Towards Executing Computer Vision Functionality on Programmable Network Devices

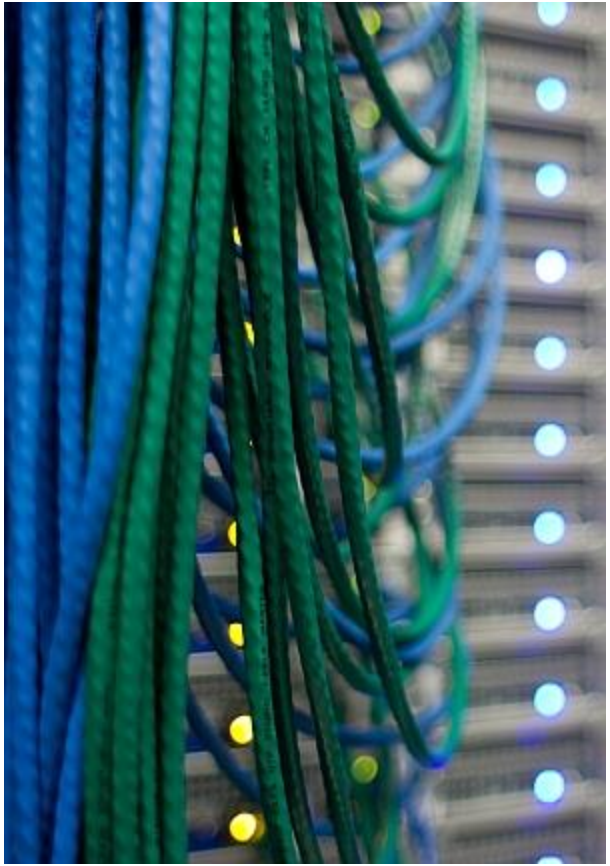
René Glebke, Johannes Krude, Ike Kunze,
Jan Rüth, Felix Senger, Klaus Wehrle

2. KuVS Fachgespräch on Network Softwarization, Online / Zoom, 2020-04-02
First presented at ACM CoNEXT ENCP'19, Orlando, FL, 2019-12-09

https://www.comsys.rwth-aachen.de/

Paper available at
https://www.comsys.rwth-aachen.de/fileadmin/papers/2019/2019-glebke-in-network-cv.pdf

In-Network Processing: State-of-the-art applications

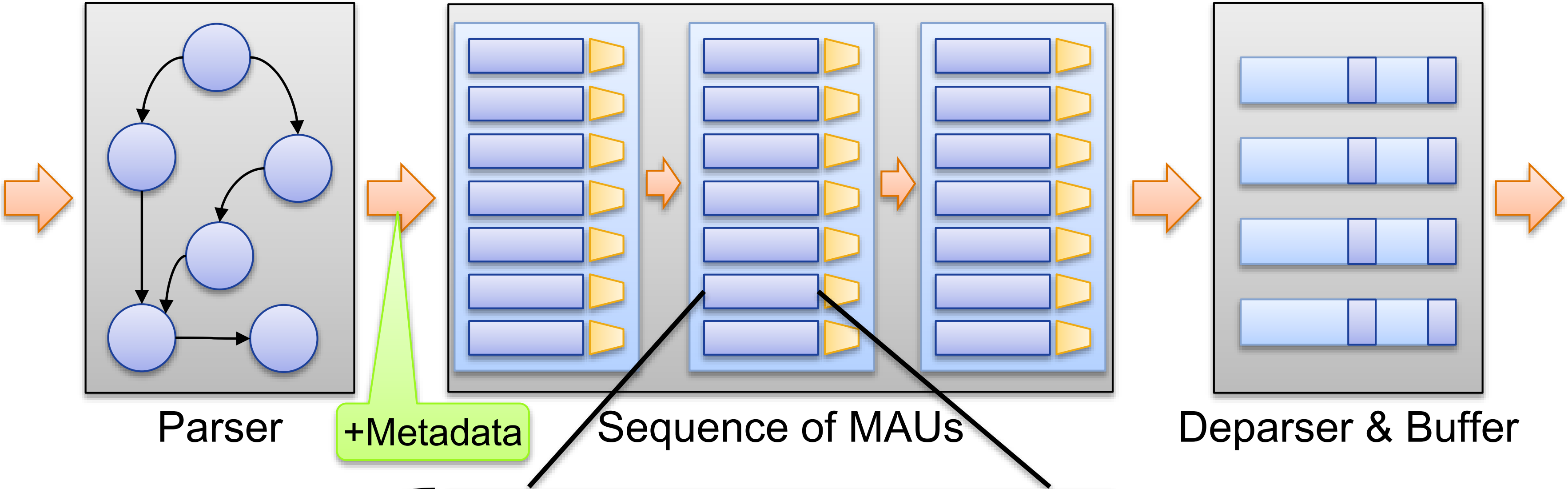


- **Programmable data planes enable scenarios that require low latencies & high bandwidths**
  - Network operation & management
    - Load balancing
    - Heavy-hitter identification
    - DDoS mitigation
  - Distributed algorithms & databases
    - Consensus
    - Key-value caching
    - MapReduce
  - Partial offloading of application logic
    - Industrial feedback control

Common pattern in most current scenarios:
Few individual operations on **many small** items

René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

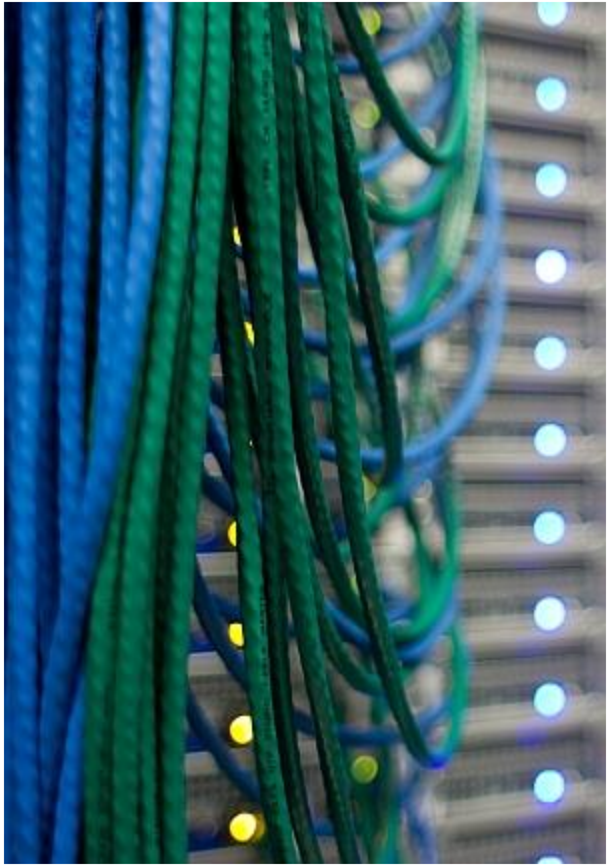
- **Match-action pipeline allows fixed-latency processing of packets**



Architecture-dependent:

- Exact/range/ternary table matches
- Header (section), metadata rewriting
- Limited arithmetic, no floating point

René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de



- **Programmable data planes enable scenarios that require low latencies & high bandwidths**
  - Network operation & management
    - Load balancing
    - Heavy-hitter identification
    - DDoS detection
  - Distributed algorithms & databases
    - Consensus
    - Key-value caching
    - MapReduce
  - Partial offloading of application logic
    - Industrial feedback control

Common pattern in most current scenarios:
Few individual operations on **many small** items

Can other scenarios also benefit from INP?

René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

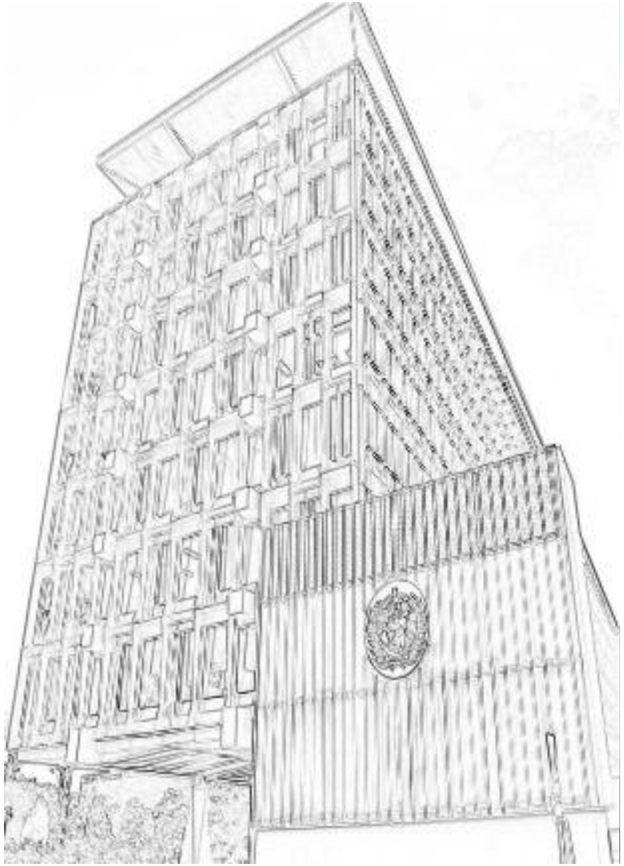
Can Computer Vision profit from In-Network Processing?

- **Computer Vision pipelines also require low latencies & high bandwidths**
  - Establishing themselves as critical system components
    - Production: Collaborative robotics / assistance, quality assurance, safety, …
    - Traffic: Crossroads monitoring (vehicle / pedestrian detection), …
  - Such systems are becoming increasingly interconnected

- **Data structure not INP compatible at first glimpse**
  - Camera images several OOM larger than packets
    - Data may be split across packets → Need to accumulate / **share state**
    - → Longer dwelling time of data on programmable network devices
  - Complex calculations
    - Matrix-vector / matrix-matrix multiplications
    - Loops / iterative refinements

René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

Convolution filters for edge detection



- **Prominent processing step in CV: Edge detection via response of *convolution operation***
  - Given: Picture P (grayscale, $p \times q$ pixels)
  - Define: Filter F (grayscale or binary, $m \times n$ pixels)
    - *Prewitt operator:*

-1	0	1
-1	0	1
-1	0	1

F_{Δ_H}

-1	-1	-1
0	0	0
1	1	1

F_{Δ_V}

    - *Scharr (symmetric Sobel) operator:*

-47	0	47
-162	0	162
-47	0	47

F_{Δ_H}

-47	-162	-47
0	0	0
47	162	47

F_{Δ_V}

  - Filter response: $R_{\Delta_{Dir}}(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{n} P(x-i+a, y-j+a)\, F_{\Delta_{Dir}}(i,j)$
    - Maximum response $|M| = \sqrt{R_{\Delta_H}(x,y)^2 + R_{\Delta_V}(x,y)^2}$
    - Can be approximated: $|M| \propto \left|R_{\Delta_H}(x,y)\right| + \left|R_{\Delta_V}(x,y)\right|$

René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

Applicability of edge detection filters on programmable network devices

Common pattern in most current scenarios:
Few individual operations on **many small** items

Independent of other pictures

Only local information needed
(surroundings of a pixel)

Only addition/subtraction
and multiplication of integers

Minimal global state (if any,
maximum $|M|$ for normalization)

- **Prominent processing step in CV: Edge detection via response of *convolution operation***
  - Given: Picture P (grayscale, $p \times q$ pixels)
  - Define: Filter F (grayscale or binary, $m \times n$ pixels)
    - *Prewitt operator:*

-1	0	1
-1	0	1
-1	0	1

F_{Δ_H}

-1	-1	-1
0	0	0
1	1	1

F_{Δ_V}

    - *Scharr (symmetric Sobel) operator:*

-47	0	47
-162	0	162
-47	0	47

F_{Δ_H}

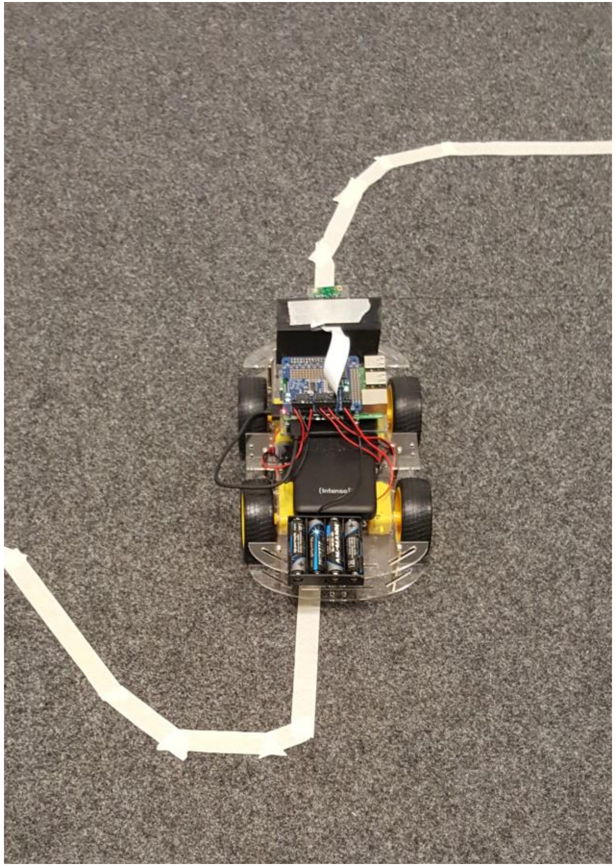
-47	-162	-47
0	0	0
47	162	47

F_{Δ_V}

  - Filter response: $R_{\Delta_{Dir}}(x,y) = \sum_{i=1}^{m} \sum_{j=1}^{n} P(x-i+a, y-j+a)\, F_{\Delta_{Dir}}(i,j)$
    - Maximum response $|M| = \sqrt{R_{\Delta_H}(x,y)^2 + R_{\Delta_V}(x,y)^2}$
    - Can be approximated: $|M| \propto |R_{\Delta_H}(x,y)| + |R_{\Delta_V}(x,y)|$

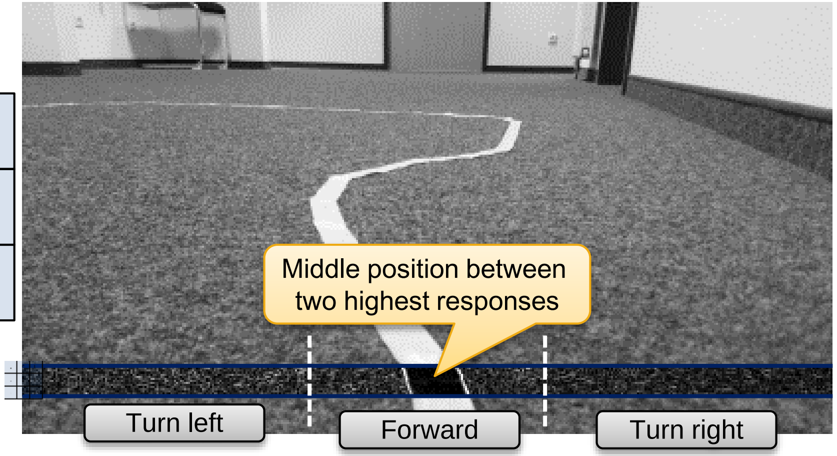
René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

Application scenario: A small line-following car



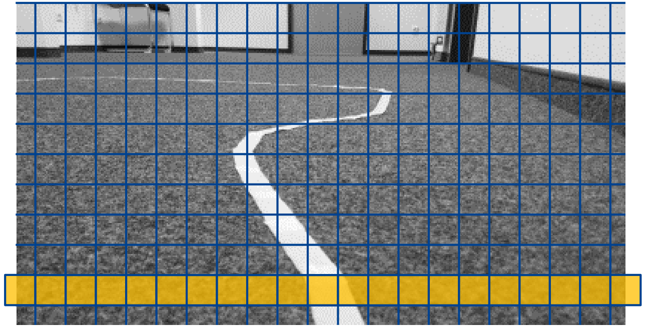
- **Edge detection via INP should work in theory, but does it work in practice?**
  - Given: Small autonomous car with mounted camera
  - Goal: Car should follow sharply contrasting line
  - Idea: P4 program scans for line in selected horizontal region and car turns towards the line if threshold surpassed

-1	0	1
-1	0	1
-1	0	1



René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

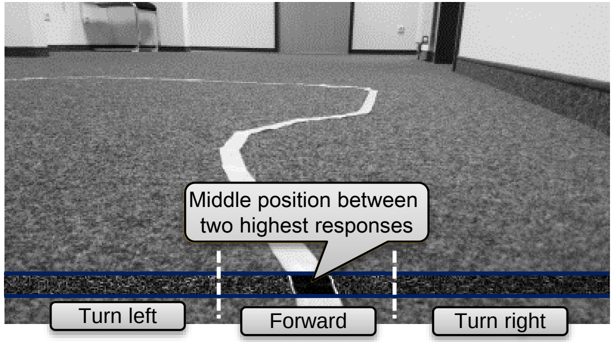
An edge detection filter in P4: Communication pattern

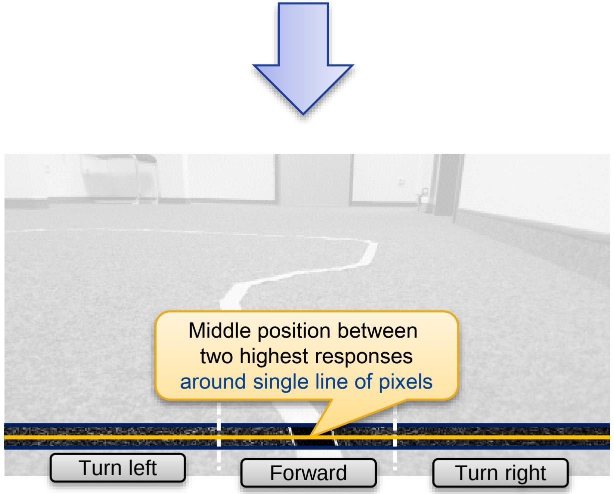


- **Each camera picture should yield a new movement decision with minimal delay**
  - Challenge 1: Full pictures (still) do not fit into packets
  - Solution: Harness locality of edge detection mechanism:
    - Split picture into $n \times n$ chunks and
    - Send in direction of filter application (custom UDP/IP protocol)
  - Chunk size equal to filter size
    - (+) Can directly trigger filter upon arrival of chunk
    - (-) Large overhead (UDP/IP headers for 25 bytes of payload for best filter)
  - Challenge 2: P4 cannot generate packets within the network
  - Solution: Rewrite & reflect packet of last scanning region
    - (+) Minimum achievable latency
    - (-) Loss of packet causes missing control signal, susceptible to reordering

René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

An edge detection filter in P4: Pipeline, step 1 (filtering)



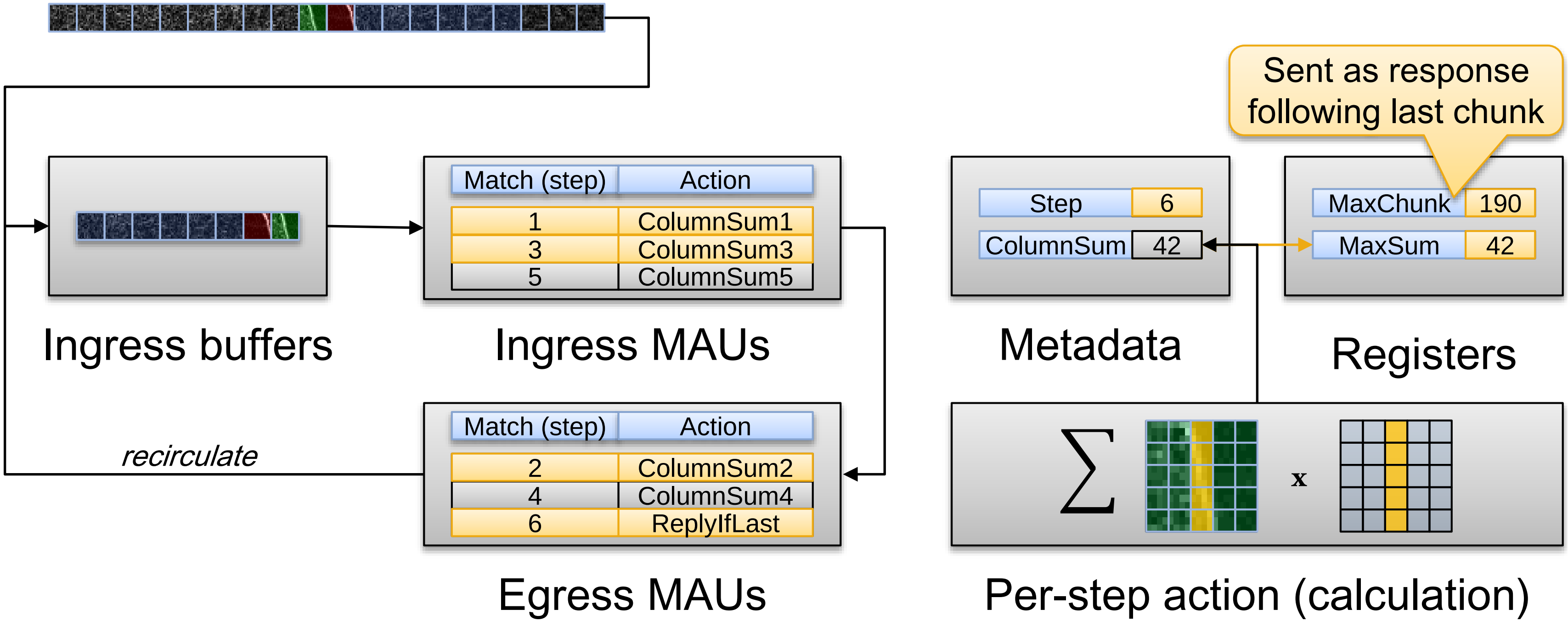


- **Problem structure allows for drastic reduction of computation effort**
  - Challenge 1: Allow other tasks besides edge detection (do not perform unnecessary operations)
  - Solution: Check chunks, drop everything but scanning region
  - Challenge 2: Filter undefined for “edges” of scanning region (no data to perform convolution with)
  - Solution: Assume scanning region is in single series of chunks and only convolve center row of scanning region

René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

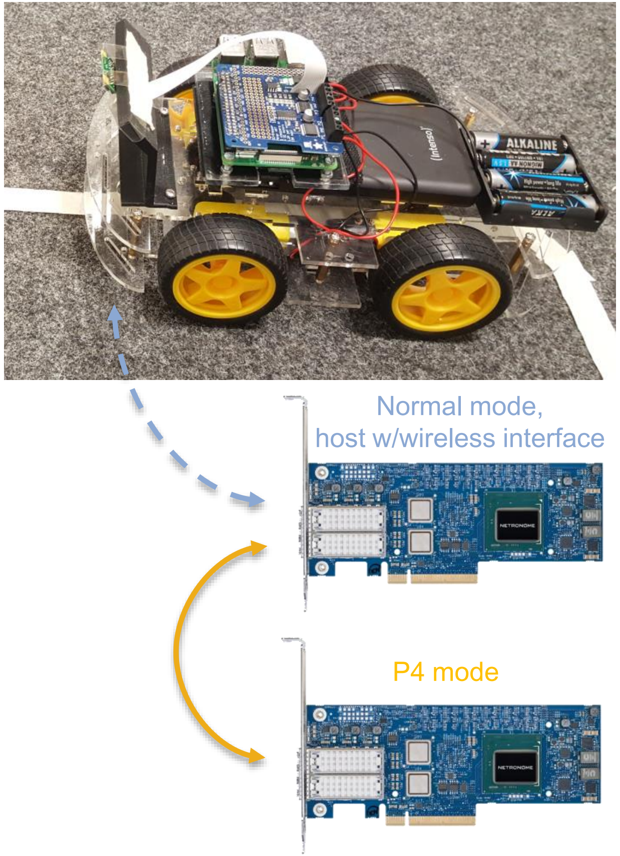
An edge detection filter in P4: Pipeline, step 2 (convolution calculation)

- **Real-world P4 pipelines do not allow single-step convolution calculation**
  - Solution: Column-wise calculation via looped ingress/egress program



René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

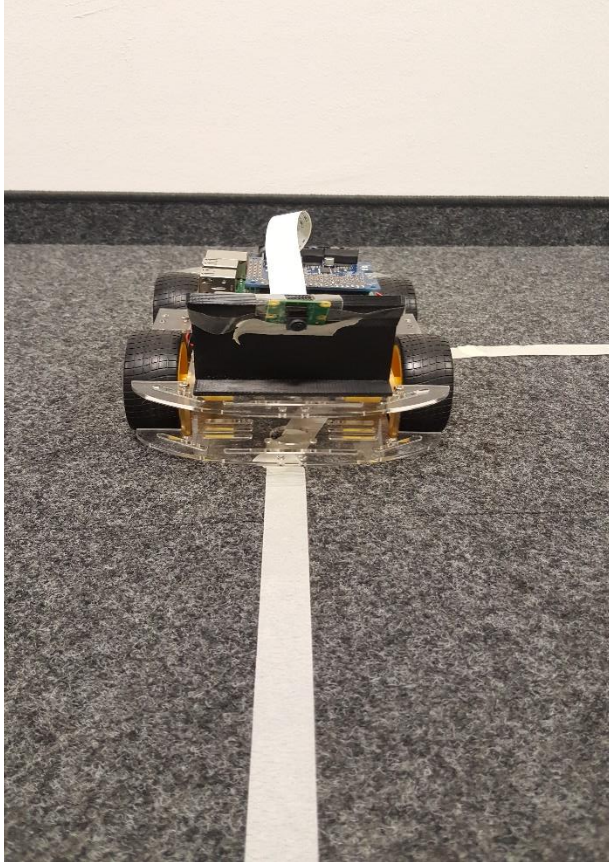
Evaluation



- **Real-world and synthetic benchmarks on Netronome Agilio CX 2x25GbE SmartNICs**
  - 2 connected NICs: 1 as car gateway/generator, 1 for P4
- **Filter- & chunk sizes: Up to 10x10 pixels**
  - O(1): Pipeline lengths (in-/egress)
  - O(n): Table entries, calculations per action
  - Good results at 5x5 already
- **Throughput: 19 fps (5x5); 77 fps (10x10)**
  - Processing of last chunk at 5x5: 150µs (stddev 1.3ms)
  - Processing of last chunk at 10x10: 187µs (stddev 0.6ms)
  - 13.7% drops at 5x5; none for 10x10 → buffering / recirculation

René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

Conclusion: Aiding Computer Vision via In-Network Processing



- **Using INP, we can indeed offer (simple, prototypical) CV-related functionality as a *service in the network***
  - Thanks to an amenable structuration of the data
- **Open questions / future work**
  - Mathematical operations & filter sizes limited
    - Harness table memory via better structure of data?
  - Coexistence with other services in the network
    - Ordering & dropping behavior assumptions (e.g., for recirculation)?
  - Limited data sharing between packets → subsampling only
    - Limited (isolated) buffers for functionality in pipelines?
- **We believe INP can assist in many *stream processing applications***

René Glebke, Johannes Krude, Ike Kunze, Jan Rüth, Felix Senger, Klaus Wehrle
inp@comsys.rwth-aachen.de

Credits

- **Picture on slides 2/4: Wikimedia / Victorgrigas**
  (adapted (cropped) from https://commons.wikimedia.org/w/index.php?title=File:Wikimedia_Foundation_Servers-8055_22.jpg&oldid=218547099)
  CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0/)
- **Picture on slide 5: Wikimedia / Jeff Green/Rethink Robotics**
  (adapted (cropped) from https://commons.wikimedia.org/w/index.php?title=File:Acorn_Sales.jpg&oldid=300471544)
  CC BY (https://creativecommons.org/licenses/by/4.0/)
- **Pictures on slides 6/7: Wikimedia / Thorkild Tylleskar**
  (adapted (cropped / filtered) from
  https://commons.wikimedia.org/w/index.php?title=File:WHO_HQ_main_building,_Geneva,_from_North.JPG&oldid=351921076)
  CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0/)
- **NIC on slide 12: Netronome.com**
- **Car/track pictures: Felix Senger, COMSYS, RWTH Aachen University**
  CC BY-SA (https://creativecommons.org/licenses/by-sa/3.0/)