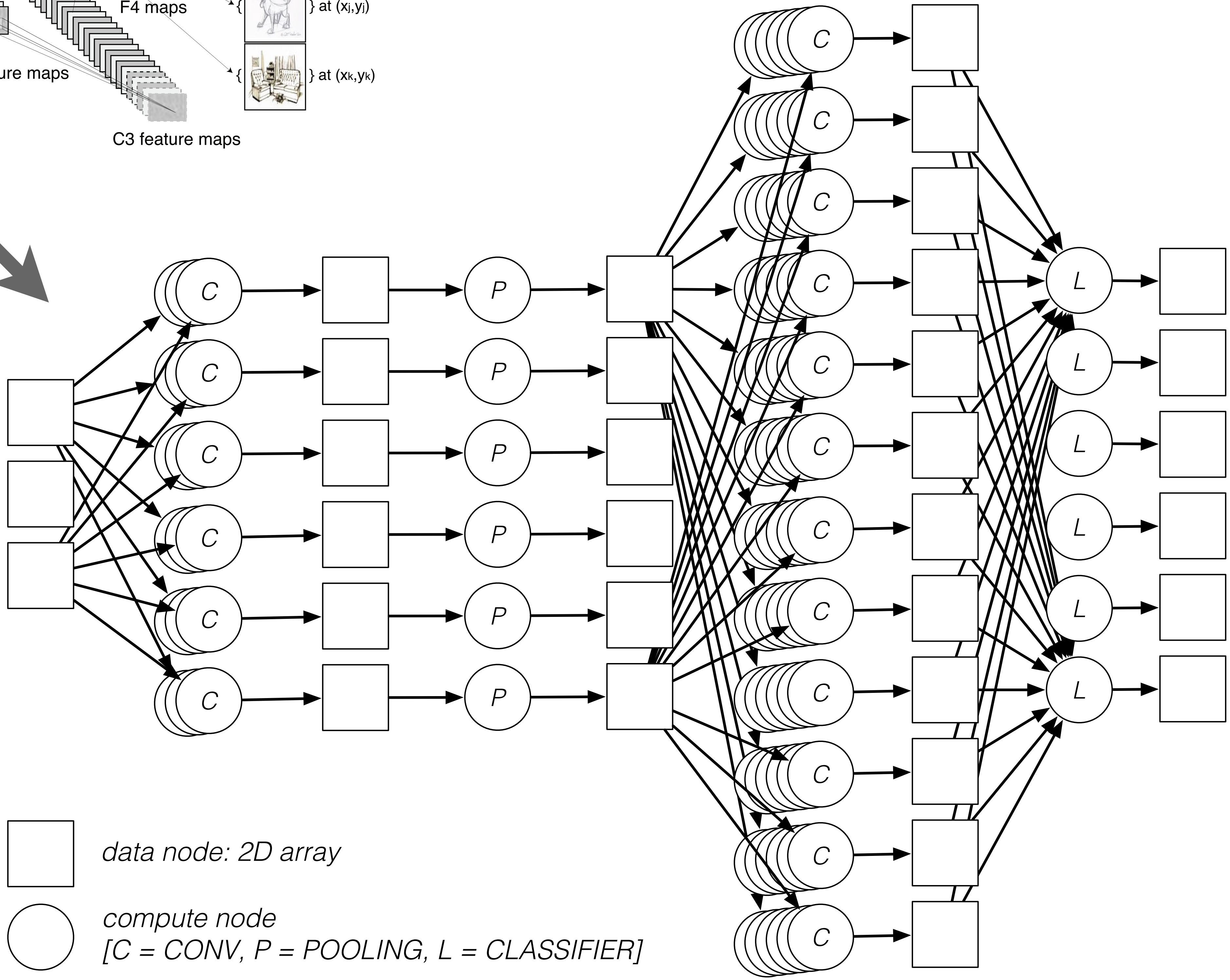
(Torch5 code)

luaFlow: A Dataflow Compiler



2/5

infer a flow-graph model from the user description



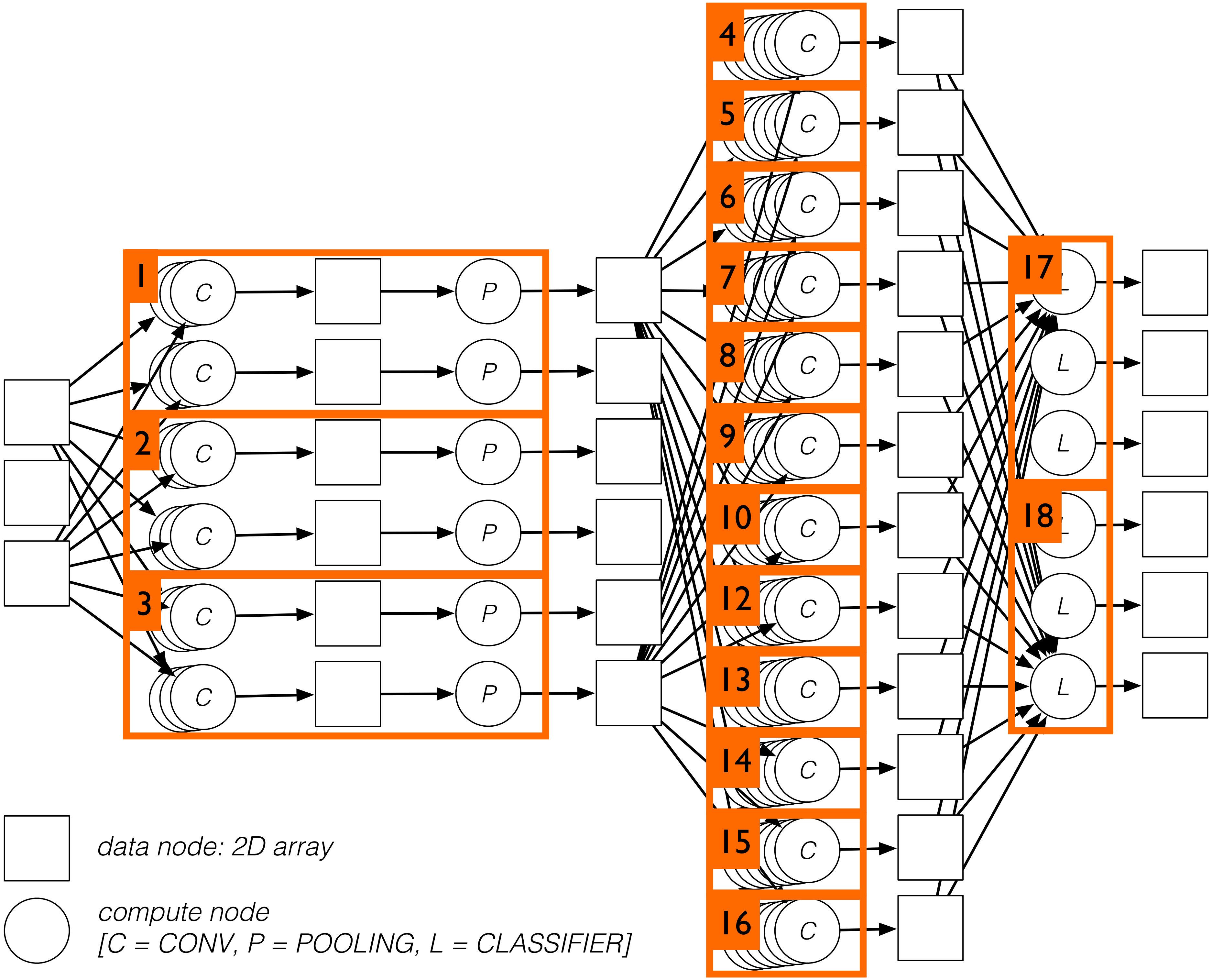
data node: 2D array

compute node
[C = CONV, P = POOLING, L = CLASSIFIER]

luaFlow: A Dataflow Compiler

3/5

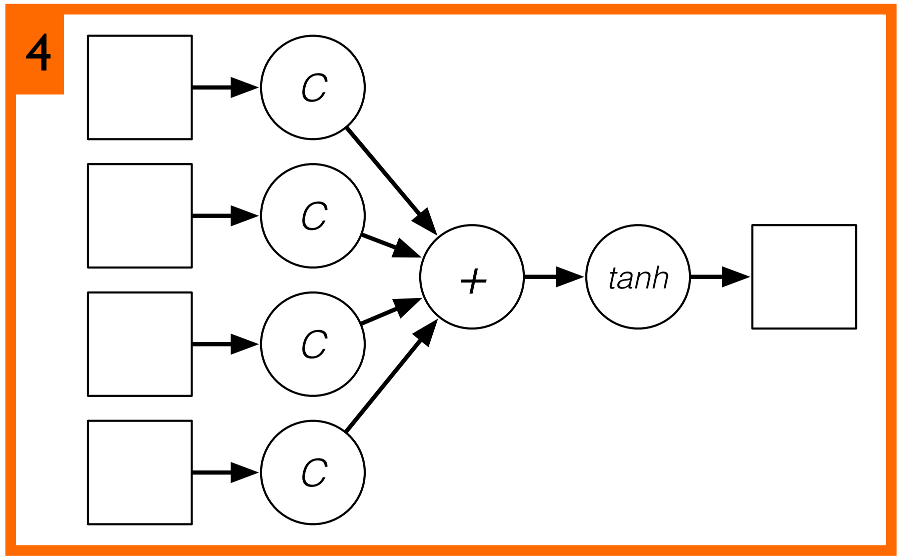
divide the graph into subgraphs that fit on the grid

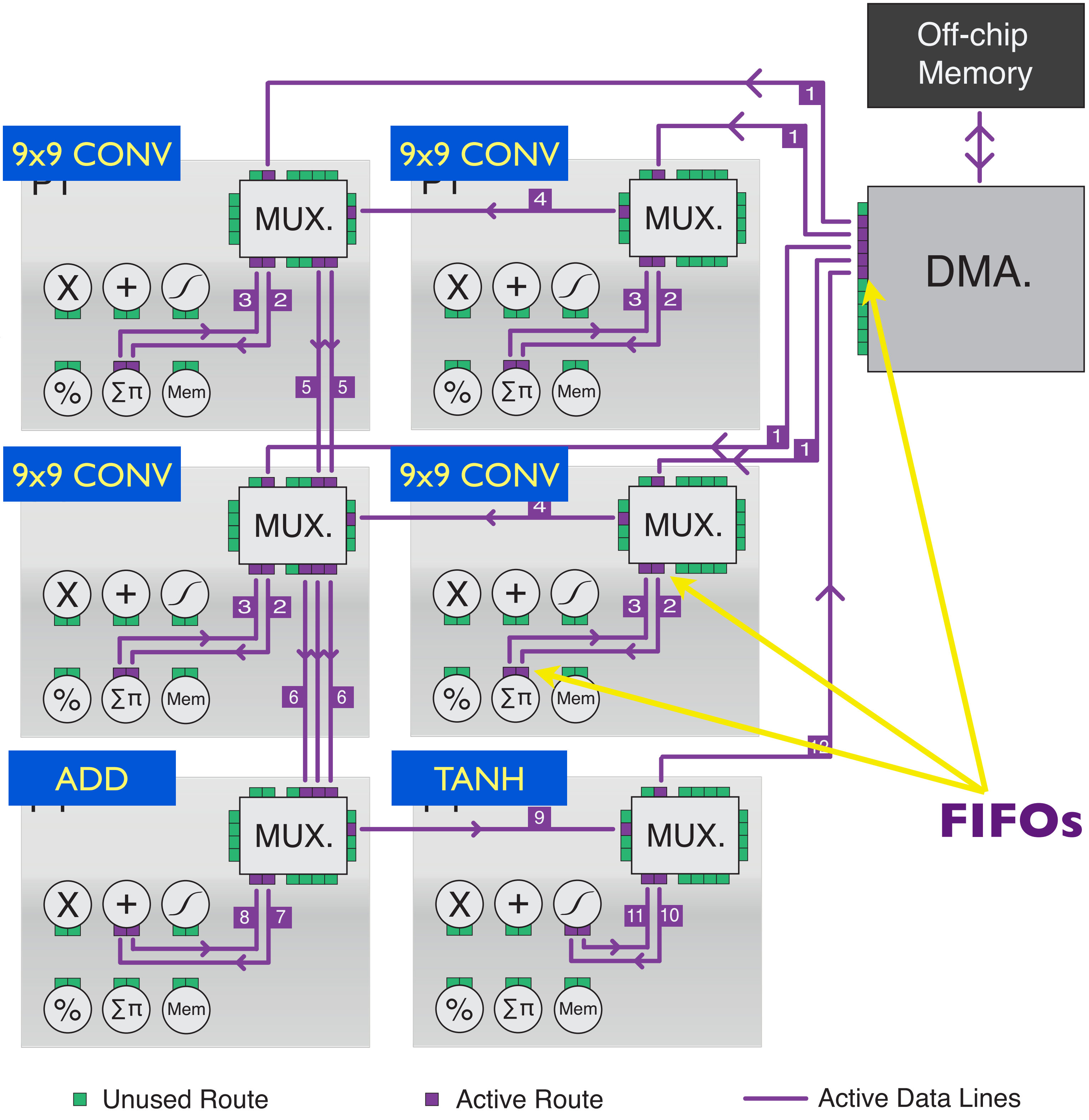


data node: 2D array

compute node
[C = CONV, P = POOLING, L = CLASSIFIER]

4/5 for each subgraph, generate the routes and configs for each PT and DMA port





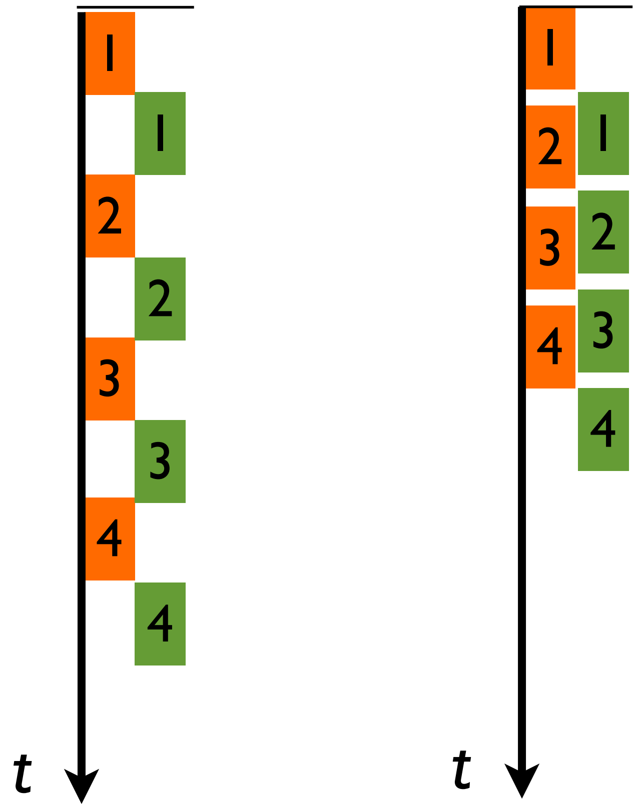
once configured, data streams ripple through the grid,

the grid is “passive”

luaFlow: A Dataflow Compiler

5/5

global optimization: instruction reordering



n configuration cycles

n data streaming cycles

luaFlow: Supported Operations

Coding: Q8.8 (16bit, fixed-point)

- 1D convolution
- 2D convolution
- local pooling/subsampling/histogramming (max,average,weighted)
- term-by-term div/add/sub/mul/muladd
- point-wise non-linear mapping
- local contrast normalization
- temporal difference
- ...

Profiling*

	Intel 2Core	neuFlow Virtex4	neuFlow Virtex 6	nVidia GT335m	neuFlow IBM 45nm	nVidia GTX480
Peak GOP/sec	10?	40	160	182	1280	1350
Actual GOP/sec	1.1	37	147	54	1164	294
FPS	1.4	46	182	67	1456	374
Power (W)	30	10	10	30	5	220
Embed? (GOP/s/W)	0.03667	3.7	14.7	1.8	232.8	1.33636

* computing a 16x10x10 filter bank over a 4x500x500 input image

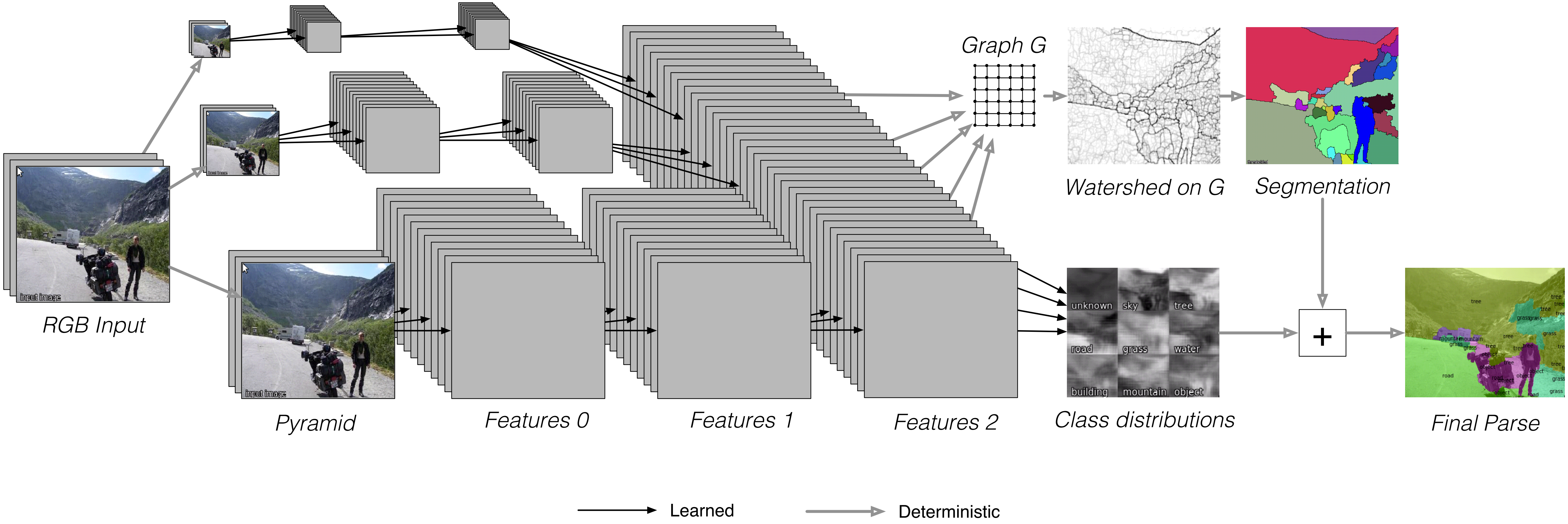
Resources

	neuFlow Virtex4	neuFlow Virtex 6	neuFlow IBM 45nm 3x3mm	neuFlow IBM 45nm 6x6mm
Peak GOP/sec	40	160	320	1280
Sys+DDR Frequency MHz	200	200	400	400
DDR Bdwdth GB/s (pins)	0.8 (16)	3 (64)	6 (64)	24 (256)
MACs #avail (#used)	192 (109)	680 (436)	436 (all)	1744 (all)
Tiles #avail	4	20	20	80

Application: Scene Parsing

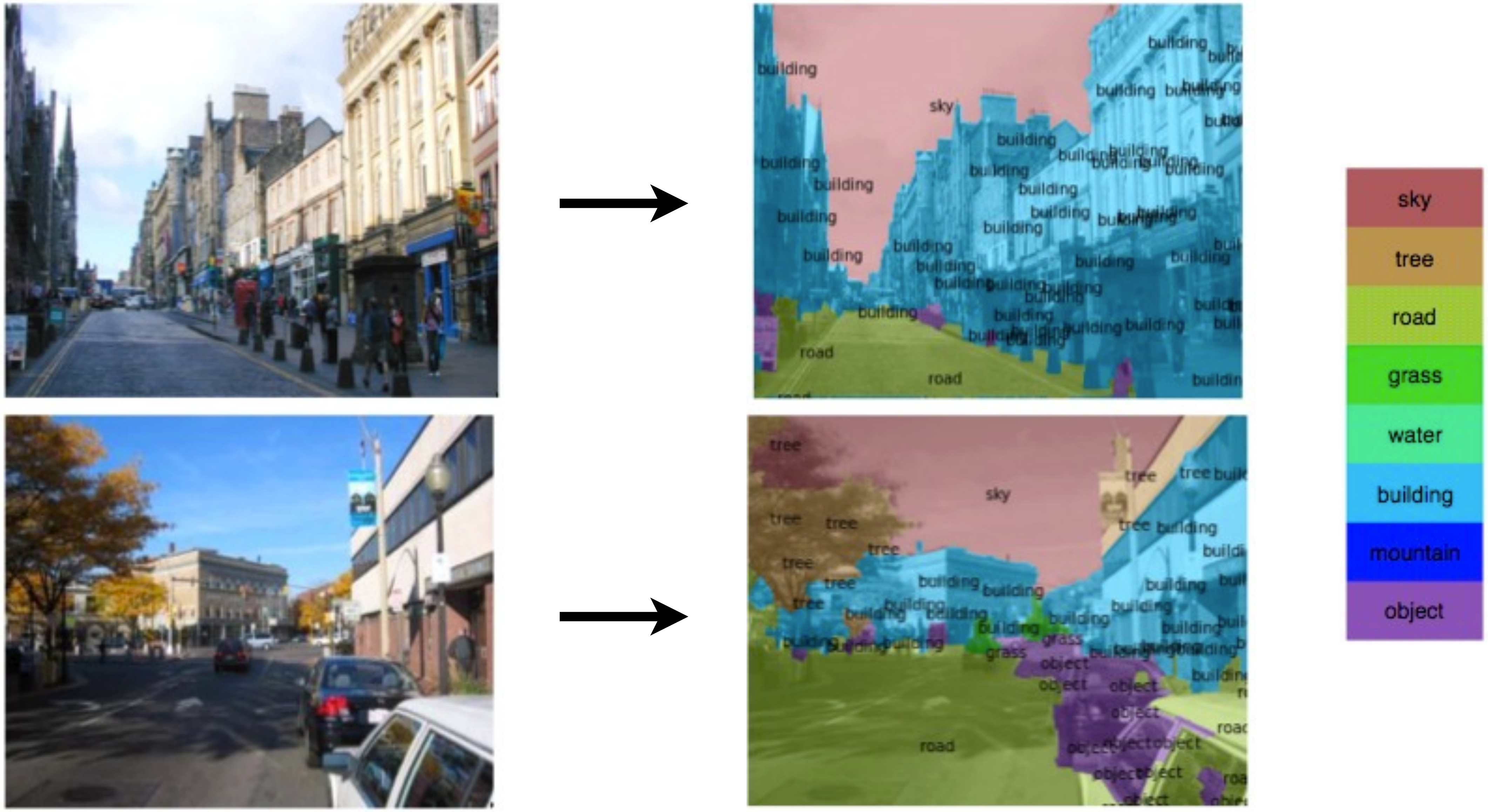
dense labeling of
natural images

Application: Scene Parsing

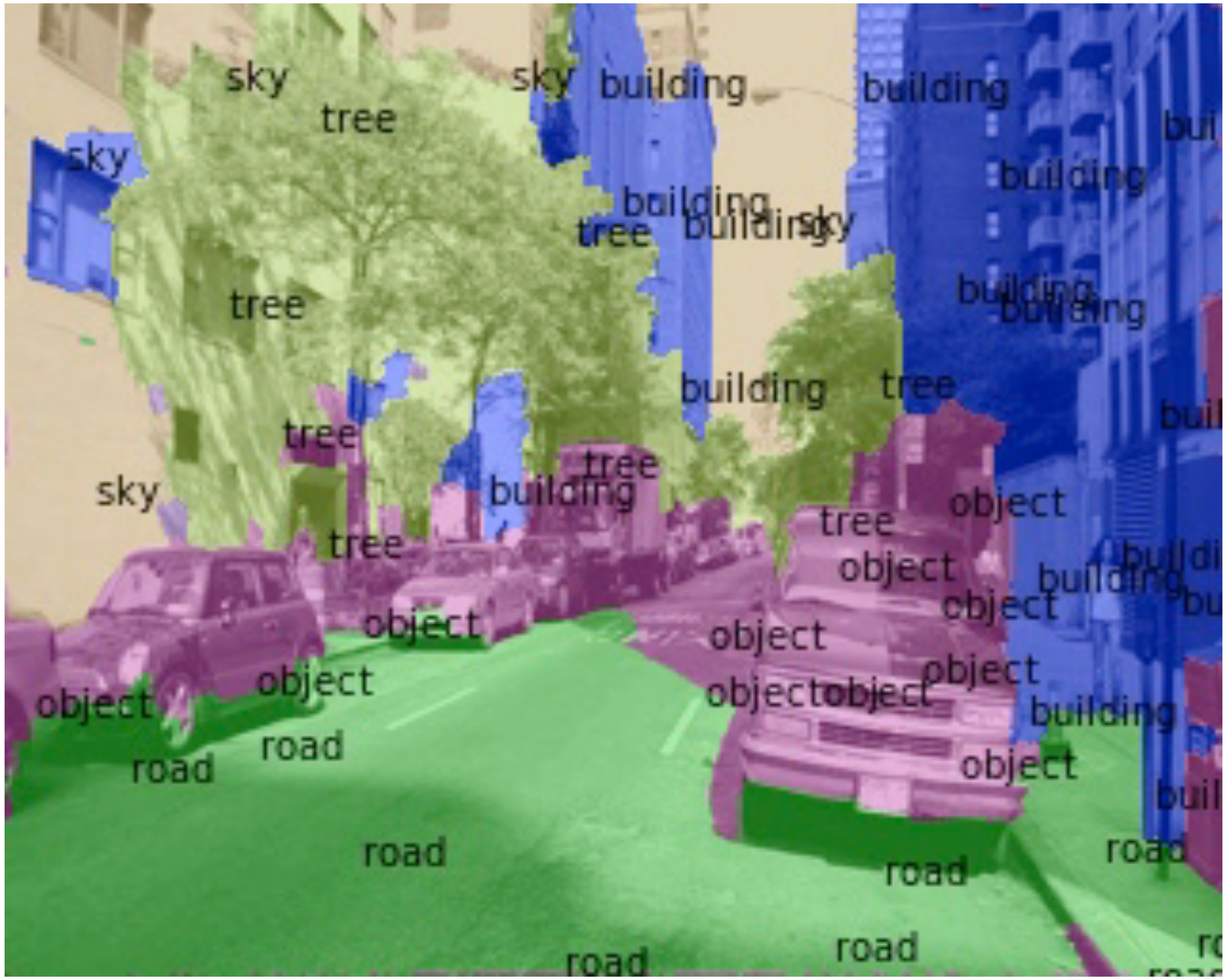


multiscale ConvNet, trained end-to-end to optimize a dual term energy: a segmentation loss and a pixelwise classification loss

Application: Scene Parsing

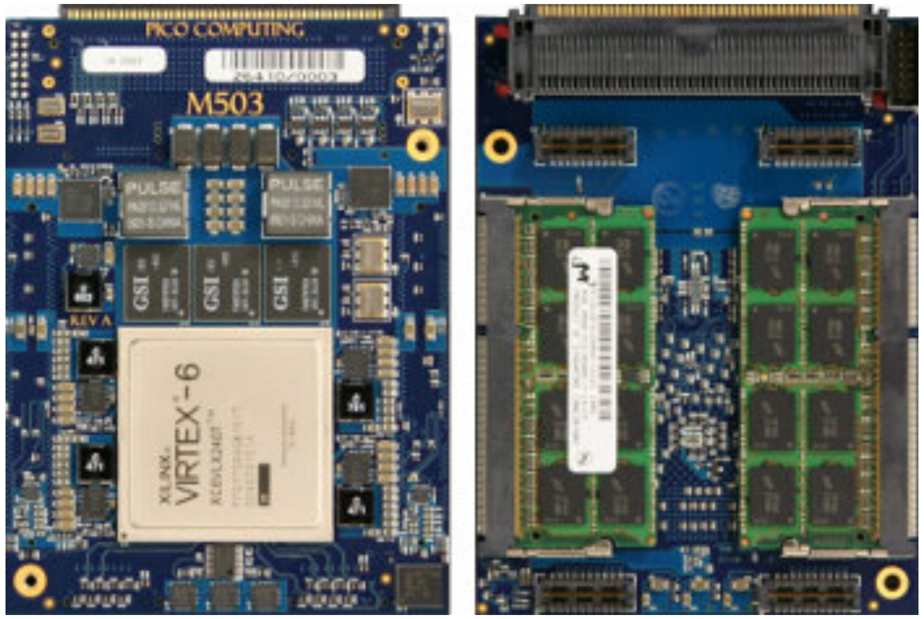


Application: Scene Parsing



Live Demo.

thank you



www.neuflow.org