



# Outline

## Pedestrian detection with HOG/SVM

- HOG feature calculation
- Clustering and merging of overlapping detections

## Contributions of this work

- Trade-off between complexity and performance
- New merging method suitable for hardware implementation

## Results

- Comparison to other works
- Real-time demonstration system

## HOG? Isn't it outdated?!

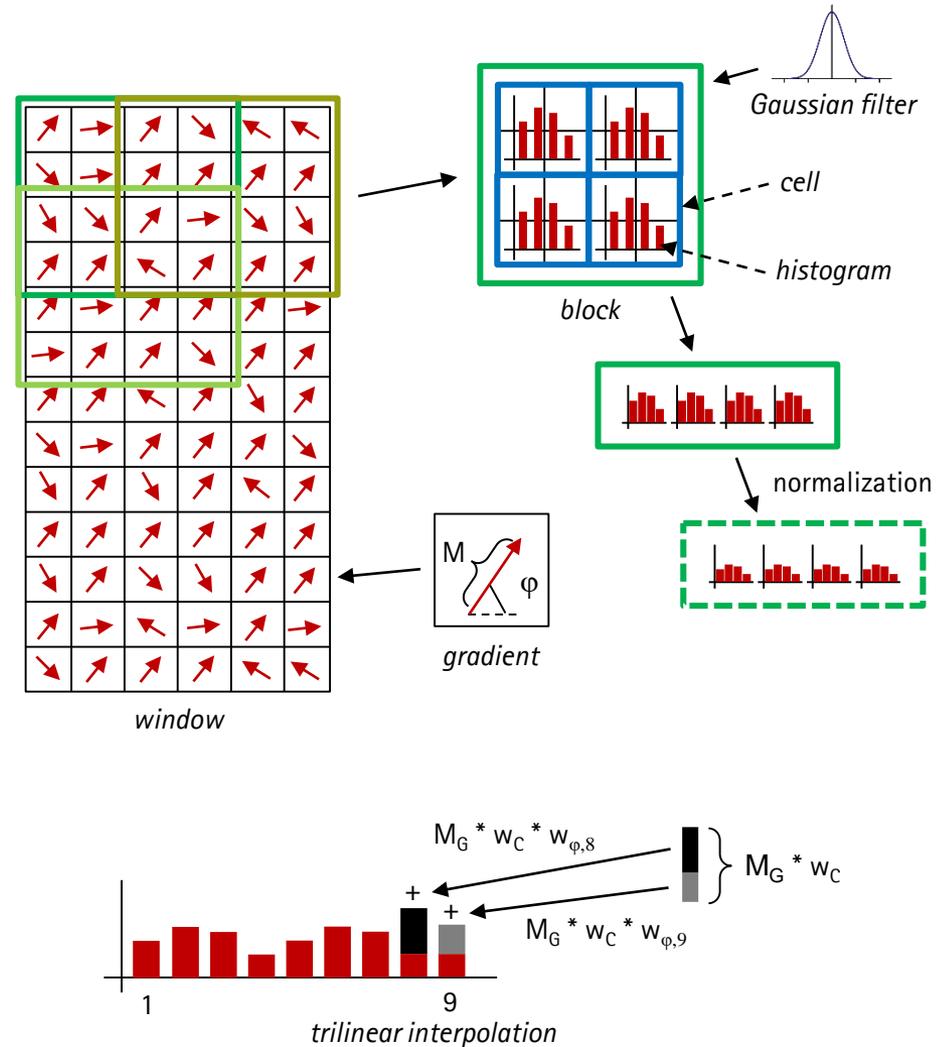
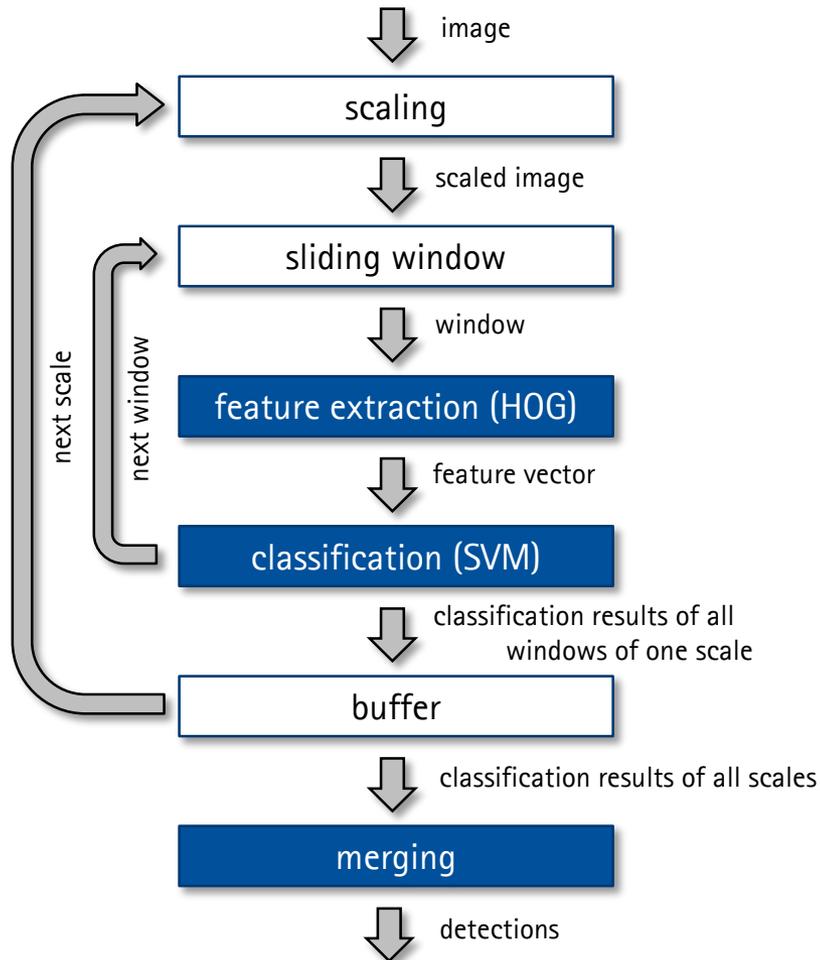
- Pedestrian detection based on feature description with histogram of oriented gradients (HOG) and SVM classification (Dalal / Triggs, CVPR, 2005)
- Detection performance:  $\sim 0.4$  MR @ 0.1 FPPI
- State-of-the-art (CNN-based):  $\sim 0.1$  MR @ 0.1 FPPI <sup>1</sup>
- Humans:  $\sim 0.05$  MR @ 0.1 FPPI <sup>1</sup>
- HOG vs CNN – significant difference in computational complexity
- CNNs require 300x to 13,000x more energy to compute compared to HOG <sup>2</sup>
- For many applications HOG is still a suitable trade-off between complexity and performance!



<sup>1</sup> Zhang et al., "Towards Reaching Human Performance in Pedestrian Detection", TPAMI, 2017

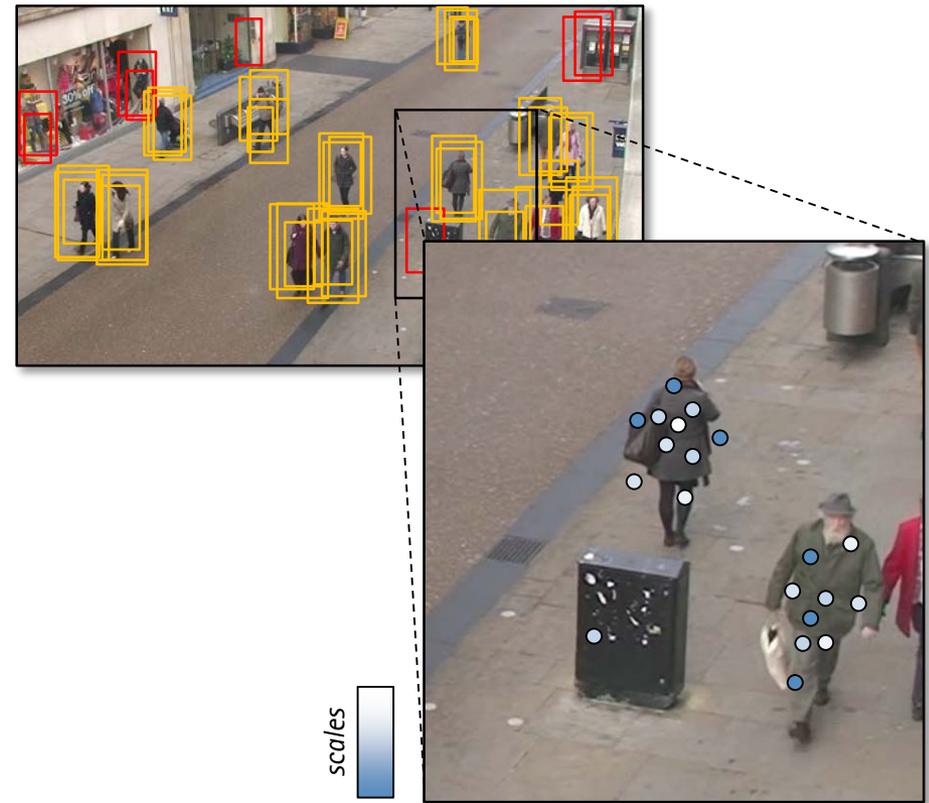
<sup>2</sup> Suleiman et al., "Towards Closing the Energy Gap Between HOG and CNN Features for Embedded Vision", ISCAS, 2017

# HOG-features for pedestrian detection



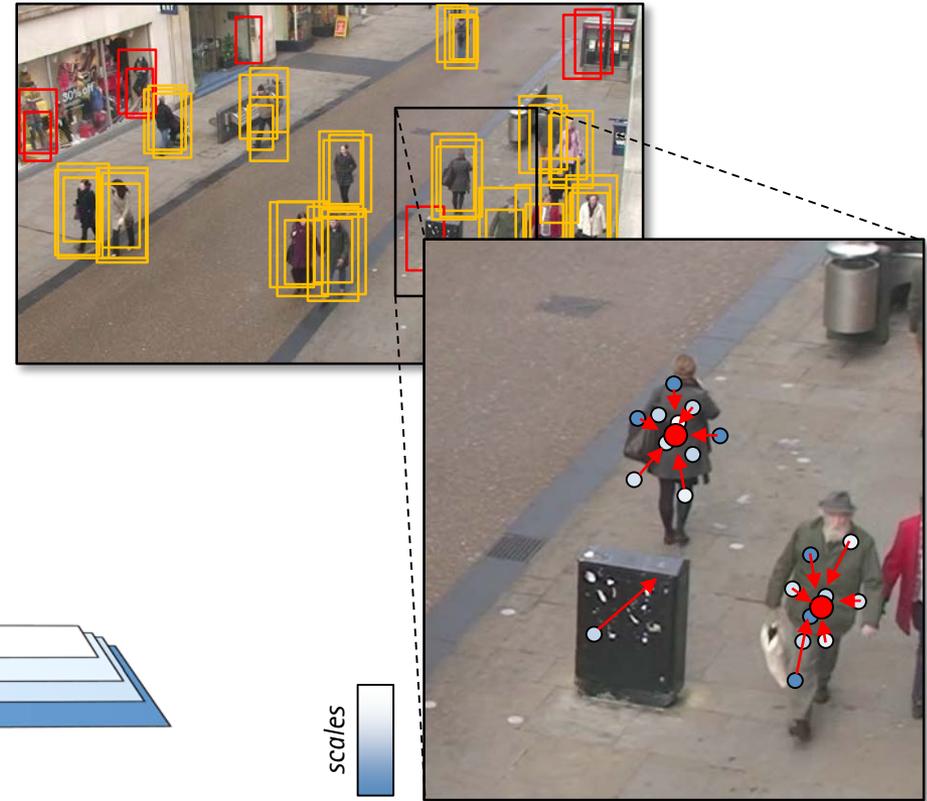
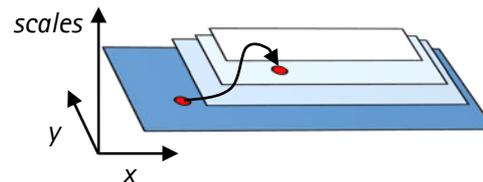
# Merging of mutli-scale detections

- Dalal suggests the iterative Mean-Shift algorithm for clustering and merging of detections



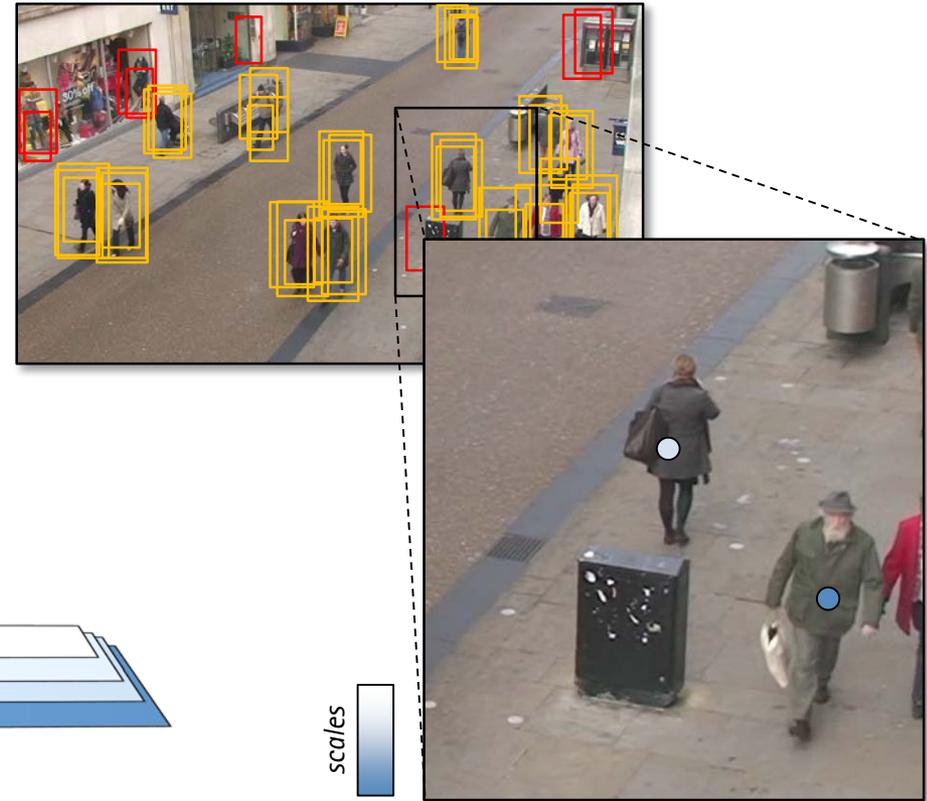
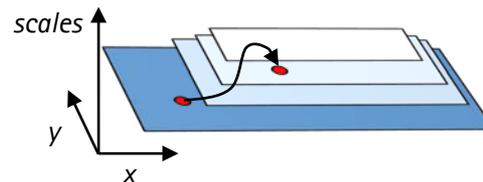
# Merging of mutli-scale detections

- Dalal suggests the iterative Mean-Shift algorithm for clustering and merging of detections
  - Calculation of "mass" centers of the positive detection window centers
  - Moving / shifting of all window centers towards "mass" centers
  - Recalculate and shift until all centers remain still



# Merging of mutli-scale detections

- Dalal suggests the iterative Mean-Shift algorithm for clustering and merging of detections
  - Calculation of "mass" centers of the positive detection window centers
  - Moving / shifting of all window centers towards "mass" centers
  - Recalculate and shift until all centers remain still



- Computationally complex and irregular
- Complexity:  $O(n^2)$

$$y_m = H_h(y_m) \cdot \left[ \sum_{i=1}^n \left( \frac{|H_i|^{-\frac{1}{2}} \cdot t(P_i) \cdot e^{-\frac{1}{2}(y_m - y_i)^T \cdot H_i^{-1} \cdot (y_m - y_i)}}{\sum_{i=1}^n |H_i|^{-\frac{1}{2}} \cdot t(P_i) \cdot e^{-\frac{1}{2}(y_m - y_i)^T \cdot H_i^{-1} \cdot (y_m - y_i)}} \right) \cdot H_i^{-1} \cdot y_i \right]$$

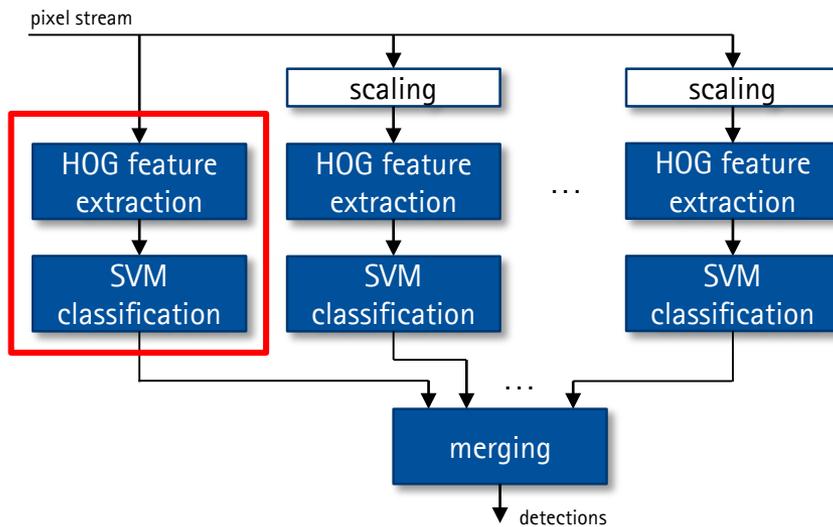
# Contributions

Processing step	Publication							
	2009 Kadato et al.	2011 Negi et al.	2012 Mizuno et al.	2013 Hahnle et al.	2015 Yuan et al.	2015 Rettkow- ski et al.	2015 Ma et al.	2018 this work
Scaling	✗	✗	✗	✓	✗	✗	✗	✓
HOG feat. extr.	✓	✓	✓	✓	✓	✓	✓	✓
SVM classification	✗	●	✓	✓	✓	●	✓	✓
Multi-scale processing	✗	✗	✗	✓	✗	✗	✓	✓
Merging	✗	✗	✗	✗	●	✗	✗	✓

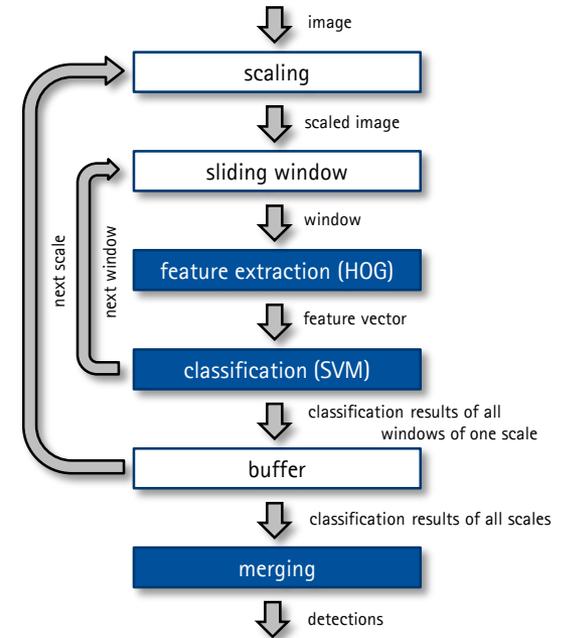
- Many FPGA implementations to be found in literature
  - Merging is rarely considered in hardware implementation
  - A reason could be its complexity and irregularity
- Novel merging method, suitable for hardware implementation
- New approach for a trade-off between complexity and detection performance in HOG feature extraction

# Architecture overview

- 1 pixel per clock, continuously
- Parallel HOG/SVM calculation for different scales

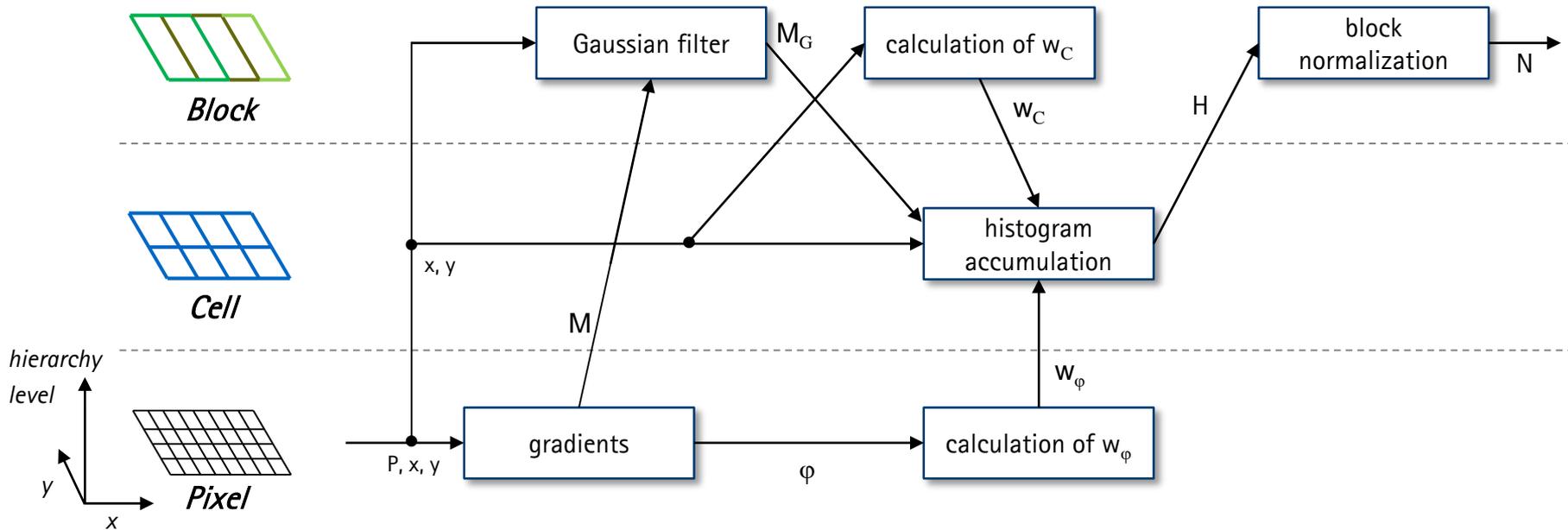


*Hardware structure*

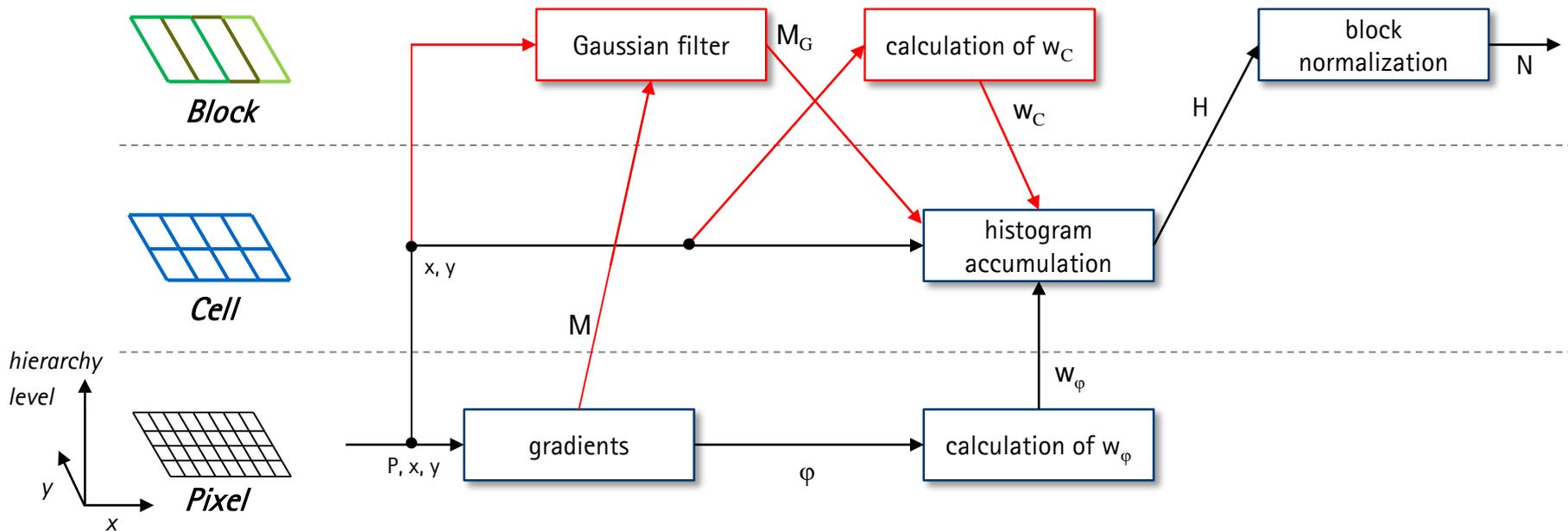


*Algorithm*

# Trade-off between complexity and detection performance in HOG feature calculation

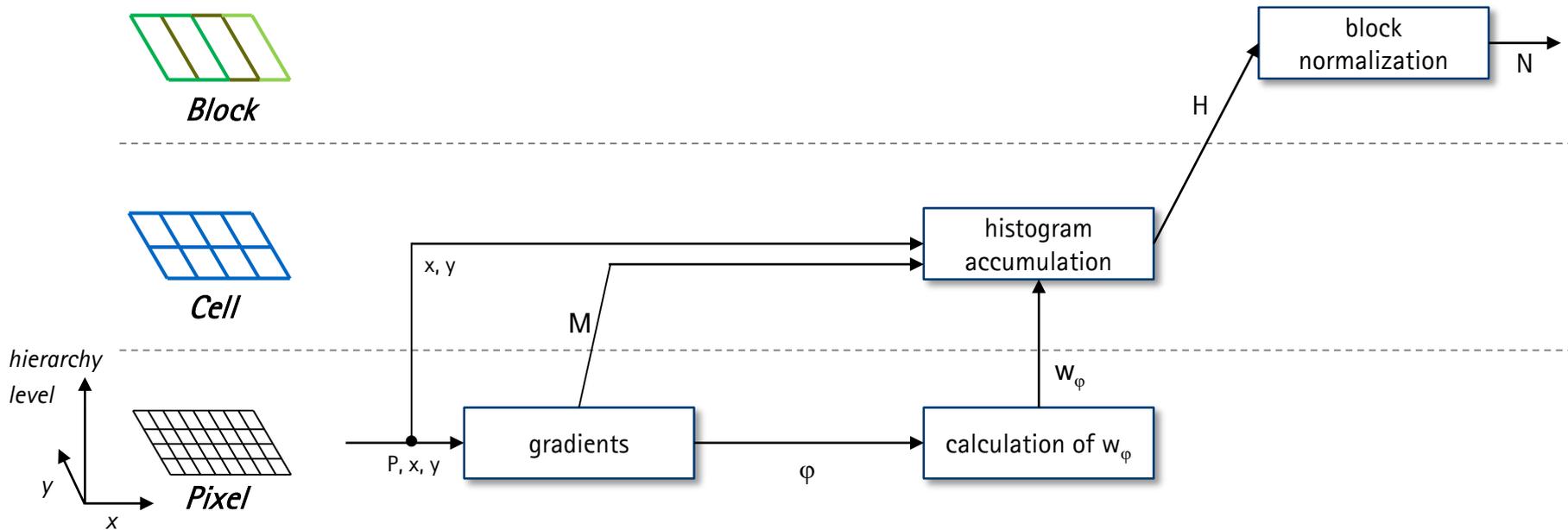


# Trade-off between complexity and detection performance in HOG feature calculation



Some complexity of the algorithm comes with jumping back and forth between hierarchy levels!

# Trade-off between complexity and detection performance in HOG feature calculation



Skipping the Gaussian filter and the inter-cell interpolation:

- Far less complex bottom-up approach
- Minor loss in detection performance of about  $\sim 3\%$

# Comparison of HOG/SVM-impl. to other work

	Year	FPGA	LUTs	Reg.	DSPs	BRAM [kBit]	Clock rate [MHz]	Resolution	Data rate [FPS]	LUT eff. $\eta$	Detection rate / FPPW	Comment
Kadota et al.	2009	Intel Stratix II	3,794 [LUT6]	6,699	12 [18x18]	?	127.49	640x480	3	1.27	- / -	no Gaussian filter, interpolation + classification
Negi et al.	2011	Xilinx Virtex-5	17,383 [LUT6]	2,181	36 [25x18]	?	44.85	640x480	112	29.42	96% / 0.2 ( $2 \times 10^{-1}$ )	no Gaussian filter + interpolation, AdaBoost
Mizuno et al.	2012	Intel Cyclone IV	34,403 [LUT4]	23,247	68 [18x18]	340	40	800x600	72	25.11	87% / 0.0001 ( $10^{-4}$ )	no Gaussian filter
Hahnle et al.	2013	Xilinx Virtex-5	5,188 [LUT6]	5,178	49 [25x18]	1,188	135 + 270	1920x1080	64	63.16	84% / 0.001 ( $10^{-3}$ )	no interpolation
Yuan et al.	2015	Xilinx Spartan-6	9,955 [LUT6]	13,350	66 [18x18]	208	100	800x600	47	15.11	? / ?	
Rettkowski et al.	2015	Xilinx Zynq	21,297 [LUT6]	5,942	4 [25x18]	-	82.2	1920x1080	40	31.59	90% / 0.04 ( $4 \times 10^{-2}$ )	no Gaussian filter + interp., AdaBoost, ext. DDR
Ma et al.	2015	Xilinx Virxtex-6	98,642 [LUT6]	8,694	63 [25x18]	4,579	150	640x480	250 (est.)	3.45	90% / 0.0001 ( $10^{-4}$ )	
this work	2018	Intel Cyclone IV	4,937 [LUT4]	2,751	47 [9x9]	849	70 + 140	1920x1080	33	101.28	87% / 0.0001 ( $10^{-4}$ )	no Gaussian filter, linear bin-interpolation
this work	2018	Intel Stratix V	3,529 [LUT6]	2,657	26 [27x27]	815	142 + 284	1920x1080	68	94.46	87% / 0.0001 ( $10^{-4}$ )	no Gaussian filter, linear bin-interpolation

$$\text{LUT Efficiency: } \eta = \frac{\text{pixels/s}}{(\alpha \cdot \#LUTs) \cdot f_{clk}}$$

$\alpha = 1.0$  for 4 – input LUTs

$\alpha = 1.5$  for 6 – input LUTs

# Comparison of HOG/SVM-impl. to other work

	Year	FPGA	LUTs	Reg.	DSPs	BRAM [kBit]	Clock rate [MHz]	Resolution	Data rate [FPS]	LUT eff. $\eta$	Detection rate / FPPW	Comment
Kadota et al.	2009	Intel Stratix II	3,794 [LUT6]	6,699	12 [18x18]	?	127.49	640x480	3	1.27	- / -	no Gaussian filter, interpolation + classification
Negi et al.	2011	Xilinx Virtex-5	17,383 [LUT6]	2,181	36 [25x18]	?	44.85	640x480	112	29.42	96% / 0.2 ( $2 \times 10^{-1}$ )	no Gaussian filter + interpolation, AdaBoost
Mizuno et al.	2012	Intel Cyclone IV	34,403 [LUT4]	23,247	68 [18x18]	340	40	800x600	72	25.11	87% / 0.0001 ( $10^{-4}$ )	no Gaussian filter
Hahnle et al.	2013	Xilinx Virtex-5	5,188 [LUT6]	5,178	49 [25x18]	1,188	135 + 270	1920x1080	64	63.16	84% / 0.001 ( $10^{-3}$ )	no interpolation
Yuan et al.	2015	Xilinx Spartan-6	9,955 [LUT6]	13,350	66 [18x18]	208	100	800x600	47	15.11	? / ?	
Rettkowski et al.	2015	Xilinx Zynq	21,297 [LUT6]	5,942	4 [25x18]	-	82.2	1920x1080	40	31.59	90% / 0.04 ( $4 \times 10^{-2}$ )	no Gaussian filter + interp., AdaBoost, ext. DDR
Ma et al.	2015	Xilinx Virxtex-6	98,642 [LUT6]	8,694	63 [25x18]	4,579	150	640x480	250 (est.)	3.45	90% / 0.0001 ( $10^{-4}$ )	
this work	2018	Intel Cyclone IV	4,937 [LUT4]	2,751	47 [9x9]	849	70 + 140	1920x1080	33	101.28	87% / 0.0001 ( $10^{-4}$ )	no Gaussian filter, linear bin-interpolation
this work	2018	Intel Stratix V	3,529 [LUT6]	2,657	26 [27x27]	815	142 + 284	1920x1080	68	94.46	87% / 0.0001 ( $10^{-4}$ )	no Gaussian filter, linear bin-interpolation

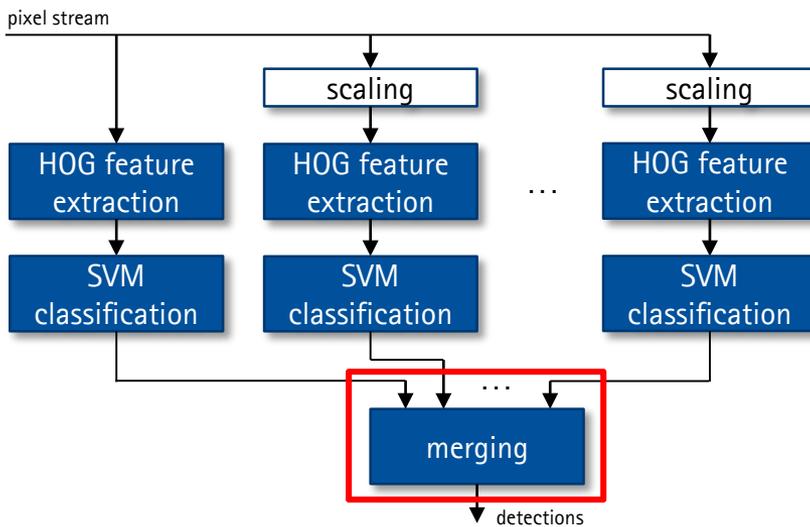
$$\text{LUT Efficiency: } \eta = \frac{\text{pixels/s}}{(\alpha \cdot \#LUTs) \cdot f_{clk}}$$

$\alpha = 1.0$  for 4 – input LUTs

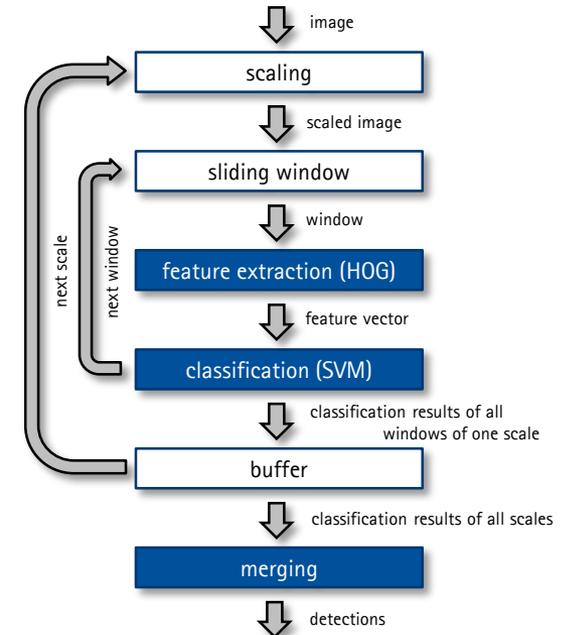
$\alpha = 1.5$  for 6 – input LUTs

# Architecture overview

- 1 pixel per clock, continuously
- Parallel HOG/SVM calculation for different scales



*Hardware structure*

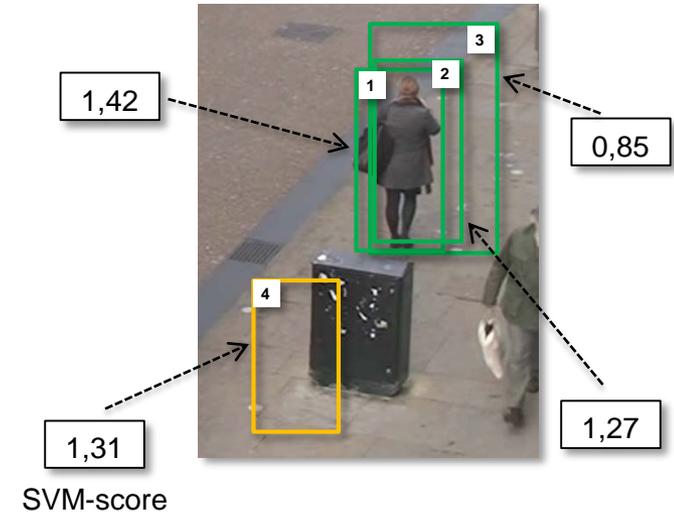


*Algorithm*

# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

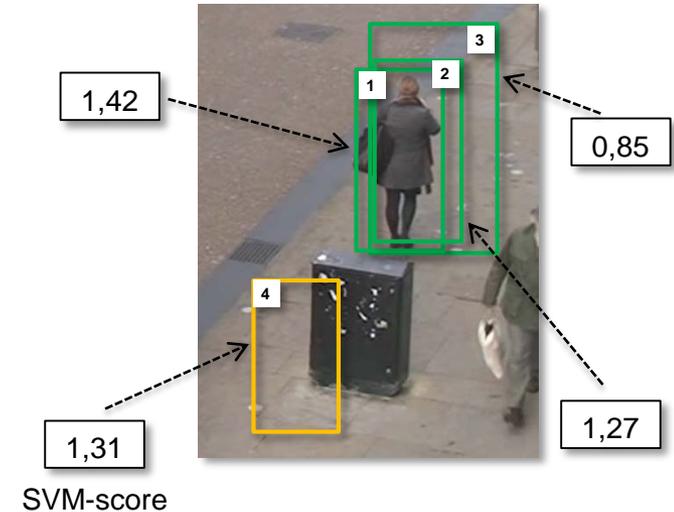
1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

Position	Index	Score
1	1	1,42
2	2	1,27
3	3	0,85
4	4	1,31

# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

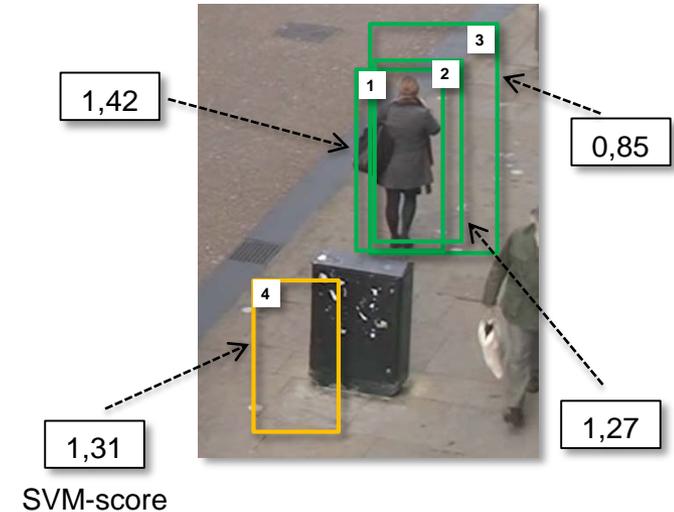
1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

Position	Index	Score ↓
1	1	1,42
2	4	1,31
3	2	1,27
4	3	0,85

# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

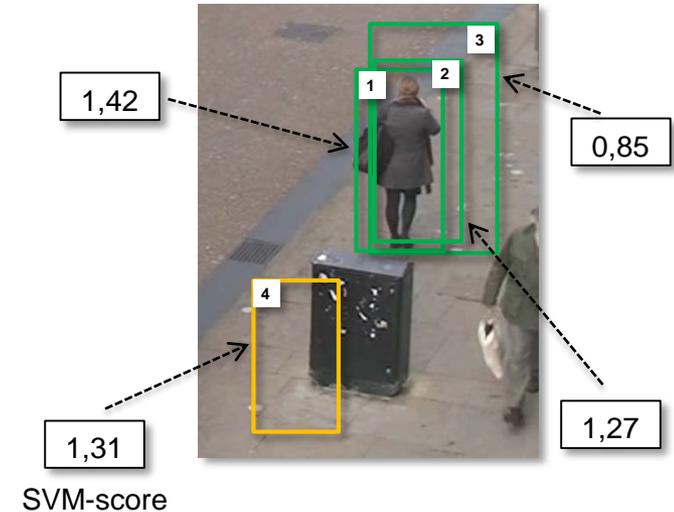
1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

Position	Index	Score
1	1	1,42
2	4	1,31
3	2	1,27
4	3	0,85

# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

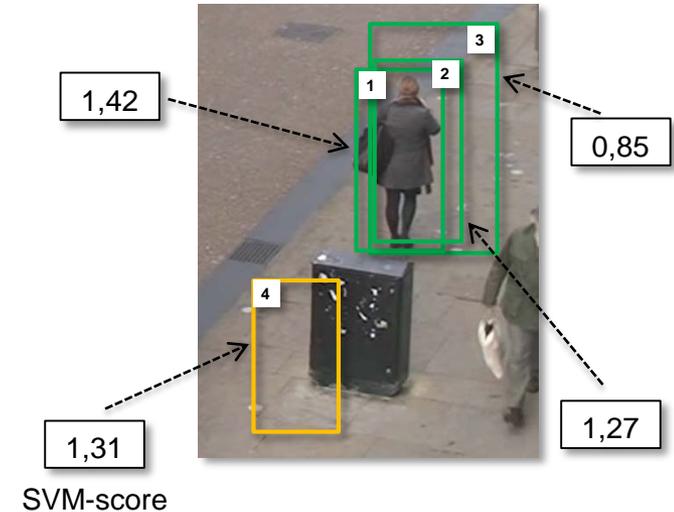
Position	Index	Score
1	1	1,42
2	4	1,31
3	2	1,27
4	3	0,85



# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

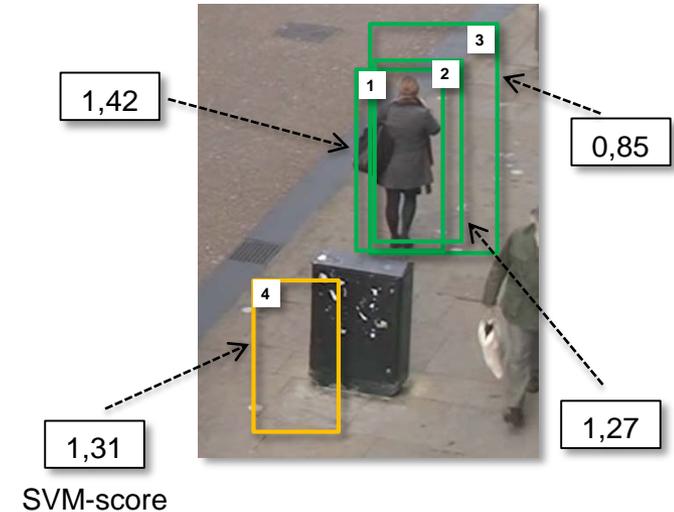
Position	Index	Score
1	1	1,42
2	4	1,31
3	2	1,27
4	3	0,85



# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

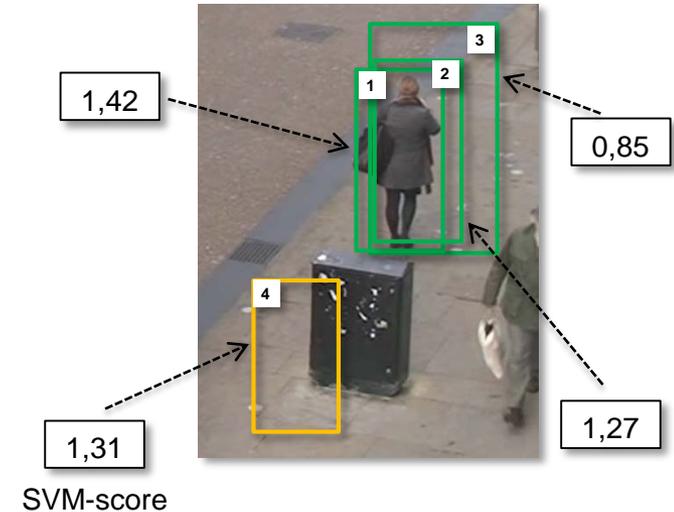
Position	Index	Score
1	1	1,42
2	4	1,31
<del>3</del>	<del>2</del>	<del>1,27</del>
4	3	0,85



# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

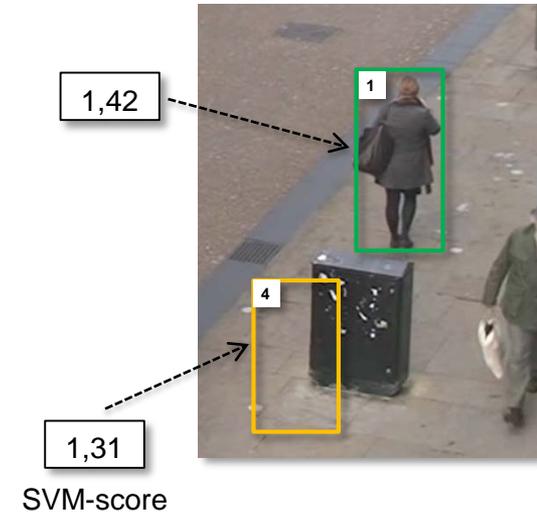
Position	Index	Score
1	1	1,42
2	4	1,31
<del>3</del>	<del>2</del>	<del>1,27</del>
<del>4</del>	<del>3</del>	<del>0,85</del>



# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

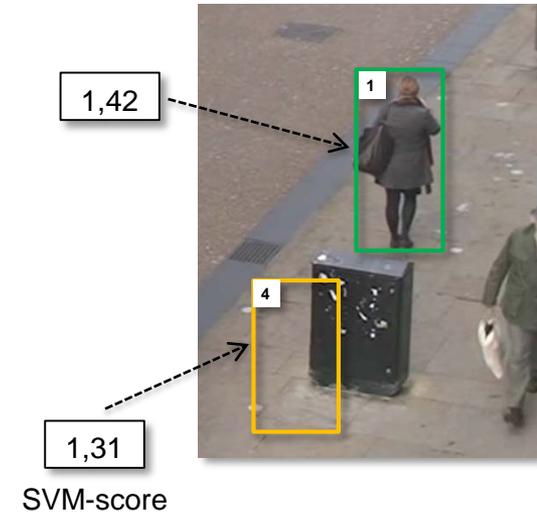
Position	Index	Score
1	1	1,42
2	4	1,31
<del>3</del>	<del>2</del>	<del>1,27</del>
<del>4</del>	<del>3</del>	<del>0,85</del>



# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

Position	Index	Score
1	1	1,42
2	4	1,31
<del>3</del>	<del>2</del>	<del>1,27</del>
<del>4</del>	<del>3</del>	<del>0,85</del>



# New Merging Method

Instead of complex mean-shift: plausibility check

- High SVM-score implies high likelihood of a true positive detection
- Highly overlapping detections are most likely multiple detections of the same object
- Objects always cause multiple overlapping detections, single non-overlapping detections are most likely false



New algorithm suitable for hardware implementation

1. Sort detections by SVM-score
2. For each entry: scan list for 50% overlapping detections
3. Removal of overlapped detections
4. Entry is true positive, if at least 2 overlapped detections have been found

(e.g. Heapsort:  $O(n \log n)$ )

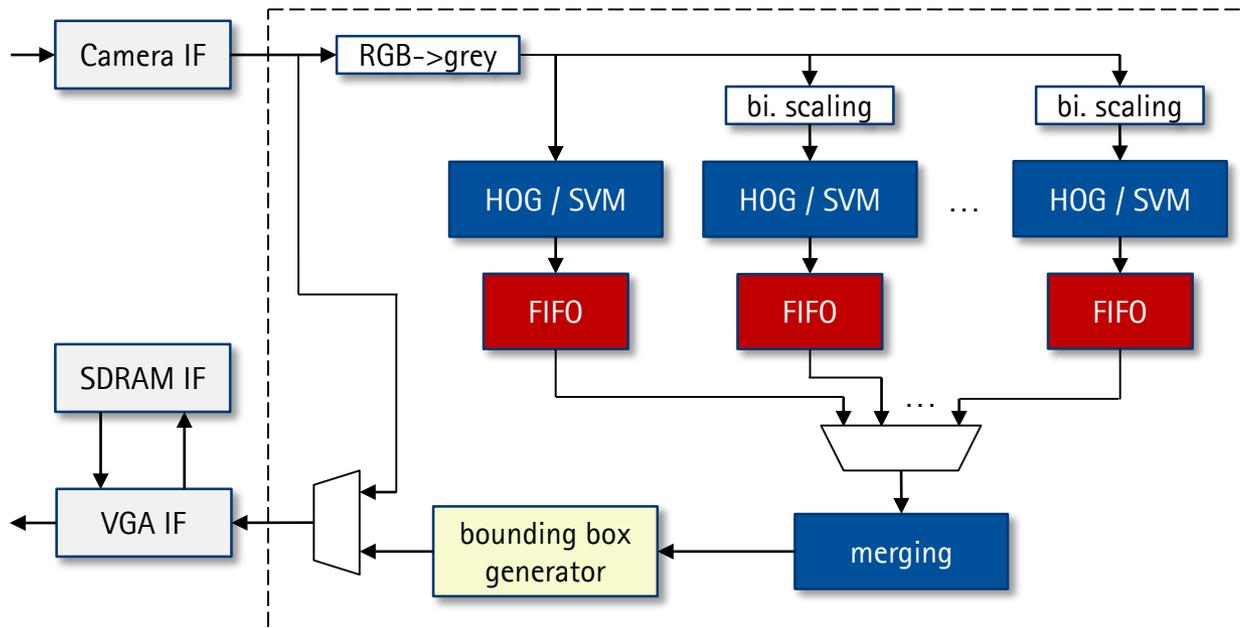
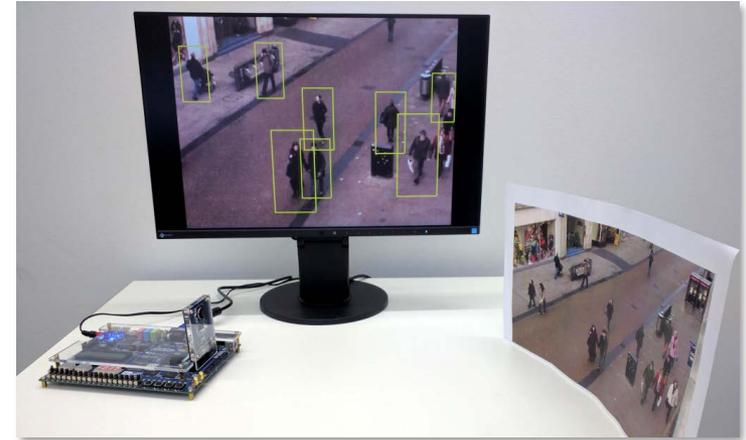
Position	Index	Score
1	1	1,42
<del>2</del>	<del>4</del>	<del>1,31</del>
<del>3</del>	<del>2</del>	<del>1,27</del>
<del>4</del>	<del>3</del>	<del>0,85</del>



*Small arithmetic complexity, run-time complexity dominated by sorting*

# Demonstration System

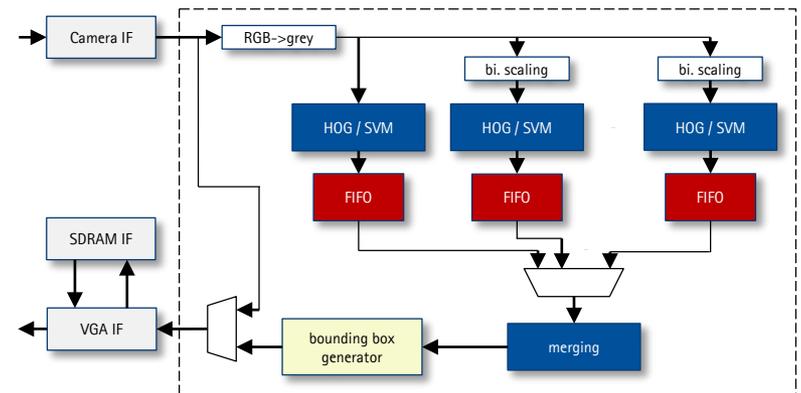
- COTS Altera / Intel DE2-115 dev. board with Cyclone IV FPGA
- 9 scales (factor 1.1), 800x600, 20 FPS
- Resolution and framerate limited by SDRAM



# Synthesis results

<i>Intel Cyclone IV EP4CE115</i> <i>800x600, 50+100 MHz, 9 scales</i>	LUTs	Reg.	DSPs	BRAM [kBit]
<b>Pedestrian Detector</b>	<b>47,175</b>	<b>25,158</b>	<b>491</b>	<b>1,222</b>
- HOG/SVM (9 instances)	41,906	23,138	423	1,137
- Bilinear scaling (8 instances)	1,611	664	64	6
- Merging	1,249	109	4	28
<b>Infrastructure (Framework)</b>	<b>3,343</b>	<b>2,137</b>	<b>0</b>	<b>543</b>
- Camera Interface	889	552	0	432
- VGA Interface	903	488	0	51
- SDRAM Interface	1081	870	0	59
<b>Total</b> <b>(% of available)</b>	<b>50,518</b> <b>(44%)</b>	<b>27,295</b> <b>(24%)</b>	<b>491</b> <b>(92%)</b>	<b>1,765</b> <b>(45%)</b>

- Small hardware costs for merging module
- Number of scales limited by DSPs and BRAM routing



## Summary

- For many embedded pedestrian detection applications HOG/SVM is still a suitable trade-off between complexity and performance
- Merging of overlapping detections was rarely considered in hardware implementations, supposedly because of its complexity and irregularity

This work...

- shows a new approach for the trade-off between algorithm simplifications and performance in HOG/SVM
- presents a highly efficient streaming architecture
- introduces a new method for merging, particularly suitable for hardware implementations
- shows a real-time demonstration system on a small COTS FPGA board