max planck institut
informatik

Ten years of pedestrian detection, what have we learned ?

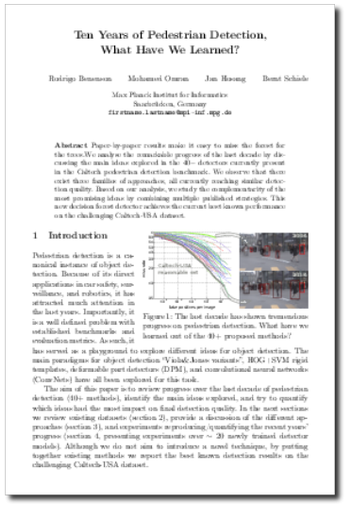
Rodrigo Benenson

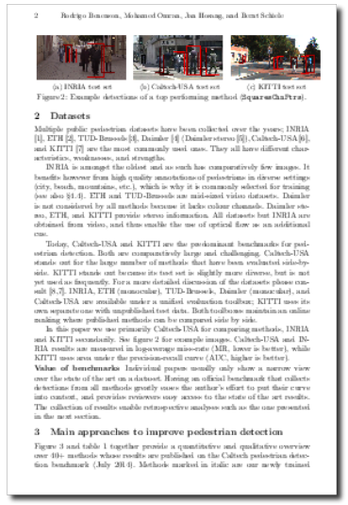
Mohamed Omran

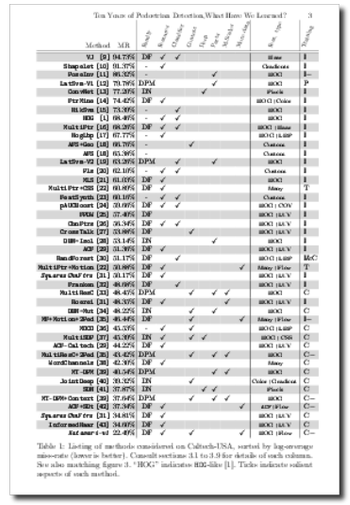
Jan Hosang

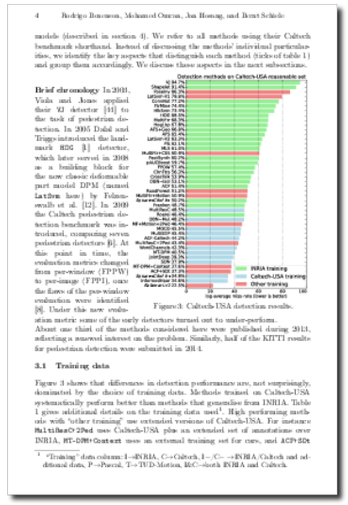
Bernt Schiele

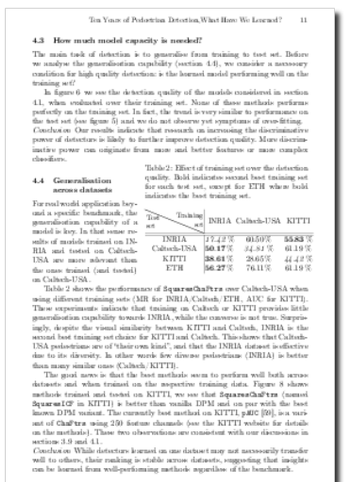
This presentation: paper overview + my 3 surprises

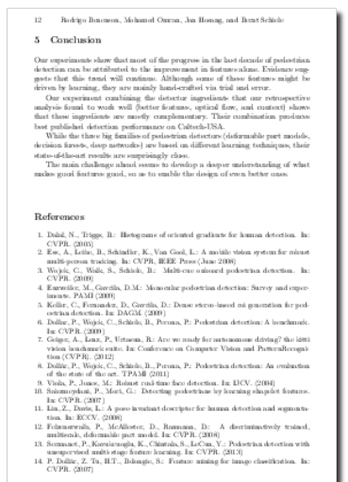


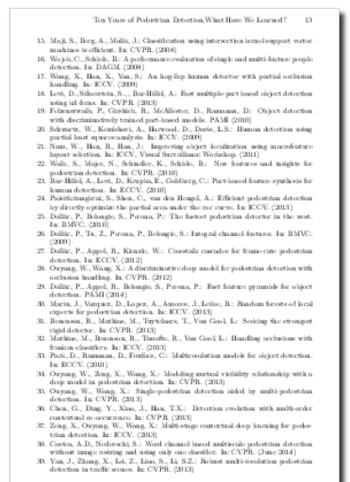


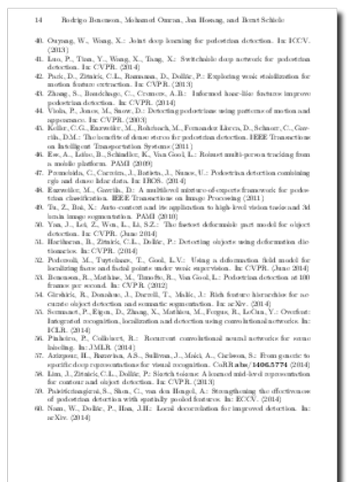




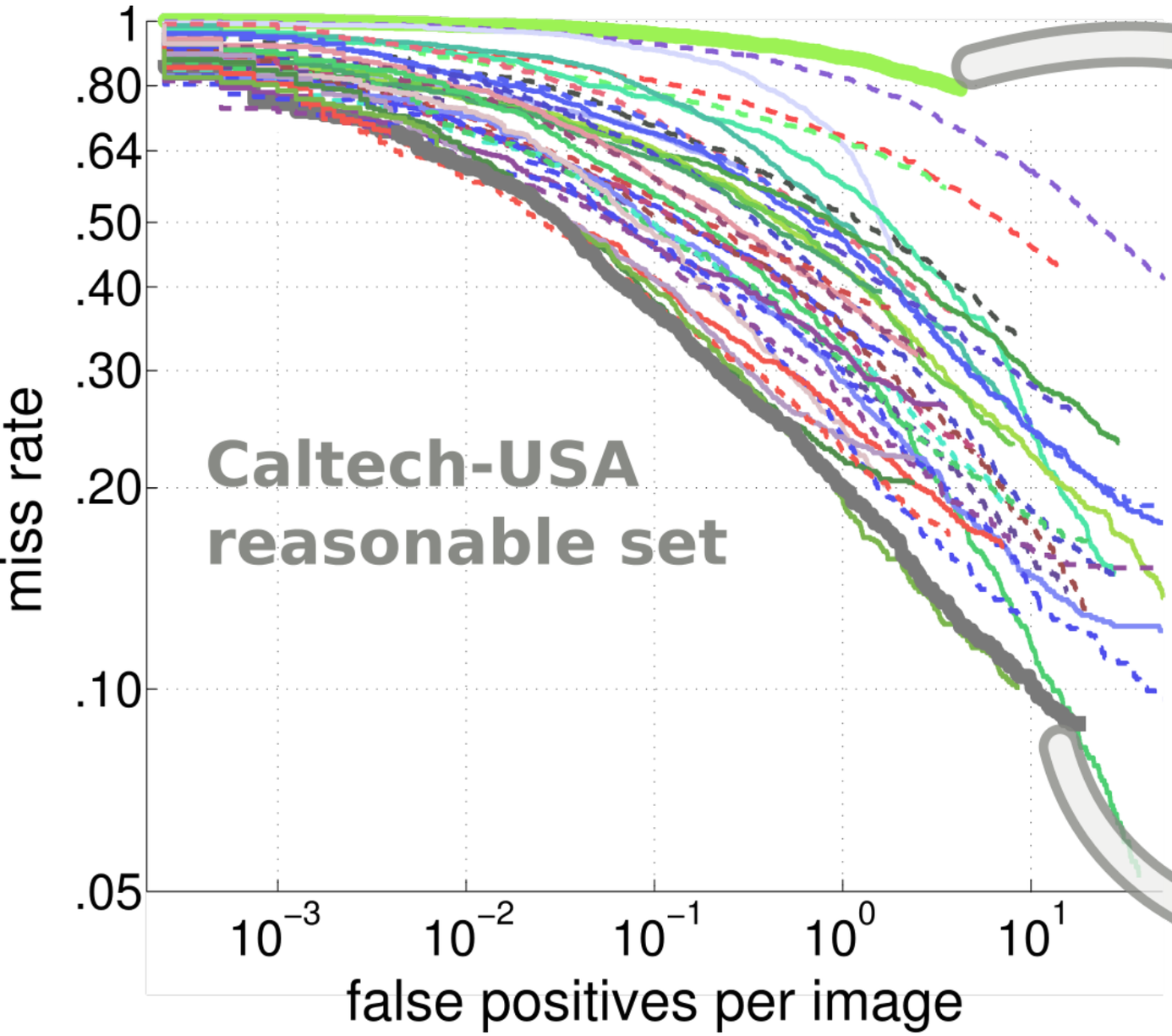


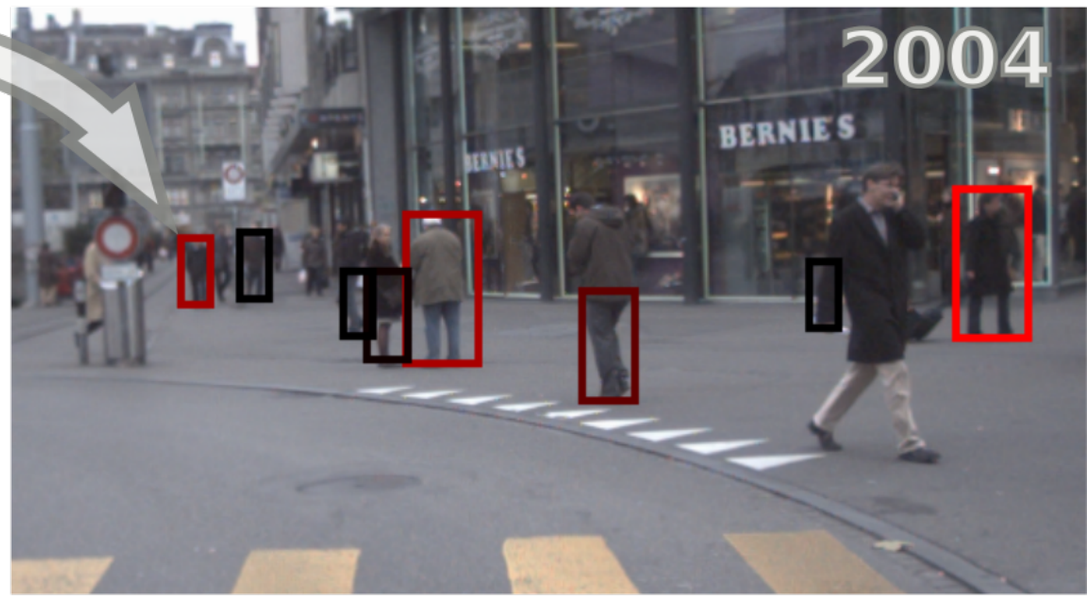






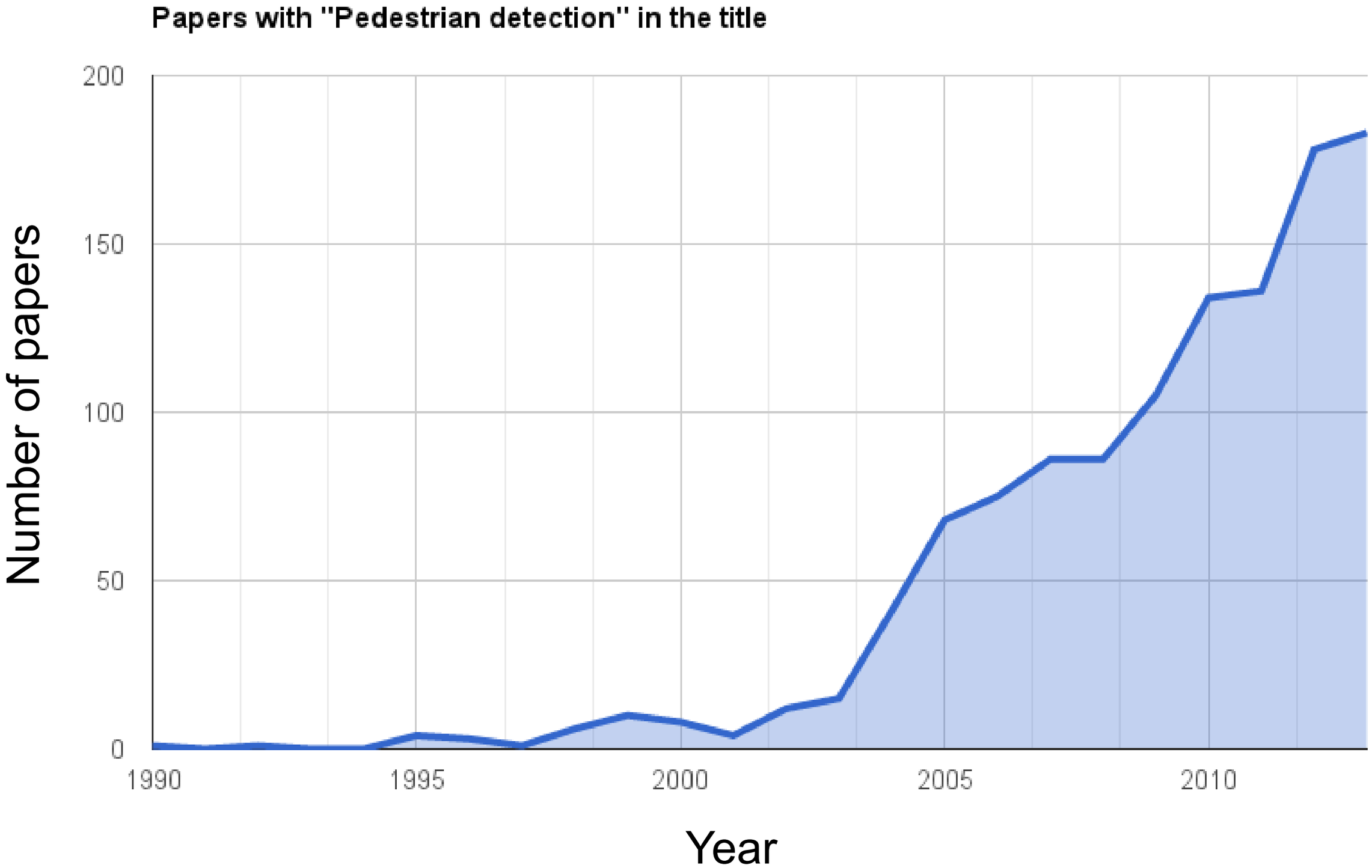
Great progress in pedestrian detection during last decade



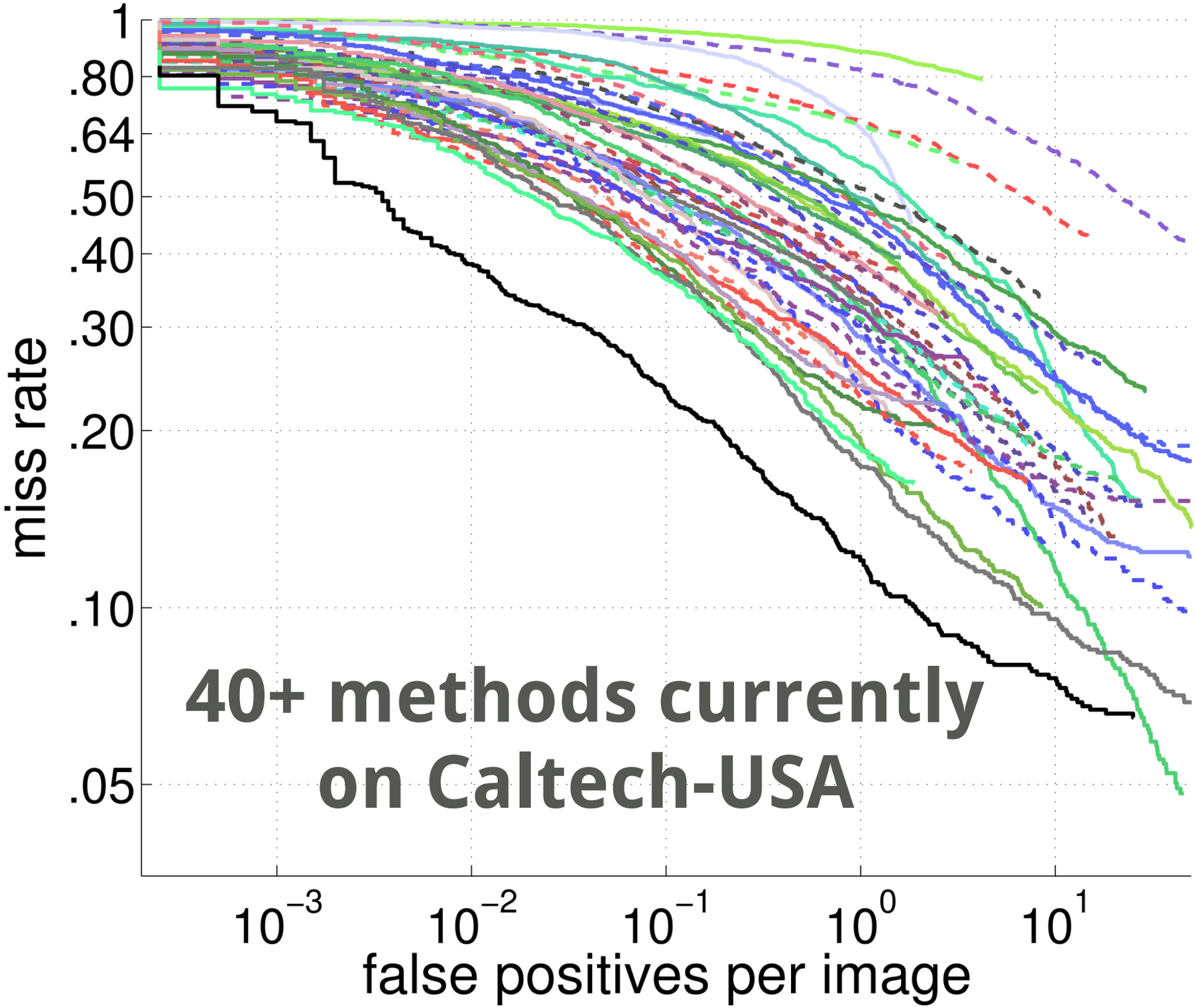


Caltech-USA is currently the most active dataset.

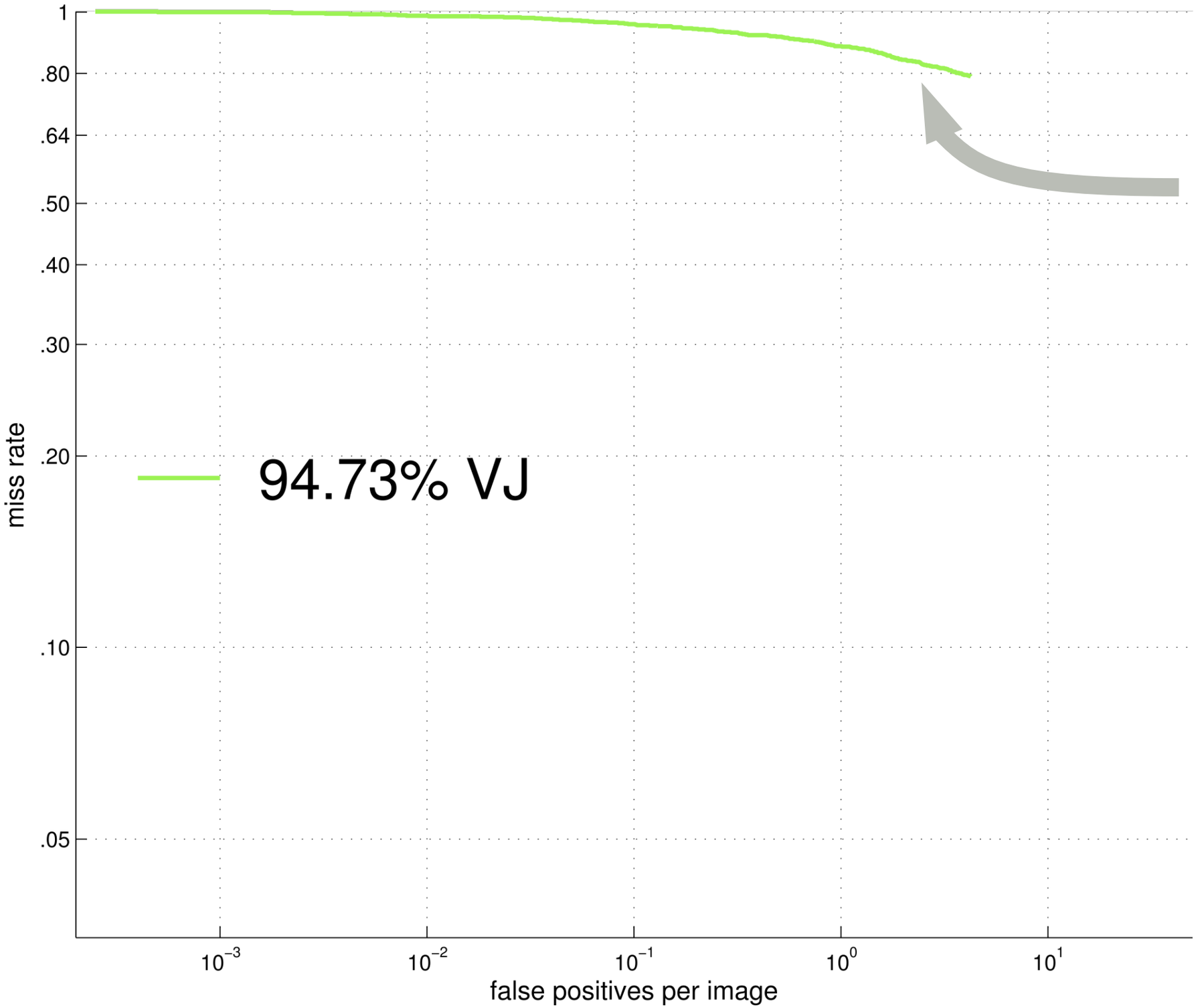
Pedestrian detection is still very active

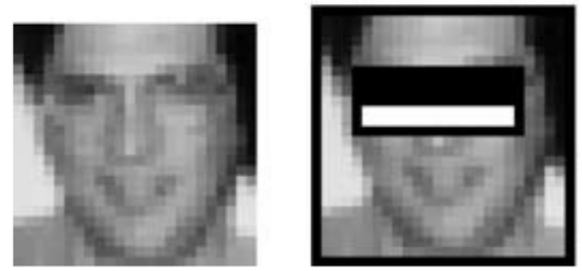


Quick chronology: 5 landmarks



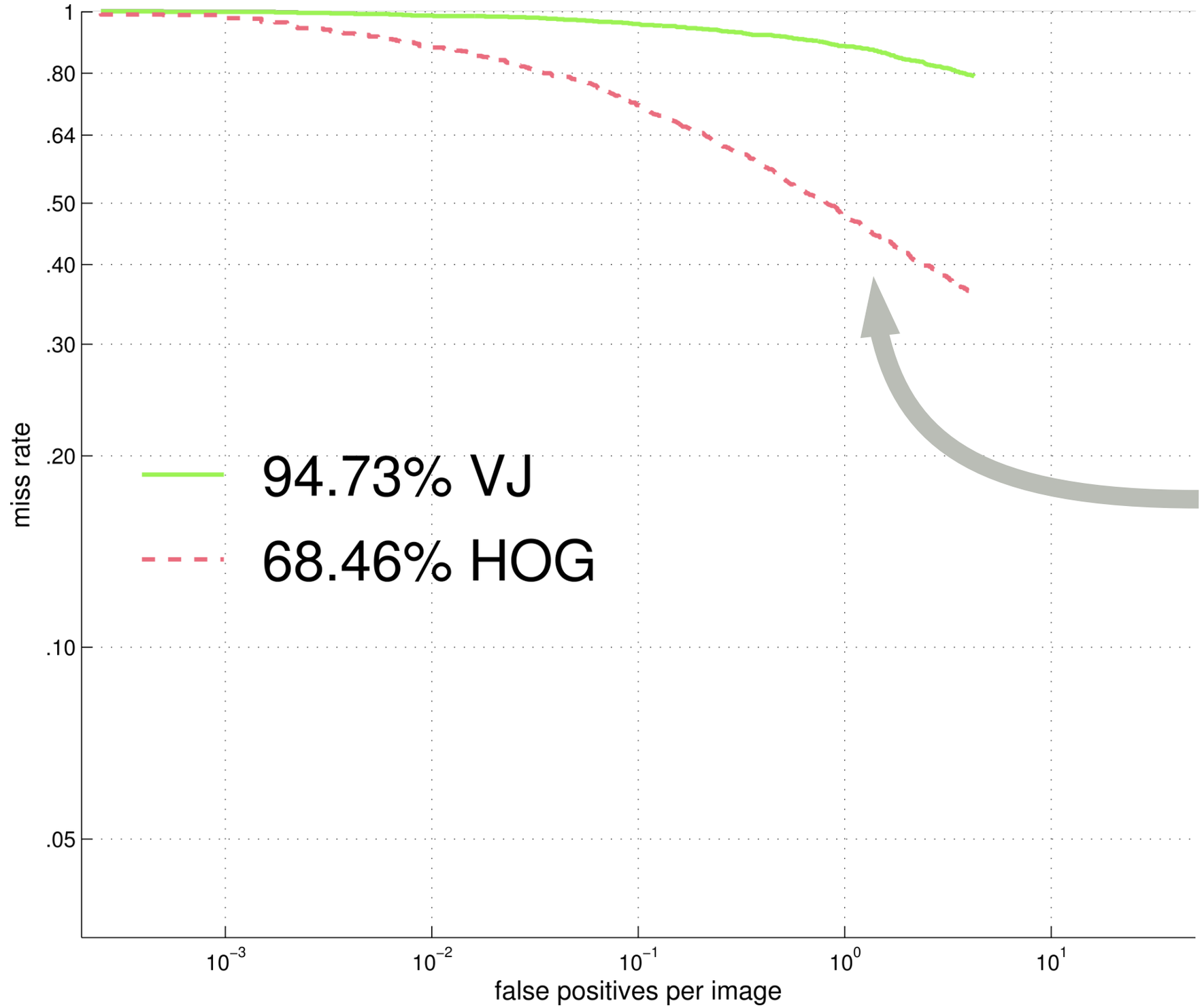
Quick chronology: 5 landmarks



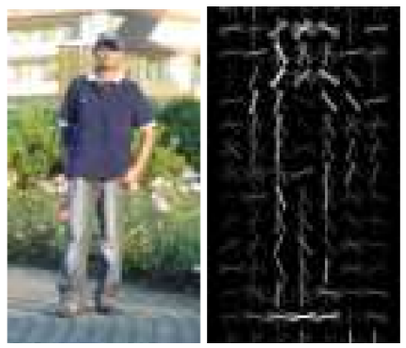


[Viola & Jones 2004]

Quick chronology: 5 landmarks

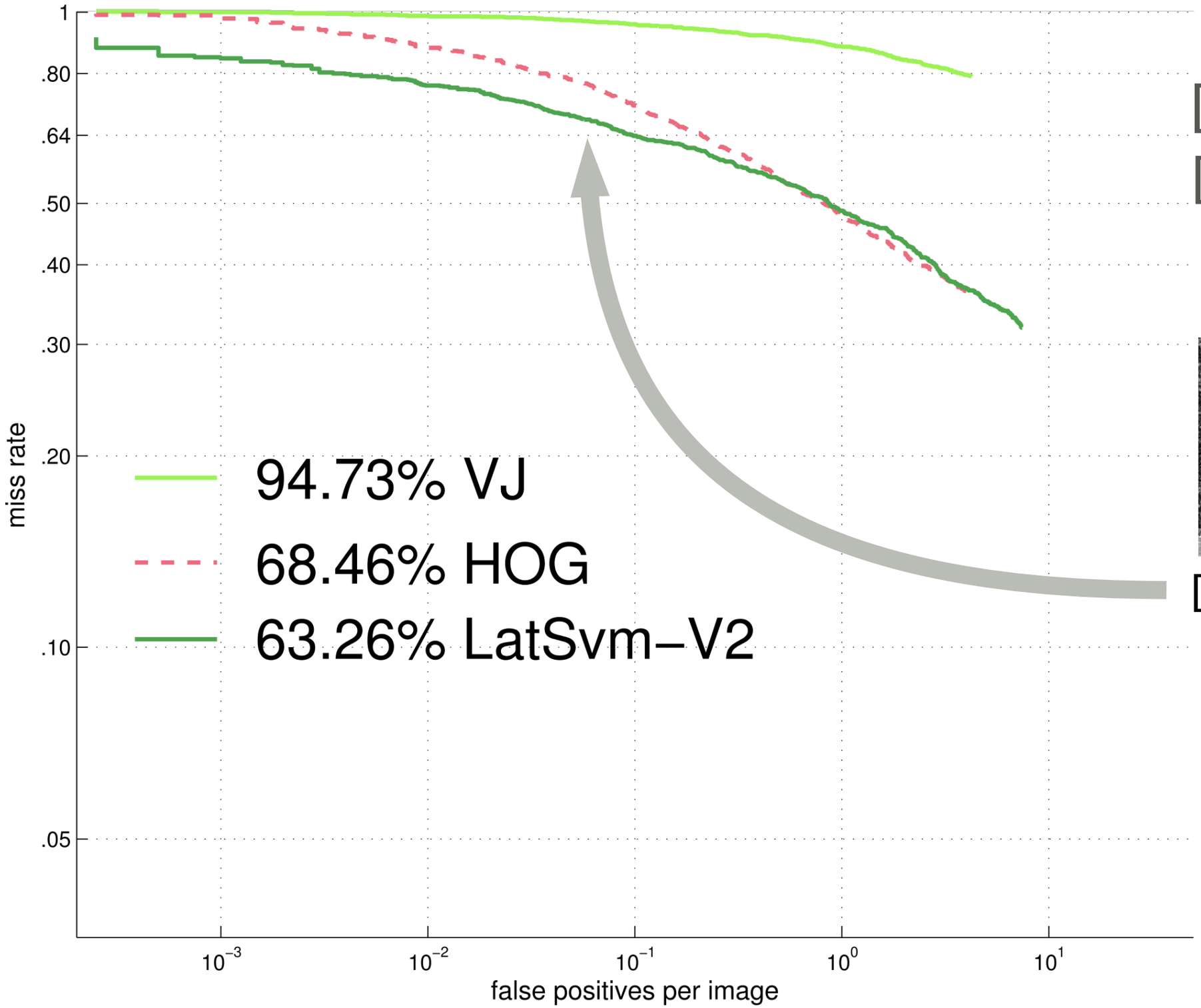


[Viola & Jones 2004]

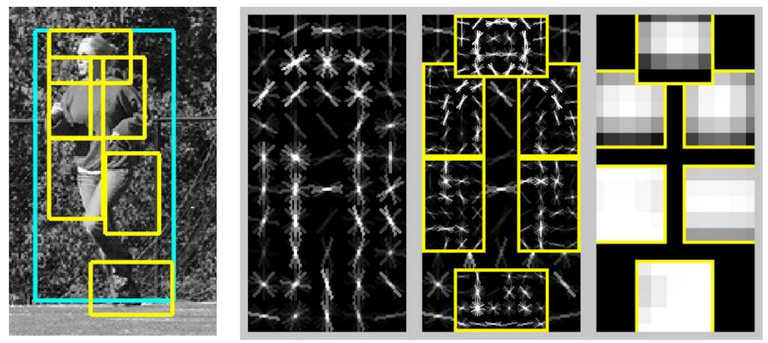


[Dalal & Triggs 2005]

Quick chronology: 5 landmarks



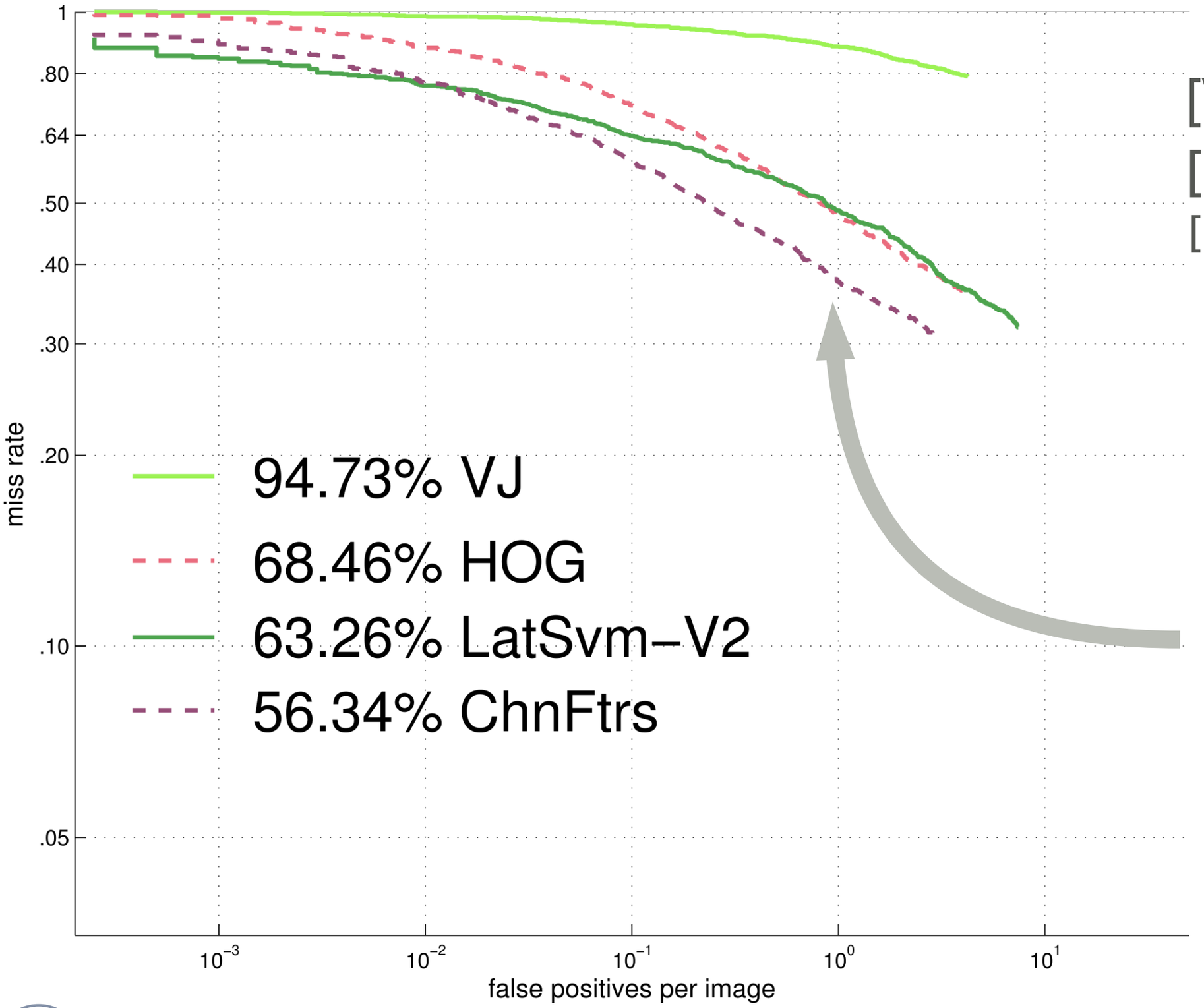
[Viola & Jones 2004]
[Dalal & Triggs 2005]



[Felzenszwalb et al. 2008]

DPM family

Quick chronology: 5 landmarks

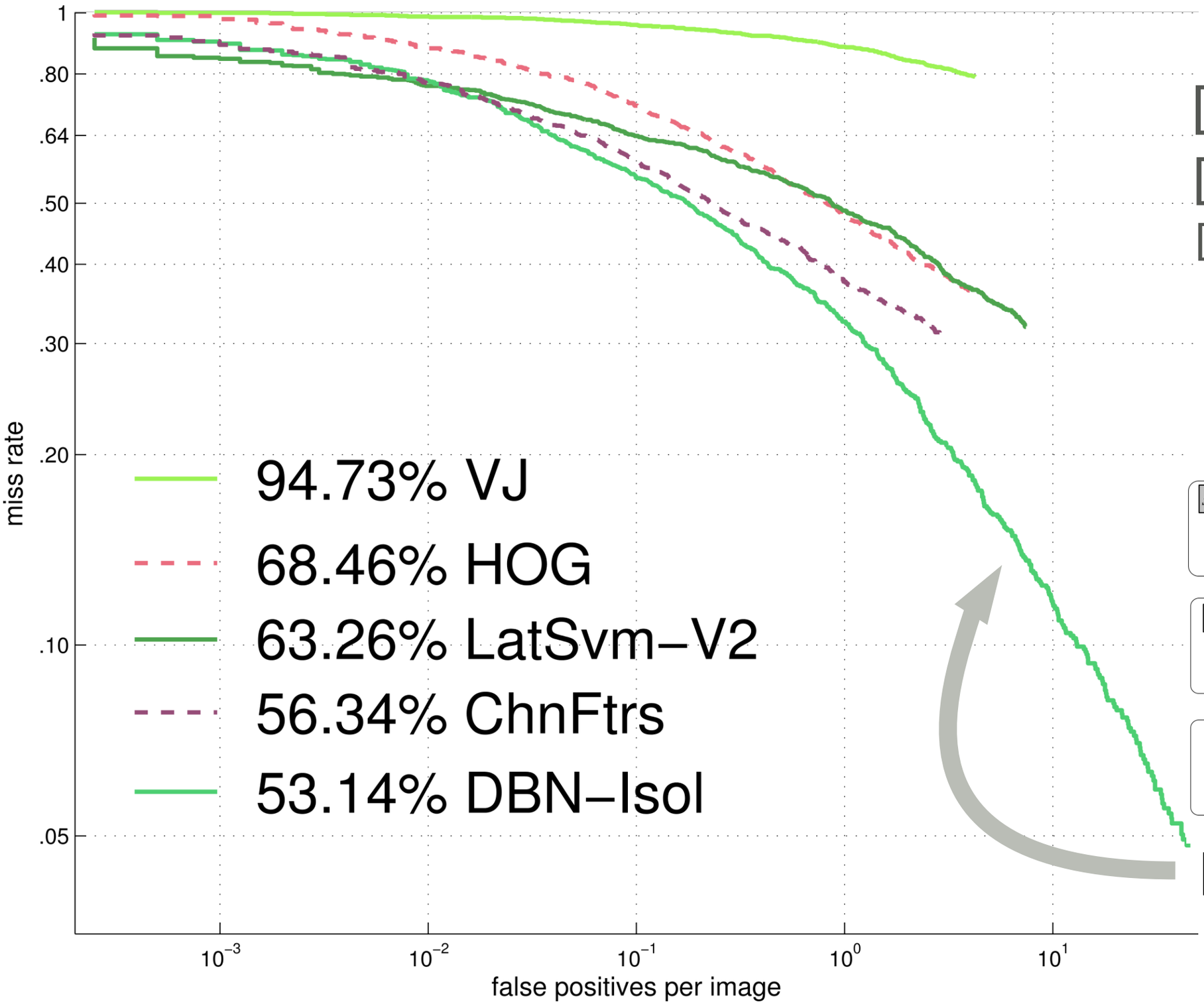


[Viola & Jones 2004]
[Dalal & Triggs 2005]
[Felzenszwalb et al. 2008]

[Dollar et al. 2009]

DF family

Quick chronology: 5 landmarks

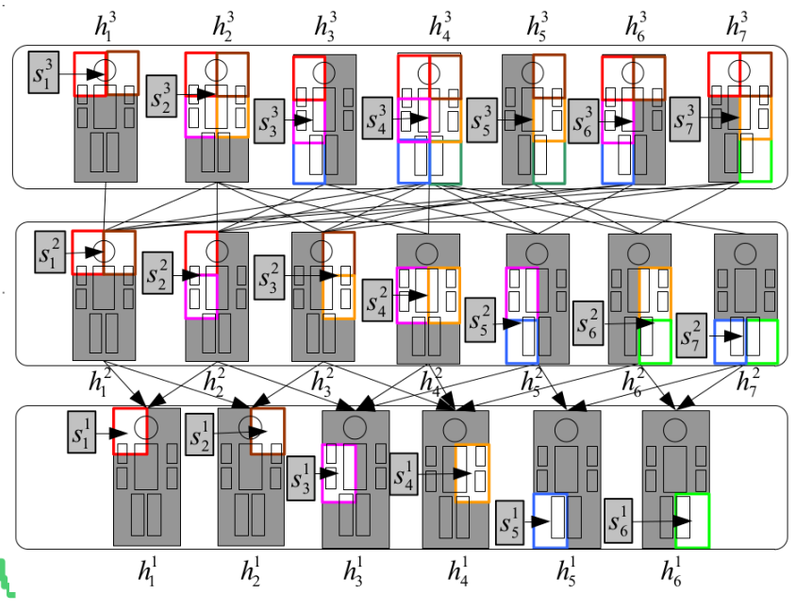


[Viola & Jones 2004]

[Dalal & Triggs 2005]

[Felzenszwalb et al. 2008]

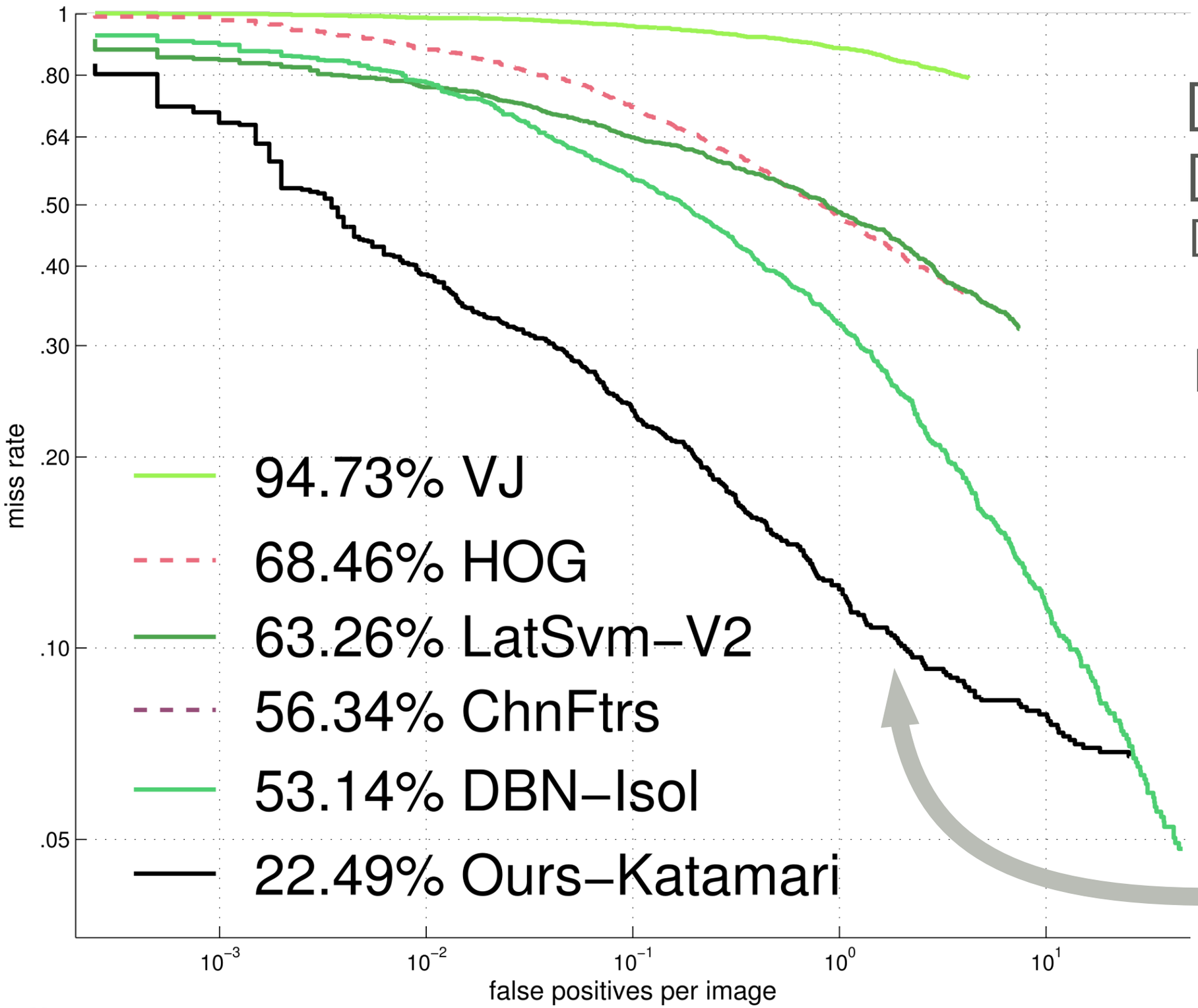
[Dollar et al. 2009]



[Ouyang & Wang 2012]

DN family

Quick chronology: 5 landmarks



[Viola & Jones 2004]

[Dalal & Triggs 2005]

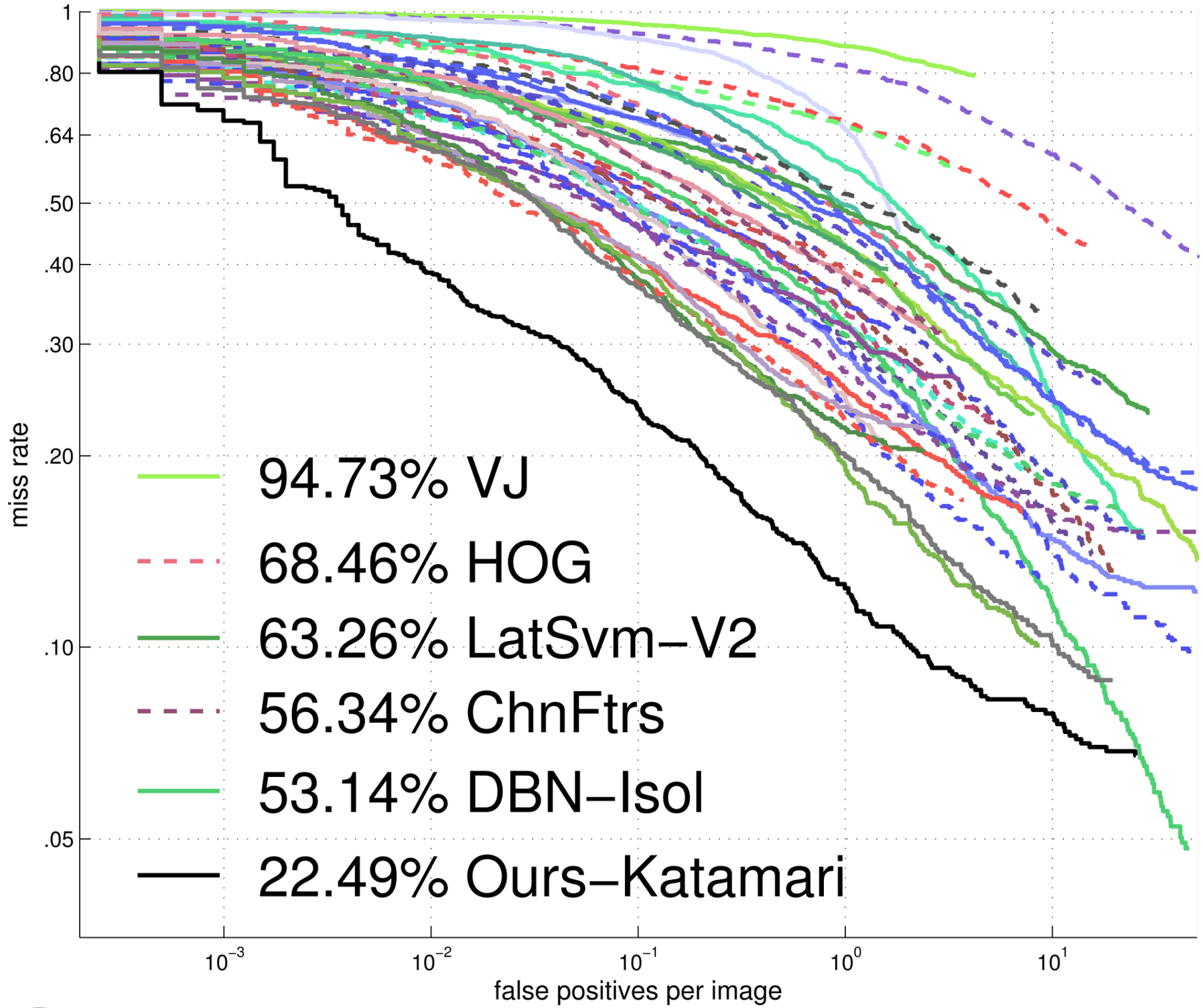
[Felzenszwalb et al. 2008]

[Dollar et al. 2009]

[Ouyang & Wang 2012]

[This paper 2014]

Quick chronology: 5 landmarks



[Viola & Jones 2004]
[Dalal & Triggs 2005]
[Felzenszwalb et al. 2008]
[Dollar et al. 2009]
[Ouyang & Wang 2012]
[This paper 2014]

Method	MR	Family	Features	Classifier	Context	Deep	Parts	M-Scales	More data	Feat. type	Training
VJ [9]	94.73%	DF	✓	✓						Haar	I
Shapelet [10]	91.37%	-	✓							Gradients	I
PoseInv [11]	86.32%	-					✓			HOG	I+
LatSvm-V1 [12]	79.78%	DPM					✓			HOG	P
ConvNet [13]	77.20%	DN				✓				Pixels	I
FtrMine [14]	74.42%	DF	✓							HOG+Color	I
HikSvm [15]	73.39%	-		✓						HOG	I
HOG [1]	68.46%	-	✓	✓						HOG	I
MultiFtr [16]	68.26%	DF	✓	✓						HOG+Haar	I
HogLbp [17]	67.77%	-	✓							HOG+LBP	I
AFS+Geo [18]	66.76%	-			✓					Custom	I
AFS [18]	65.38%	-								Custom	I
LatSvm-V2 [19]	63.26%	DPM		✓			✓			HOG	I
Pls [20]	62.10%	-	✓	✓						Custom	I
MLS [21]	61.03%	DF	✓							HOG	I
MultiFtr+CSS [22]	60.89%	DF	✓							Many	T
FeatSynth [23]	60.16%	-	✓	✓						Custom	I
pAUCBoost [24]	59.66%	DF	✓	✓						HOG+COV	I
FPDW [25]	57.40%	DF								HOG+LUV	I
ChnFtrs [26]	56.34%	DF	✓	✓						HOG+LUV	I
CrossTalk [27]	53.88%	DF			✓					HOG+LUV	I
DBN-Isol [28]	53.14%	DN					✓			HOG	I
ACF [29]	51.36%	DF	✓							HOG+LUV	I
RandForest [30]	51.17%	DF		✓						HOG+LBP	I&C
MultiFtr+Motion [22]	50.88%	DF	✓						✓	Many+Flow	T
SquaresChnFtrs [31]	50.17%	DF	✓							HOG+LUV	I
Franken [32]	48.68%	DF		✓						HOG+LUV	I
MultiResC [33]	48.45%	DPM			✓		✓	✓		HOG	C
Roerei [31]	48.35%	DF	✓					✓		HOG+LUV	I
DBN-Mut [34]	48.22%	DN			✓		✓			HOG	C
MF+Motion+2Ped [35]	46.44%	DF			✓				✓	Many+Flow	I+
MOCO [36]	45.53%	-	✓		✓					HOG+LBP	C
MultiSDP [37]	45.39%	DN	✓		✓	✓				HOG+CSS	C
ACF-Caltech [29]	44.22%	DF	✓							HOG+LUV	C
MultiResC+2Ped [35]	43.42%	DPM			✓		✓	✓		HOG	C+
WordChannels [38]	42.30%	DF	✓							Many	C
MT-DPM [39]	40.54%	DPM					✓	✓		HOG	C
JointDeep [40]	39.32%	DN			✓					Color+Gradient	C
SDN [41]	37.87%	DN				✓	✓			Pixels	C
MT-DPM+Context [39]	37.64%	DPM			✓		✓	✓		HOG	C+
ACF+SDt [42]	37.34%	DF	✓						✓	ACF+Flow	C+
SquaresChnFtrs [31]	34.81%	DF	✓							HOG+LUV	C
InformedHaar [43]	34.60%	DF	✓							HOG+LUV	C
Katamari-v1	22.49%	DF	✓		✓				✓	HOG+Flow	C+

What is driving the quality progress ?

- solution family (DPM, deep networks, decision forests)
- better classifiers
- deformable parts
- multi-scale models
- deep architectures
- training data
- additional (test time) data
- exploiting context
- better features

What is driving the quality progress ?

- solution family (DPM, deep networks, decision forests)
- better classifiers
- deformable parts
- multi-scale models
- deep architectures
- training data
- additional (test time) data
- exploiting context
- better features

What is driving the quality progress ?

- solution family
  (DPM, deep networks, decision forests)
- better classifiers

- deformable parts
- multi-scale models
- deep architectures
- training data
- additional (test time) data
- exploiting context
- better features

Surprise 1:

There is no clear winner regarding solution family (DPM, DN, or DF) or classifier type.

What is driving the quality progress ?

- ~~solution family (DPM, deep networks, decision forests)~~
- ~~better classifiers~~
- deformable parts
- multi-scale models
- deep architectures
- training data
- additional (test time) data
- exploiting context
- better features

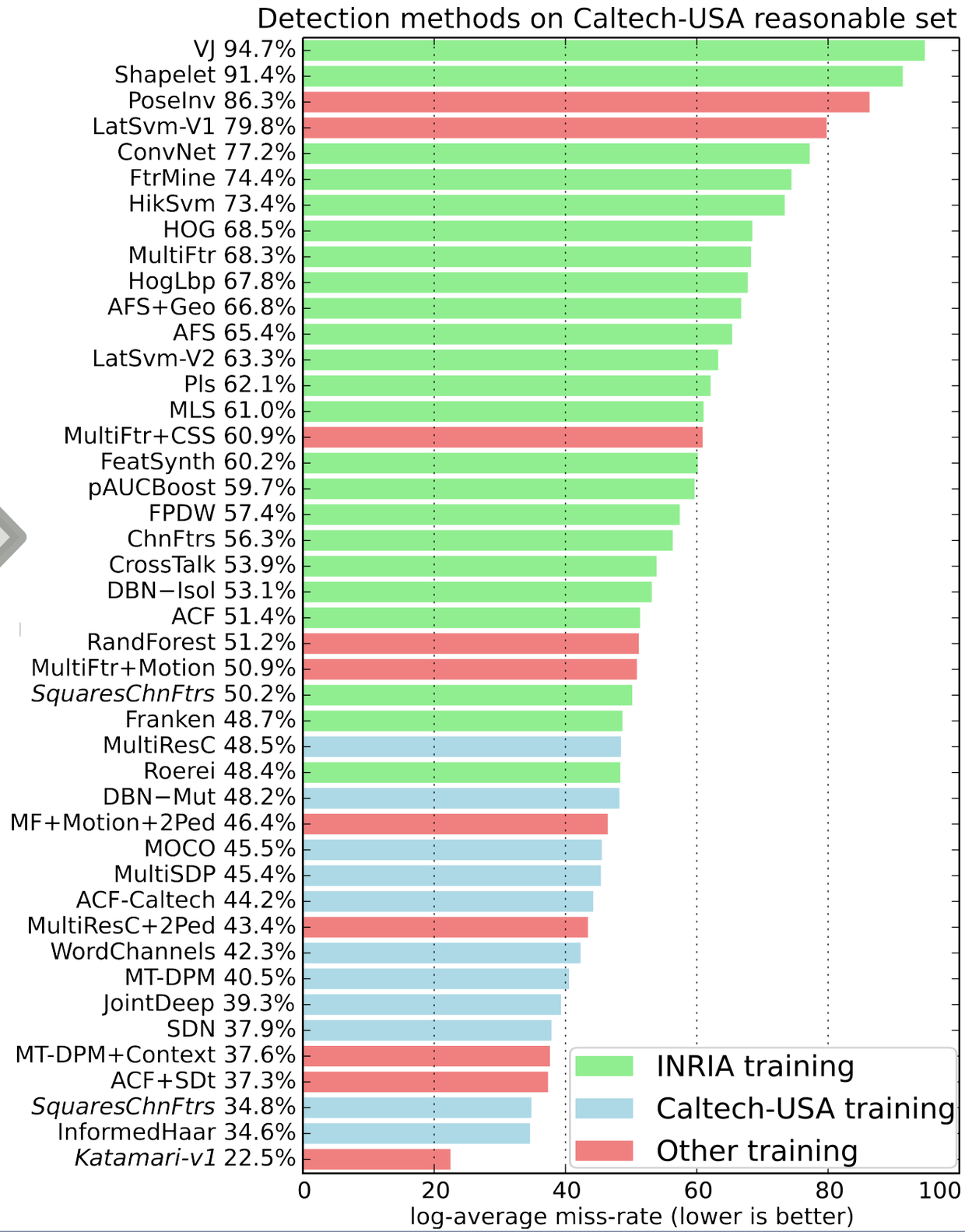
What is driving the quality progress ?

- ~~solution family (DPM, deep networks, decision forests)~~
- ~~better classifiers~~
- ~~deformable parts~~
- ~~multi-scale models~~
- ~~deep architectures~~

- training data
- additional (test time) data
- exploiting context
- better features

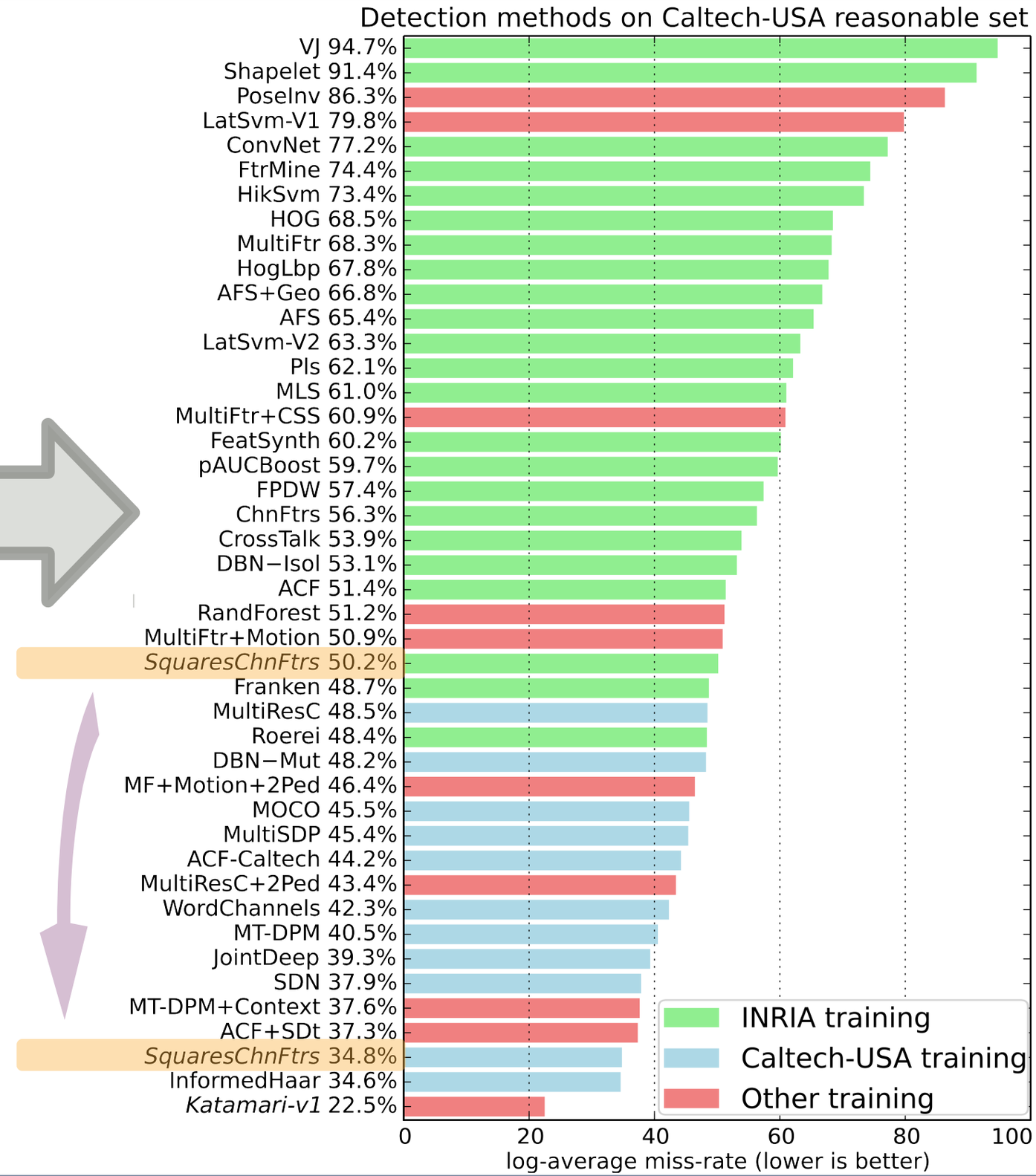
Training data matters (you knew this already)

Method	MR	Family	Features	Classifier	Context	Deep	Parts	M-Scales	More data	Feat. type	Training
VJ [9]	94.73%	DF	✓	✓						Haar	I
Shapelet [10]	91.37%	-	✓							Gradients	I
PoseInv [11]	86.32%	-					✓			HOG	I+
LatSvm-V1 [12]	79.78%	DPM					✓			HOG	P
ConvNet [13]	77.20%	DN				✓				Pixels	I
FtrMine [14]	74.42%	DF	✓							HOG+Color	I
HikSvm [15]	73.39%	-		✓						HOG	I
HOG [1]	68.46%	-	✓	✓						HOG	I
MultiFtr [16]	68.26%	DF	✓	✓						HOG+Haar	I
HogLbp [17]	67.77%	-	✓							HOG+LBP	I
AFS+Geo [18]	66.76%	-			✓					Custom	I
AFS [18]	65.38%	-								Custom	I
LatSvm-V2 [19]	63.26%	DPM		✓			✓			HOG	I
Pls [20]	62.10%	-	✓	✓						Custom	I
MLS [21]	61.03%	DF	✓							HOG	I
MultiFtr+CSS [22]	60.89%	DF	✓							Many	T
FeatSynth [23]	60.16%	-	✓	✓						Custom	I
pAUCBoost [24]	59.66%	DF	✓	✓						HOG+COV	I
FPDW [25]	57.40%	DF								HOG+LUV	I
ChnFtrs [26]	56.34%	DF	✓	✓						HOG+LUV	I
CrossTalk [27]	53.88%	DF			✓					HOG+LUV	I
DBN-Isol [28]	53.14%	DN					✓			HOG	I
ACF [29]	51.36%	DF	✓							HOG+LUV	I
RandForest [30]	51.17%	DF		✓						HOG+LBP	I&C
MultiFtr+Motion [22]	50.88%	DF	✓						✓	Many+Flow	T
SquaresChnFtrs [31]	50.17%	DF	✓							HOG+LUV	I
Franken [32]	48.68%	DF		✓						HOG+LUV	I
MultiResC [33]	48.45%	DPM			✓		✓	✓		HOG	C
Roerei [31]	48.35%	DF	✓					✓		HOG+LUV	I
DBN-Mut [34]	48.22%	DN			✓		✓			HOG	C
MF+Motion+2Ped [35]	46.44%	DF			✓				✓	Many+Flow	I+
MOCO [36]	45.53%	-	✓		✓					HOG+LBP	C
MultiSDP [37]	45.39%	DN	✓		✓	✓				HOG+CSS	C
ACF-Caltech [29]	44.22%	DF	✓							HOG+LUV	C
MultiResC+2Ped [35]	43.42%	DPM			✓		✓	✓		HOG	C+
WordChannels [38]	42.30%	DF	✓							Many	C
MT-DPM [39]	40.54%	DPM					✓	✓		HOG	C
JointDeep [40]	39.32%	DN			✓					Color+Gradient	C
SDN [41]	37.87%	DN				✓	✓			Pixels	C
MT-DPM+Context [39]	37.64%	DPM			✓		✓	✓		HOG	C+
ACF+SDt [42]	37.34%	DF	✓						✓	ACF+Flow	C+
SquaresChnFtrs [31]	34.81%	DF	✓							HOG+LUV	C
InformedHaar [43]	34.60%	DF	✓							HOG+LUV	C
Katamari-v1	22.49%	DF	✓		✓				✓	HOG+Flow	C+



Training data matters (you knew this already)

Method	MR	Family	Features	Classifier	Context	Deep	Parts	M-Scales	More data	Feat. type	Training
VJ [9]	94.73%	DF	✓	✓						Haar	I
Shapelet [10]	91.37%	-	✓							Gradients	I
PoseInv [11]	86.32%	-					✓			HOG	I+
LatSvm-V1 [12]	79.78%	DPM					✓			HOG	P
ConvNet [13]	77.20%	DN				✓				Pixels	I
FtrMine [14]	74.42%	DF	✓							HOG+Color	I
HikSvm [15]	73.39%	-		✓						HOG	I
HOG [1]	68.46%	-	✓	✓						HOG	I
MultiFtr [16]	68.26%	DF	✓	✓						HOG+Haar	I
HogLbp [17]	67.77%	-	✓							HOG+LBP	I
AFS+Geo [18]	66.76%	-			✓					Custom	I
AFS [18]	65.38%	-								Custom	I
LatSvm-V2 [19]	63.26%	DPM		✓			✓			HOG	I
Pls [20]	62.10%	-	✓	✓						Custom	I
MLS [21]	61.03%	DF	✓							HOG	I
MultiFtr+CSS [22]	60.89%	DF	✓							Many	T
FeatSynth [23]	60.16%	-	✓	✓						Custom	I
pAUCBoost [24]	59.66%	DF	✓	✓						HOG+COV	I
FPDW [25]	57.40%	DF								HOG+LUV	I
ChnFtrs [26]	56.34%	DF	✓	✓						HOG+LUV	I
CrossTalk [27]	53.88%	DF			✓					HOG+LUV	I
DBN-Isol [28]	53.14%	DN					✓			HOG	I
ACF [29]	51.36%	DF	✓							HOG+LUV	I
RandForest [30]	51.17%	DF		✓						HOG+LBP	I&C
MultiFtr+Motion [22]	50.88%	DF	✓						✓	Many+Flow	T
SquaresChnFtrs [31]	50.17%	DF	✓							HOG+LUV	I
Franken [32]	48.68%	DF		✓						HOG+LUV	I
MultiResC [33]	48.45%	DPM			✓		✓	✓		HOG	C
Roerei [31]	48.35%	DF	✓					✓		HOG+LUV	I
DBN-Mut [34]	48.22%	DN			✓		✓			HOG	C
MF+Motion+2Ped [35]	46.44%	DF			✓				✓	Many+Flow	I+
MOCO [36]	45.53%	-	✓		✓					HOG+LBP	C
MultiSDP [37]	45.39%	DN	✓		✓	✓				HOG+CSS	C
ACF-Caltech [29]	44.22%	DF	✓							HOG+LUV	C
MultiResC+2Ped [35]	43.42%	DPM			✓		✓	✓		HOG	C+
WordChannels [38]	42.30%	DF	✓							Many	C
MT-DPM [39]	40.54%	DPM					✓	✓		HOG	C
JointDeep [40]	39.32%	DN			✓					Color+Gradient	C
SDN [41]	37.87%	DN				✓	✓			Pixels	C
MT-DPM+Context [39]	37.64%	DPM			✓		✓	✓		HOG	C+
ACF+SDt [42]	37.34%	DF	✓						✓	ACF+Flow	C+
SquaresChnFtrs [31]	34.81%	DF	✓							HOG+LUV	C
InformedHaar [43]	34.60%	DF	✓							HOG+LUV	C
Katamari-v1	22.49%	DF	✓		✓				✓	HOG+Flow	C+



What is driving the quality progress ?

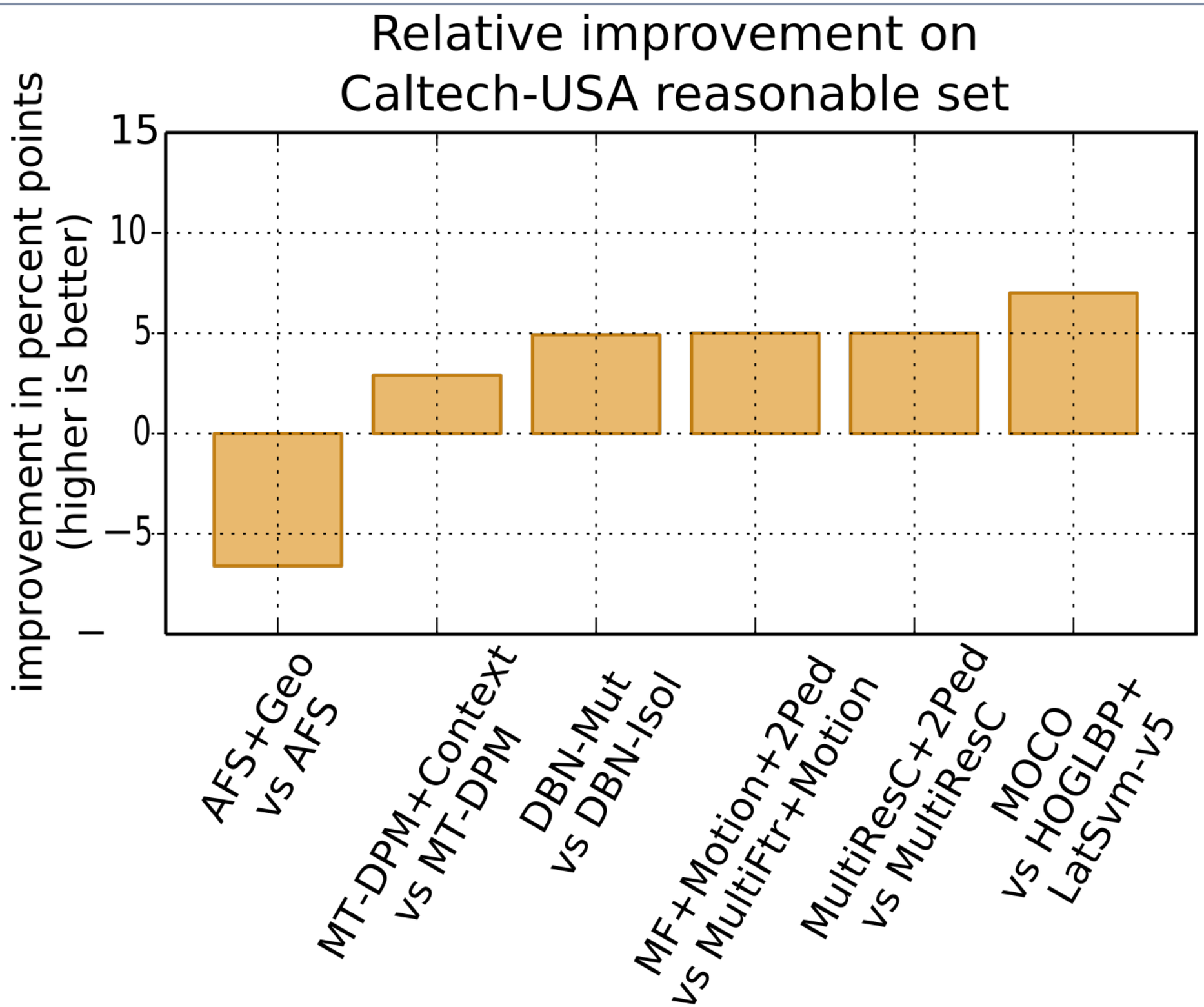
- ~~solution family (DPM, deep networks, decision forests)~~
- ~~better classifiers~~
- ~~deformable parts~~
- ~~multi-scale models~~
- ~~deep architectures~~
- ~~training data~~
- additional (test time) data

  ⇨ using more frames (flow or stereo) helps (you knew this already)
- exploiting context
- better features

What is driving the quality progress ?

- ~~solution family (DPM, deep networks, decision forests)~~
- ~~better classifiers~~
- ~~deformable parts~~
- ~~multi-scale models~~
- ~~deep architectures~~
- ~~training data~~
- ~~additional (test time) data~~
- exploiting context
- better features

Using context helps (expect ~5 pp improvement)



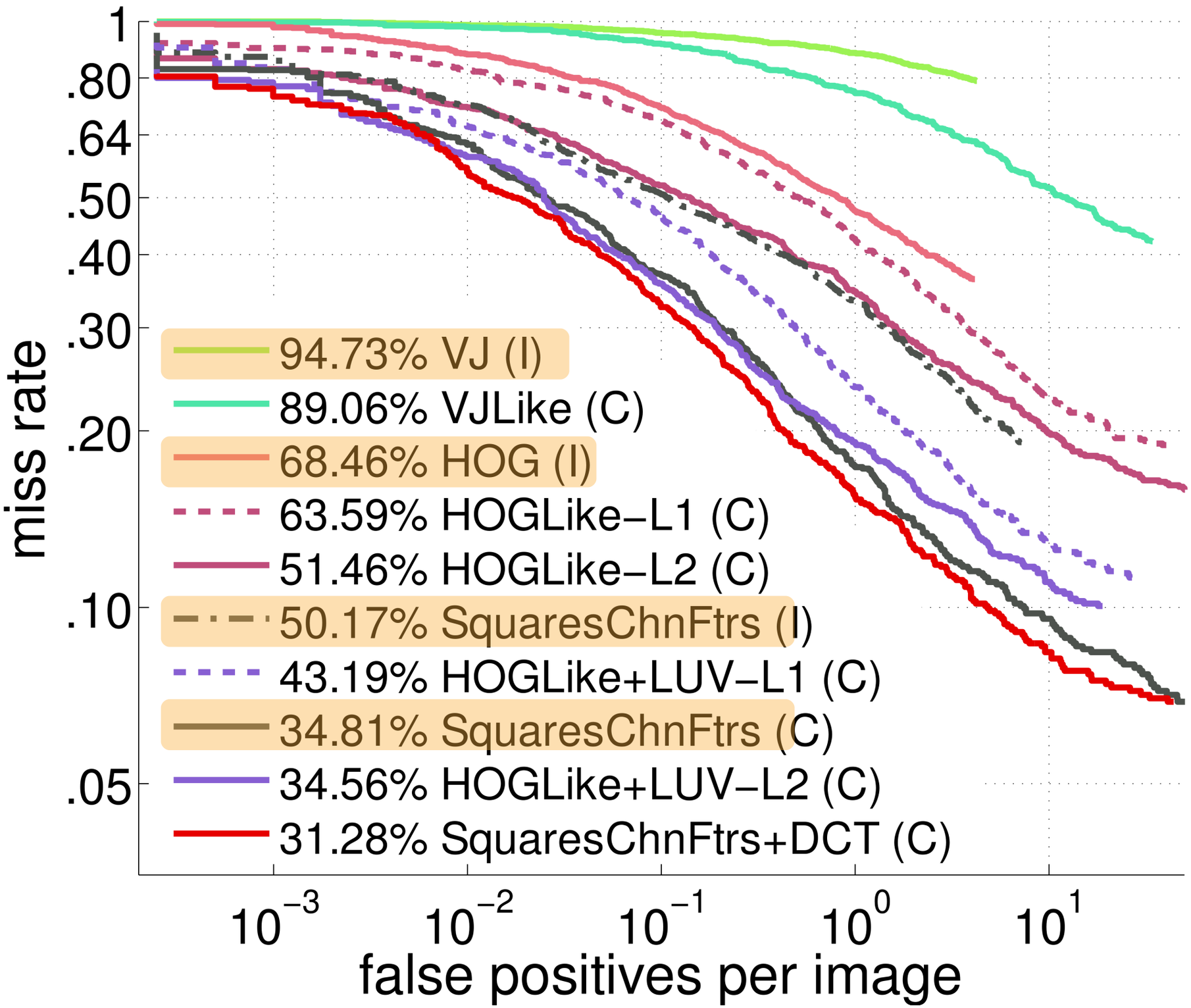
What is driving the quality progress ?

- ~~solution family (DPM, deep networks, decision forests)~~
- ~~better classifiers~~
- ~~deformable parts~~
- ~~multi-scale models~~
- ~~deep architectures~~
- ~~training data~~
- ~~additional (test time) data~~
- ~~exploiting context~~
- better features

Experiments

(some of them)

Features alone can explain 10 years of progress



What is driving the quality progress ?

- ~~solution family (DPM, deep networks, decision forests)~~
- ~~better classifiers~~
- ~~deformable parts~~
- ~~multi-scale models~~
- ~~deep architectures~~
- ~~training data~~
- additional (test time) data
- exploiting context
- better features

Strong features/Flow/Context are very complementary

Method	Results	Improvement	Expected improvement
`SquaresChnFtrs`	34.81%	-	-
+Better features (`DCT`)	31.28%	3.53	-
+Flow (`SDt`)	30.34%	4.47	-
+Context (`2Ped`)	29.42%	5.39	-

Results in MR (lower is better). Improvement in MR percent points.

[DCT: Nam et al. ArXiv 2014]
[SDt: Park et al. CVPR 2013]
[2Ped: Ouyang & Wang CVPR 2013]

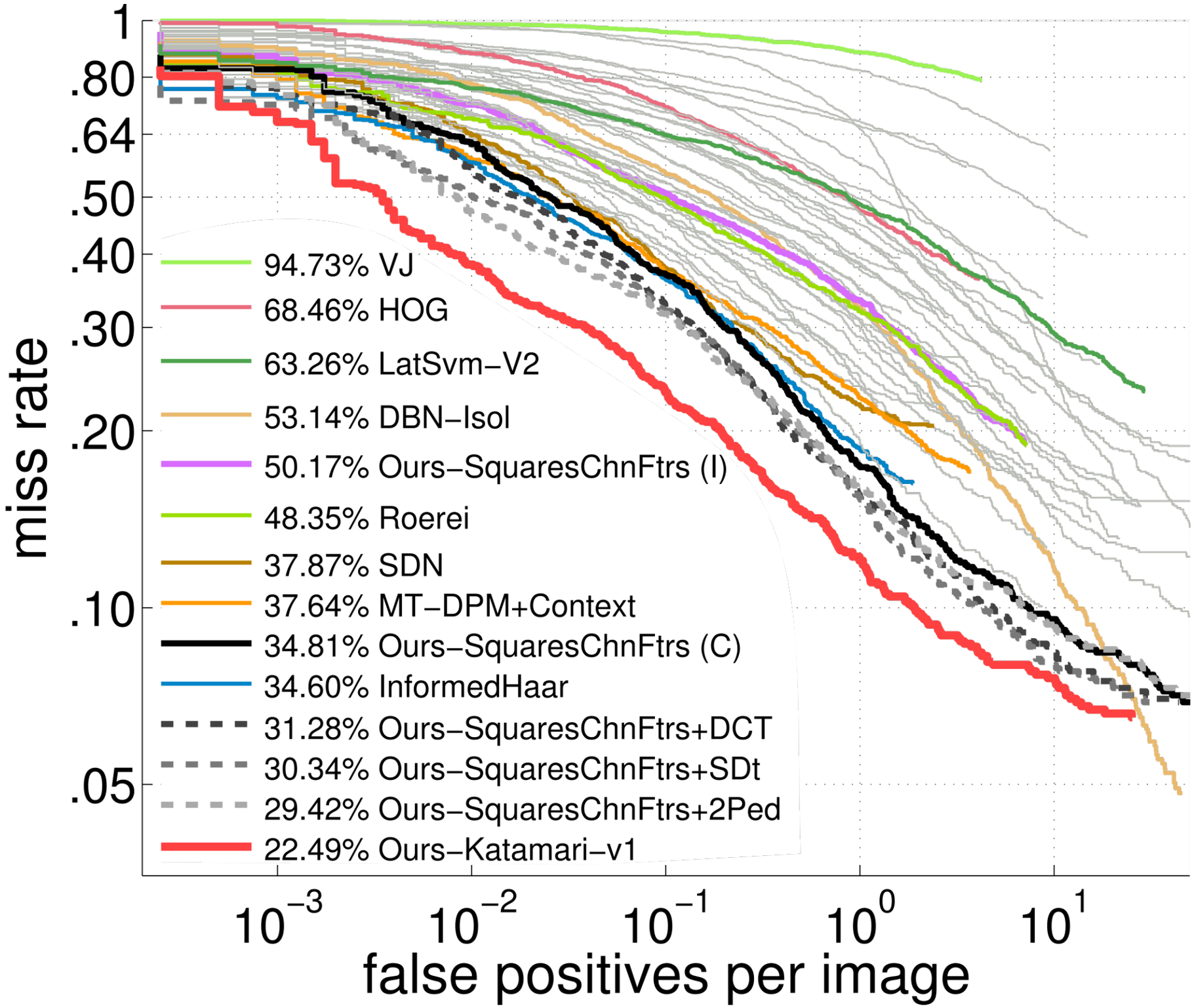
Strong features/Flow/Context are very complementary

Method	Results	Improvement	Expected improvement
SquaresChnFtrs	34.81%	-	-
+Better features (DCT)	31.28%	3.53	-
+Flow (SDt)	30.34%	4.47	-
+Context (2Ped)	29.42%	5.39	-
+DCT+2Ped	27.40%	7.41	8.92
+SDt+2Ped	26.68%	8.13	9.86
+DCT+SDt	25.24%	9.57	8.00
All-in-one (Katamari)	*22.49%*	12.32	13.39

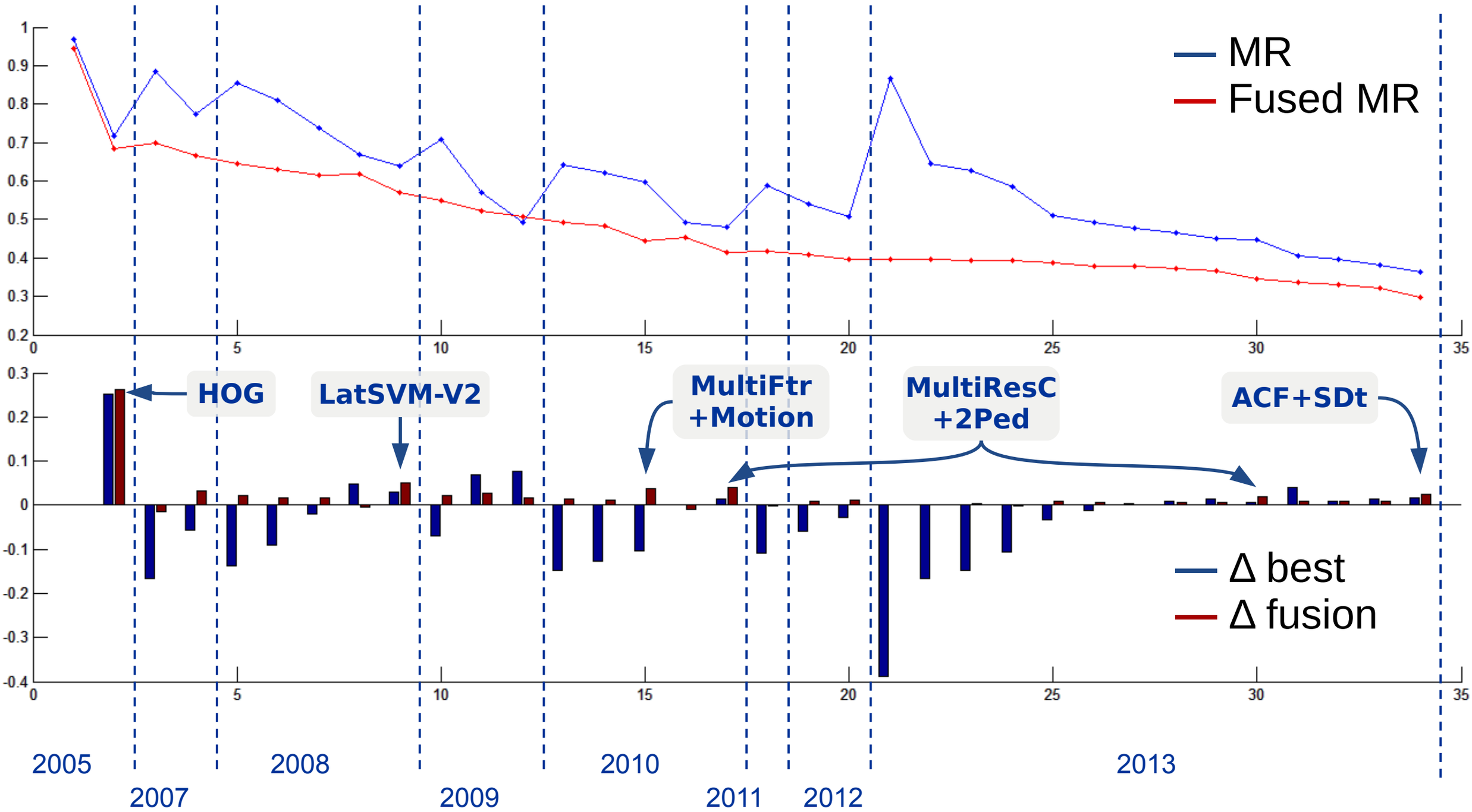
Results in MR (lower is better). Improvement in MR percent points.

Surprise 2: no diminishing return observed (yet).

Strong features/Flow/Context are very complementary

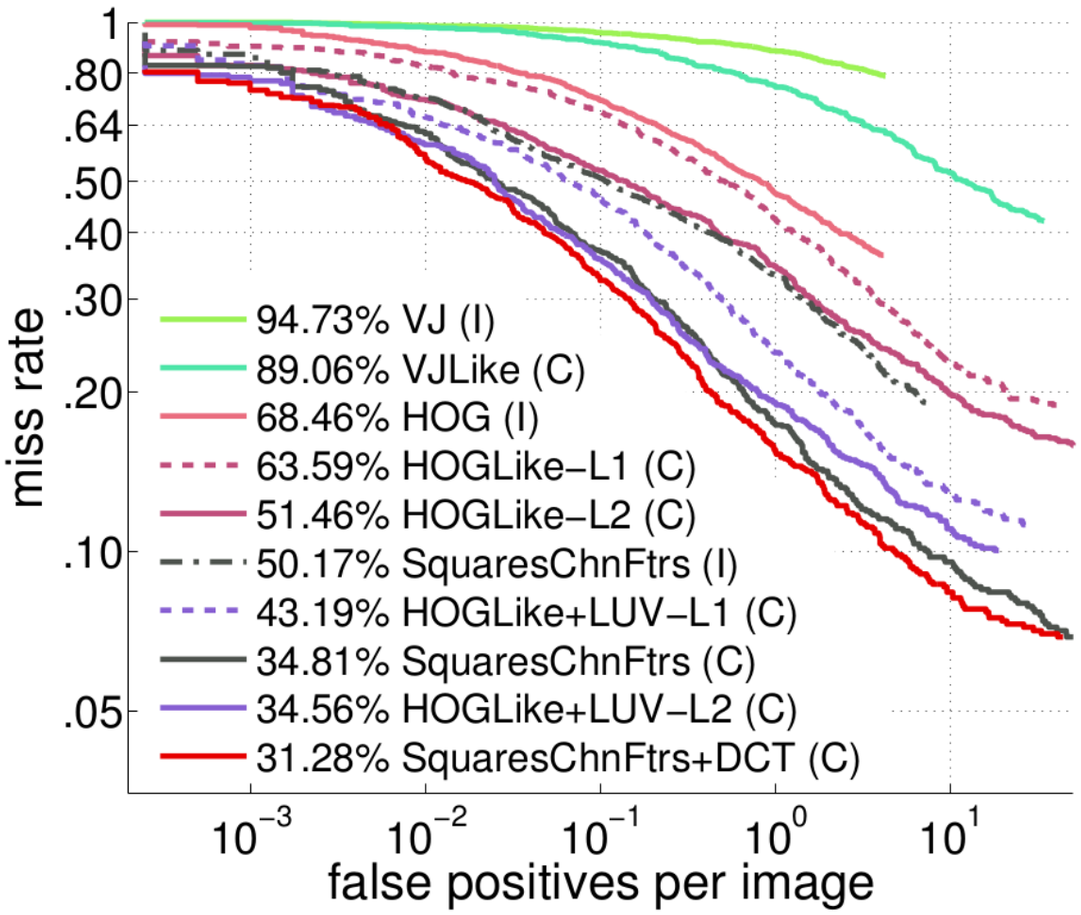


Merging all methods over time



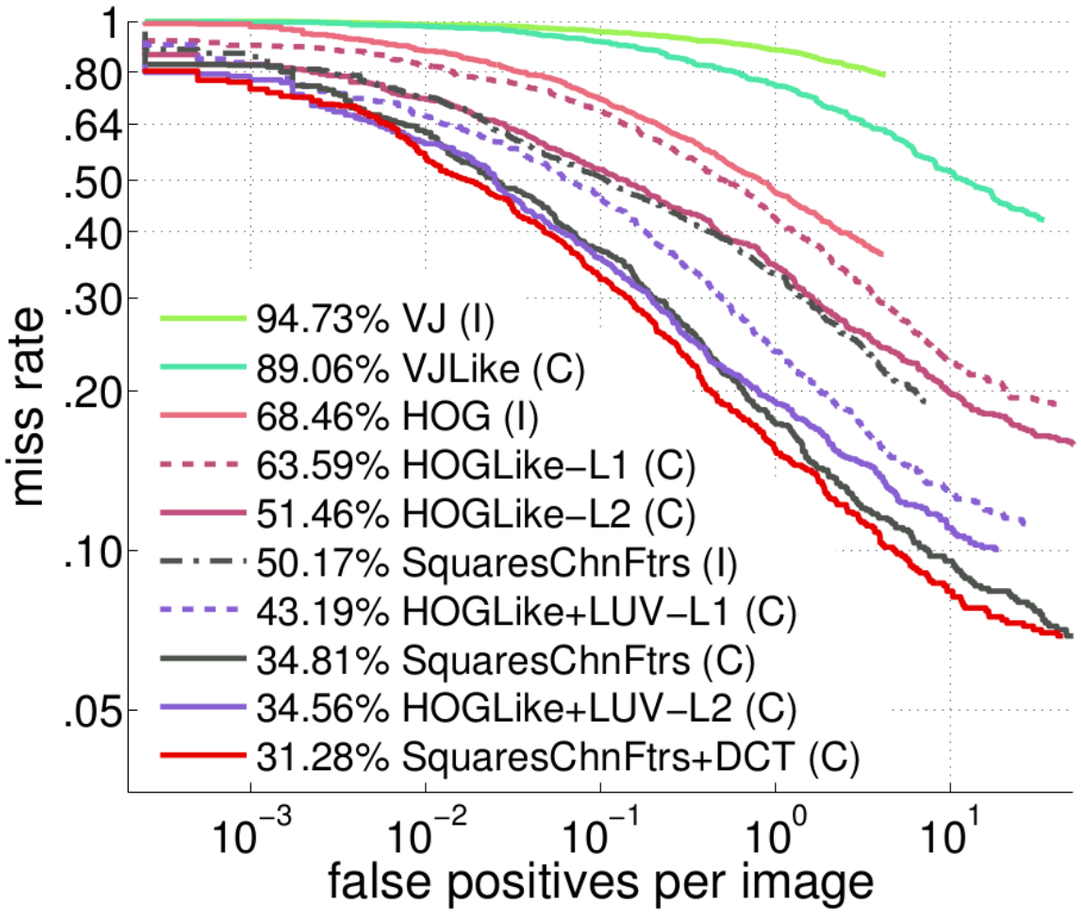
Slide from [Xu et al. BMVC 2014]

Surprise 3: Model capacity has not saturated

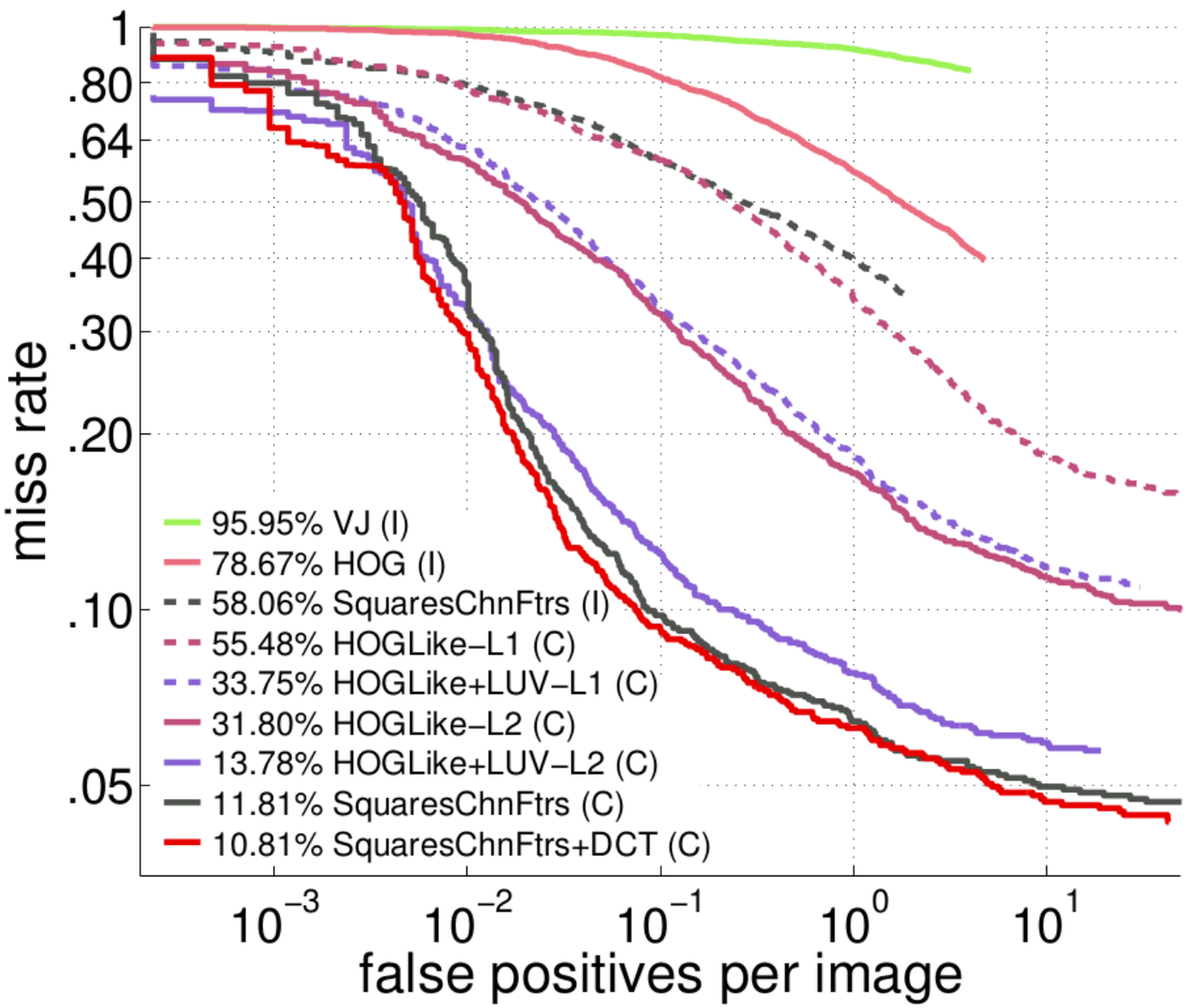


Caltech-USA
test set

Surprise 3: Model capacity has not saturated



Caltech-USA
test set



Caltech-USA
training set

What have we learned ?

- “Sooner or later, everything old is new again.” - Stephen King
  Decade-old ideas still rule detection quality.
- Switching training data is not comparing apples-to-apples.
- Flow, context, and strong features are very complementary (still).
- All other aspects have yet to make a “definitive statement”.
- Features alone can explain a decade of detection quality progress.
- There is room for further improvement
  by increasing model capacity (and better features).

How to further improve quality ?

- Stronger use of additional data
  (scene flow on KITTI ?)
- Better context
  (exploiting scene geometry)
- Further developing deep architectures
  (end-to-end fine tuning)
- Most importantly: understanding
  what makes good features good?


Questions ?

See you at the poster !

?

Rodrigo Benenson

http://rodrigob.github.com

max planck institut
informatik

What is driving the quality progress ?

- ~~solution family (DPM, deep networks, decision forests)~~
- ~~better classifiers~~
- deformable parts

  ⇨ no clear case for their use in pedestrians
- multi-scale models
- deep architectures
- training data
- additional (test time) data
- exploiting context
- better features

What is driving the quality progress ?

- ~~solution family (DPM, deep networks, decision forests)~~
- ~~better classifiers~~
- ~~deformable parts~~
- multi-scale models

  ⇨ helps a bit, but not key for quality in Caltech-USA
- deep architectures
- training data
- additional (test time) data
- exploiting context
- better features

What is driving the quality progress ?

- ~~solution family (DPM, deep networks, decision forests)~~
- ~~better classifiers~~
- ~~deformable parts~~
- ~~multi-scale models~~
- deep architectures

  ⇨ active area, has yet to reach top quality: CVPR 2015 ?
- training data
- additional (test time) data
- exploiting context
- better features