Introspective Classification for Pedestrian Detection

Calum Blair

John Thompson
Institute for Digital Communications,
University of Edinburgh

Neil Robertson
Visionlab,
Heriot-Watt University

SSPD 2014
Edinburgh
8th September 2014

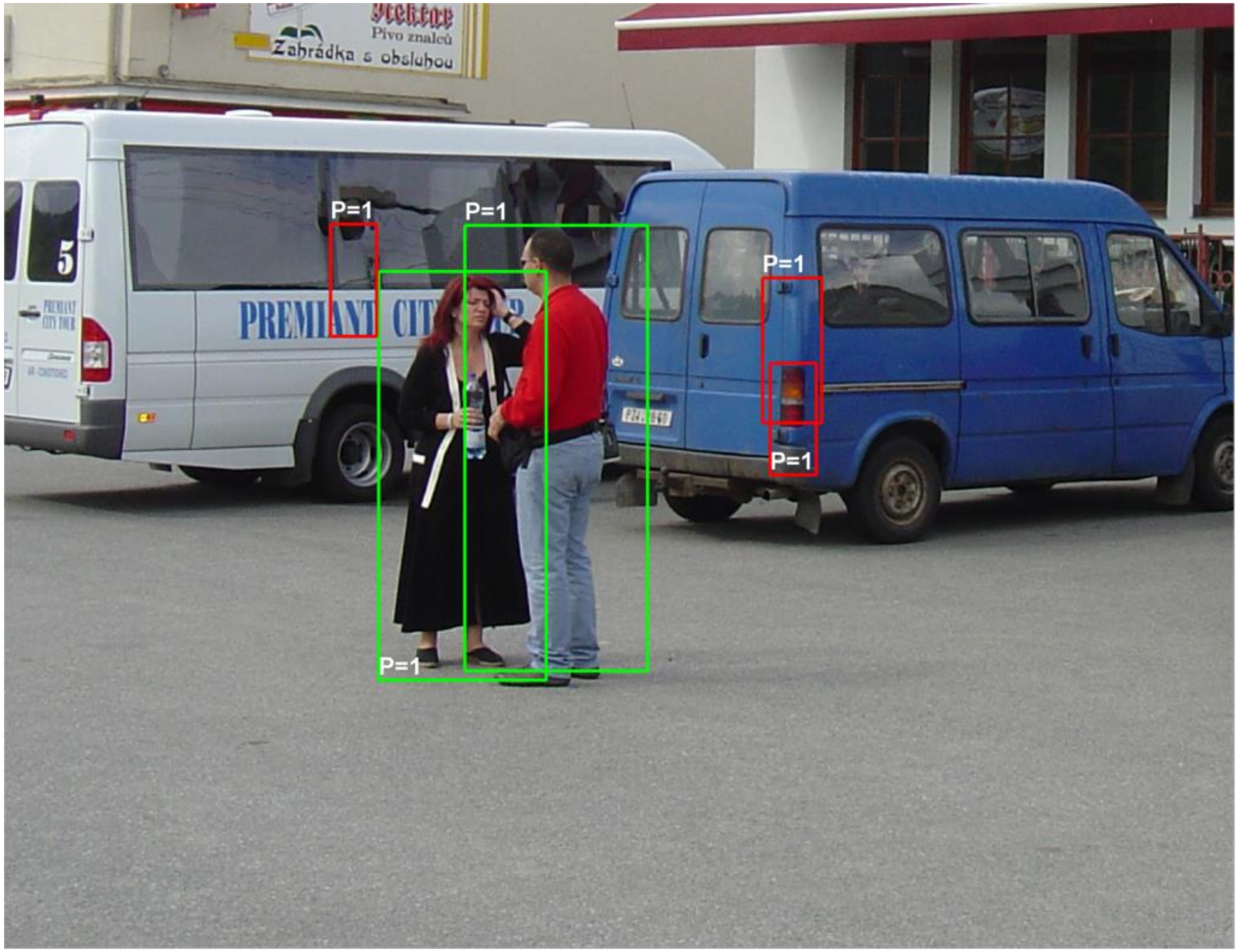
THE UNIVERSITY of EDINBURGH

HERIOT WATT UNIVERSITY

dstl

EPSRC
Engineering and Physical Sciences Research Council

Motivation


Pivo znalců
Zahrádka s obsluhou
5
PREMIANT CITY
P=1
P=1
P=1
P=1
P=1

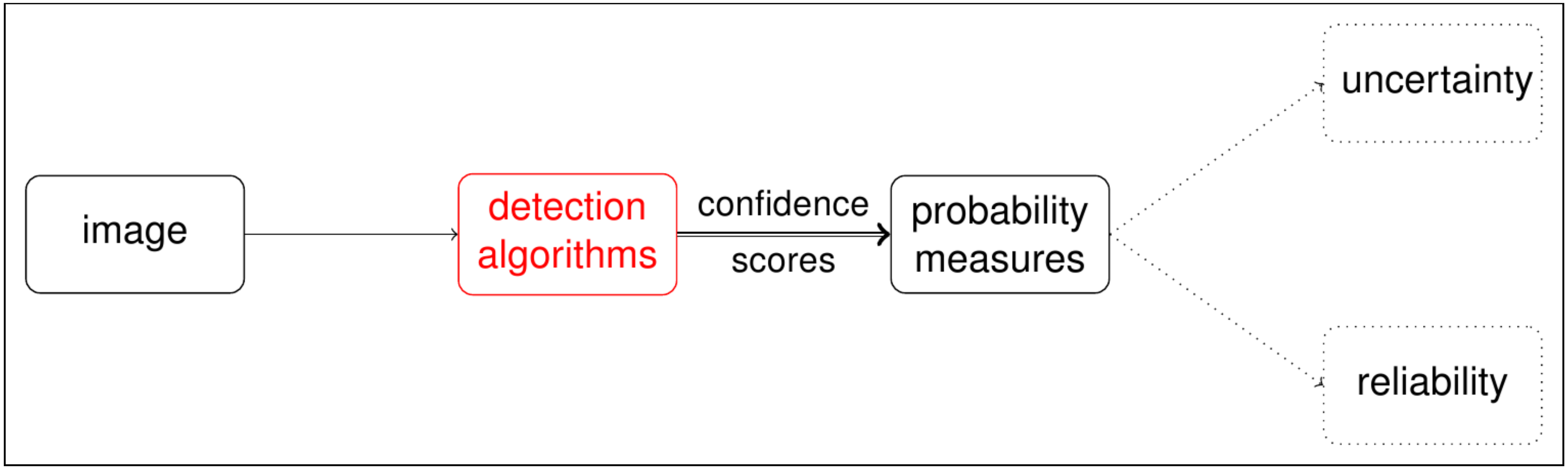
Motivation

- Machine learning in battlespace scenarios: algorithm reliability & effect on operator trust
- Timely & accurate detections
- Separate *accuracy* (% of samples right) from *reliability* (variations in confidence measure)
- Detector performance comparison
- [dstl] technical challenges:
  - #27 (SWaP reduction)
  - #29 (Accreditable machine learning)
- Pedestrian → other objects & modalities

Previous Work

- Grimmet, Paul, et al: Introspective classification for mission-critical decision making (2013):
  - Traffic light recognition (goal: autonomous system)
  - Detector which knows when it is uncertain: "reflect an amount of ambiguity appropriate to a given situation".
  - Gaussian Processes (GPs) perform well, but compute- and memory-intensive
- Trade off speed and accuracy? Still true with 'harder' objects (when $p_d < 99\%$) ?

Structure



- Evaluate classification approaches & probability generation techniques
- Results - misclassification & reliability

State of the art:

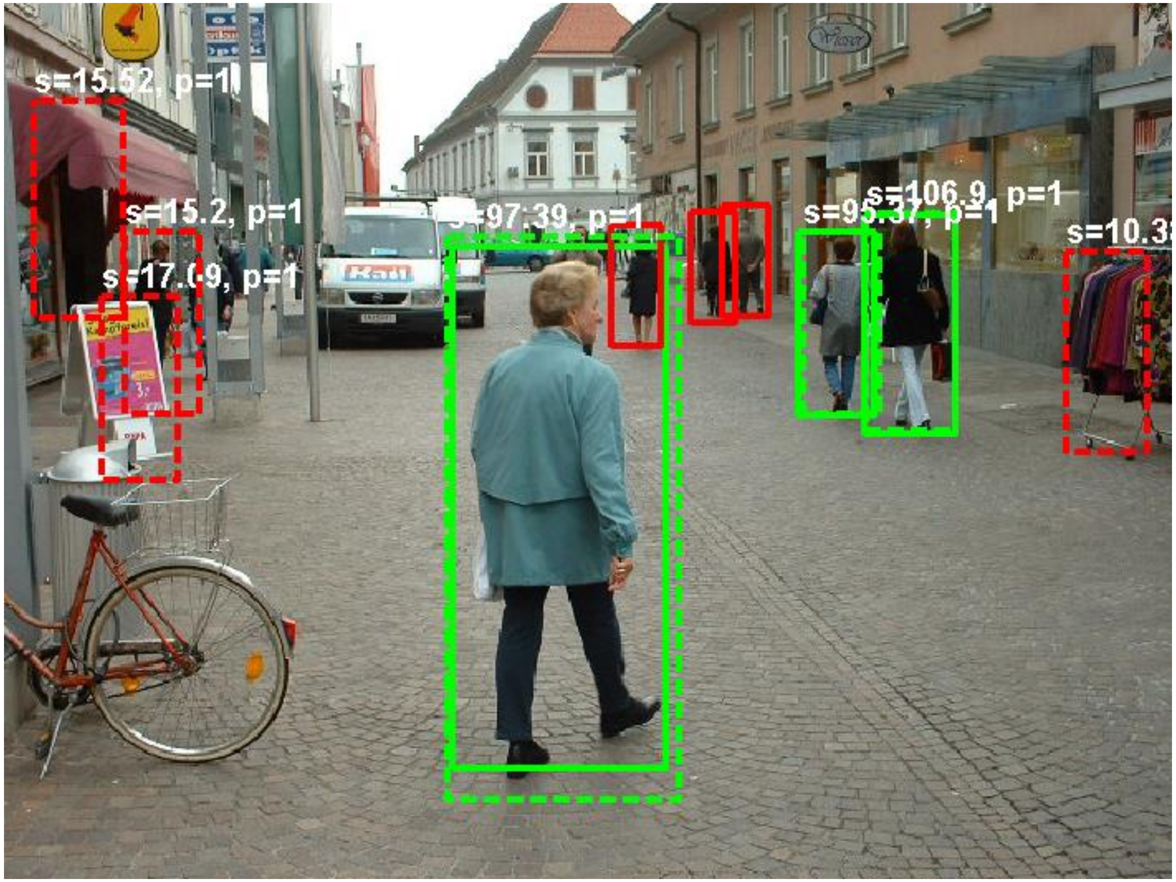
ACF (Dollar et al, PAMI 2014):

- Classifier based on random forests
- 2048 trees in model
- Confidence output: [-200: 200], based on interaction of test window's feature vector with trees

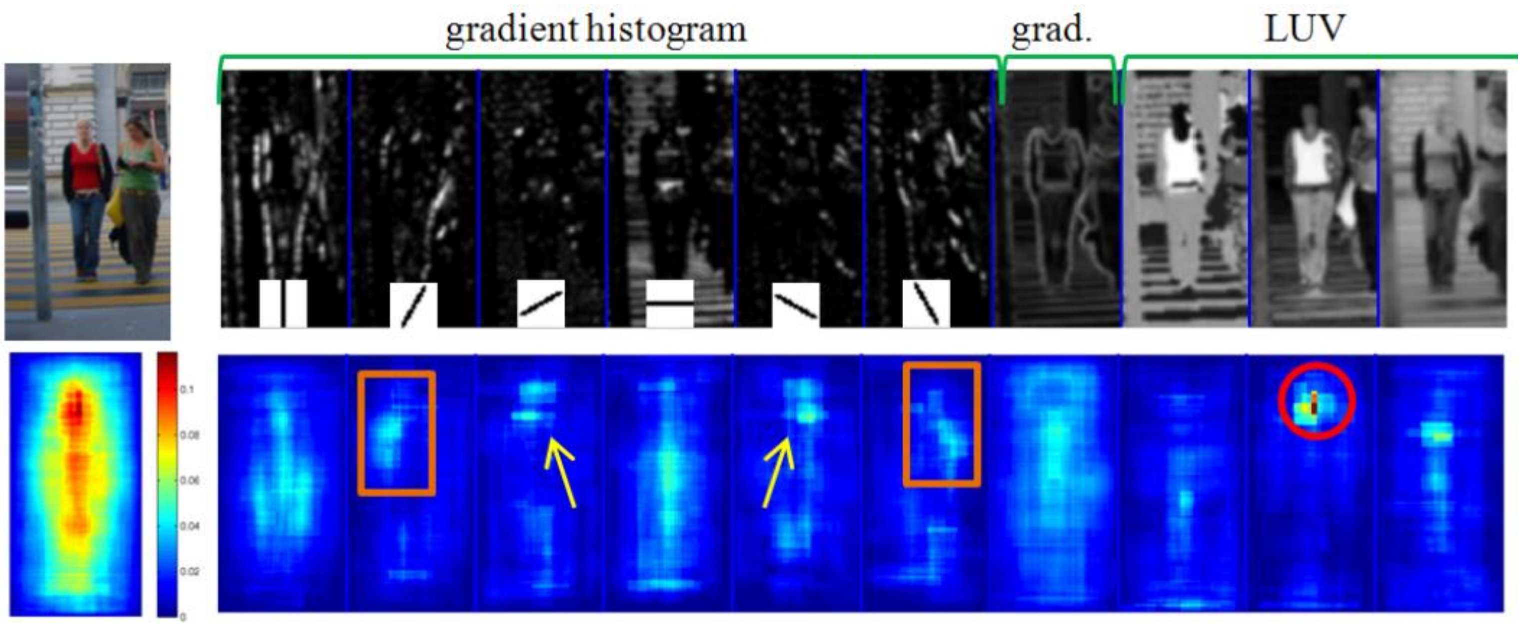
Squash score with sigmoid to obtain detection probability measure :

$$p = \frac{1}{\exp(-2 * score)}$$

Use extracted features as starting point for detection eval.



Feature extraction (ACF)

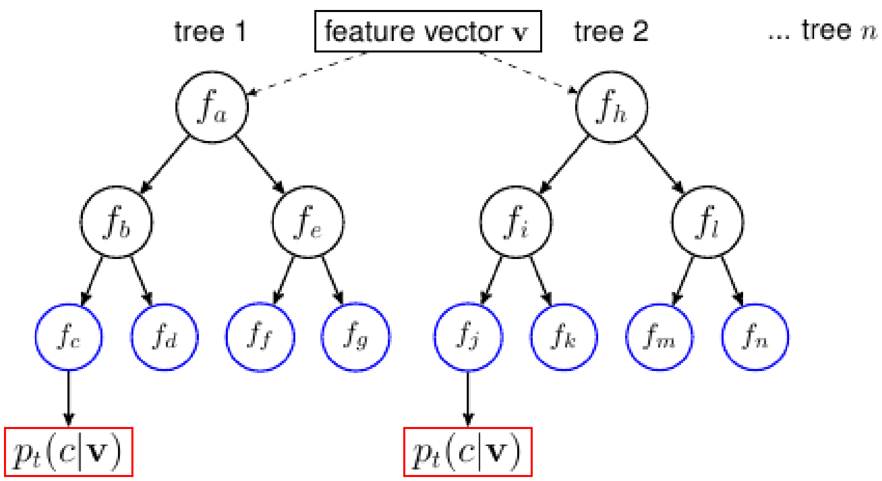


64x128 window, d=5120 feature vector
Common approach (HOG etc)
Starting point for classifier algorithms
INRIA pedestrian dataset: 1150 training & 450 test images

Dollar et al, Integral Channel Features, 2009

Adaboost/Random Forests

- Adaboost cascade classifier:
  - learn best features & thresholds at each decision tree
  - sum weights over all trees to get score *s*
  - aim is to produce strong decisions over training data, not give confidence score for borderline samples



$$s = \frac{1}{T}\sum_{k=1}^{T} w_k p_t(c|\mathbf{v})$$

Support Vectors

- Maximum margin classifier: best separation of classes
- Confidence score increases further from hyperplane: not always representative?
- Linear kernel (accuracy vs runtime compromise)

$$f(\mathbf{x}) = \sum_{i=1}^{N} (\mathbf{x}_i \cdot w_i) + b$$

Gaussian Processes

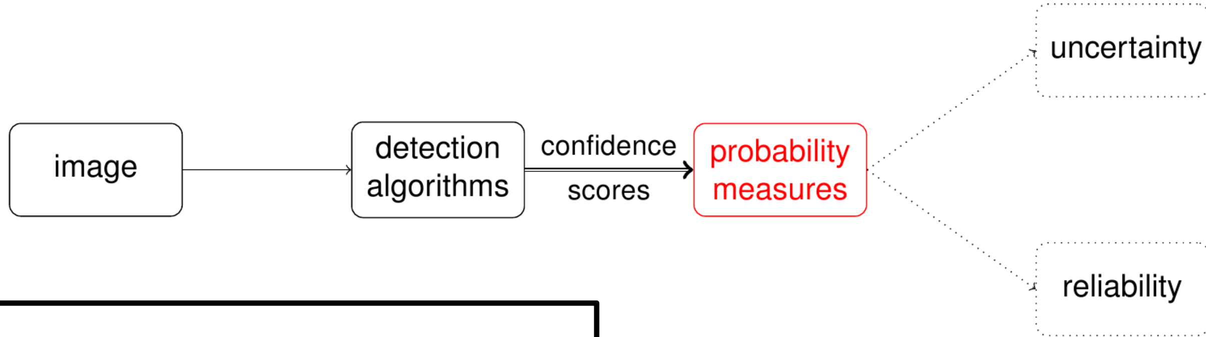
- Form Gaussian distribution based on training, test data.

$$p(\mathbf{y}_T|\mathbf{y}) = \mathcal{N}(\boldsymbol{\mu}_T, \boldsymbol{\Sigma}_T)$$

- Gaussian processes classifier (GP)
  - obtain predictive variance by computing covariance of *entire training set* with test sample
  - $O(n^2)$ over test data
  - Probabilistic output given directly
  - GPU: 4x speedup but still not realtime

$$\mathbf{K}_{N+T} = \begin{bmatrix} \mathbf{K}_N & \mathbf{K}_{NT} \\ \mathbf{K}_{TN} & \mathbf{K}_T \end{bmatrix}$$

$$k(x) = exp(-\frac{(\mathbf{x_i} - \mathbf{x_j})^2}{2\ell^2})$$



sigmoid:

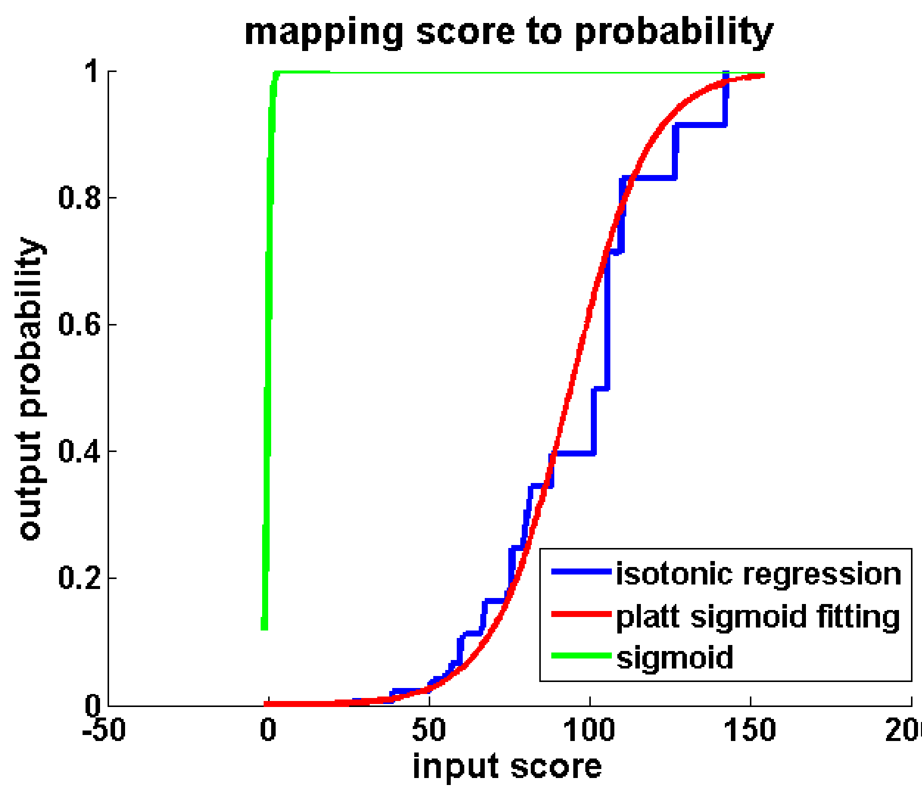
$$p(y = C_k \mid X) = \frac{1}{1 + \exp(-2 * score)}$$

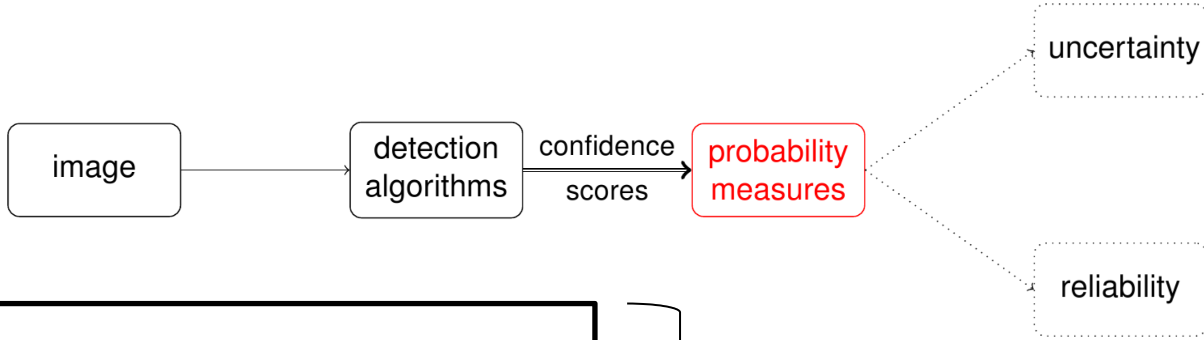
Platt:

$$p(y = C_k \mid X) = \frac{1}{1 + \exp(a * score + b)}$$

a, b learned from validation set

Isotonic Regression:
assign output values to input range;
learn on validation set





sigmoid:

$$p(y = C_k \mid X) = \frac{1}{1 + \exp(-2 * score)}$$

Platt:

$$p(y = C_k \mid X) = \frac{1}{1 + \exp(a * score + b)}$$

a, b learned from validation set

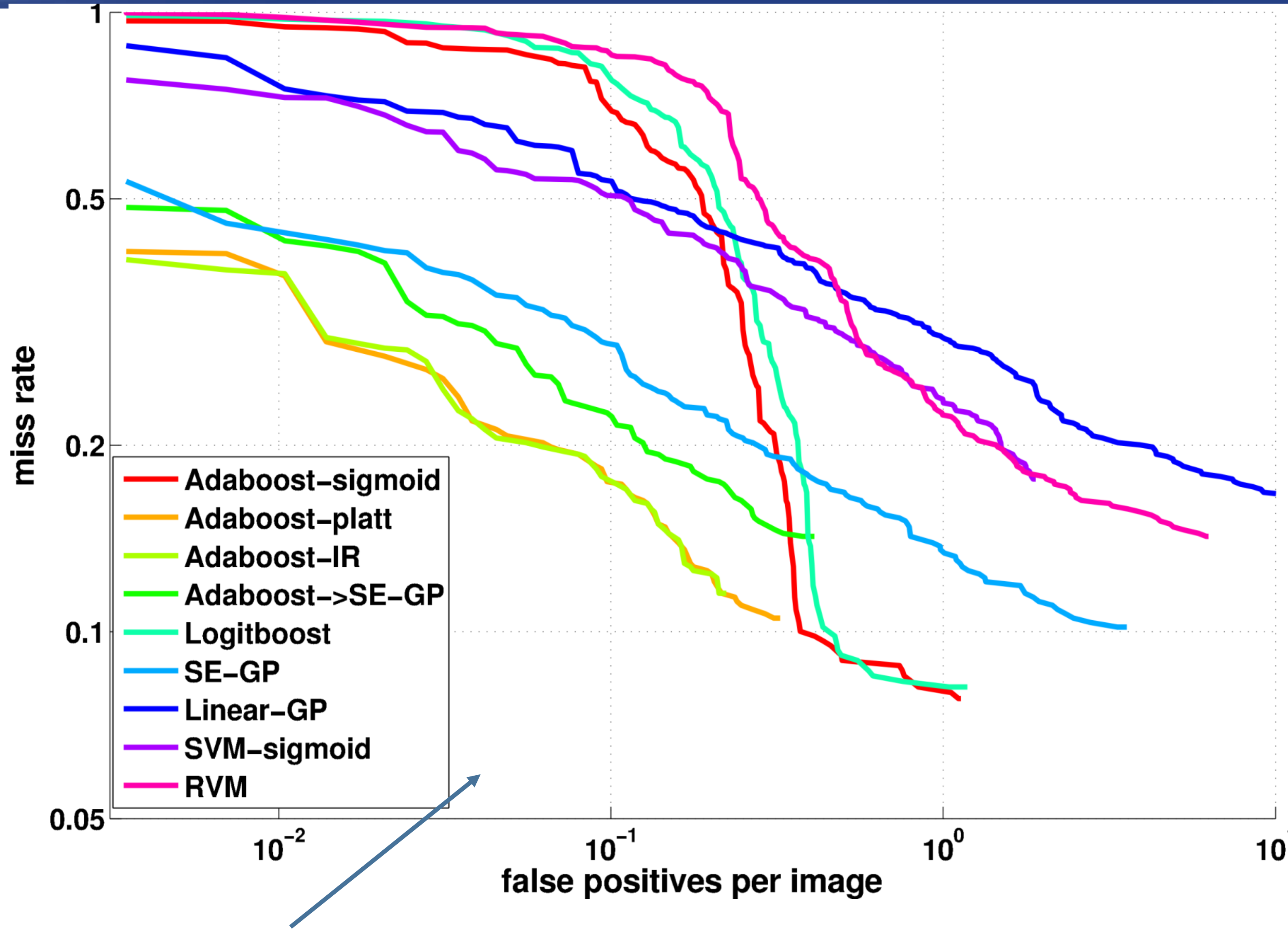
Isotonic Regression:
assign output values to input range;
learn on validation set

Apply to adaboost & SVM; GP generates probabilities anyway.

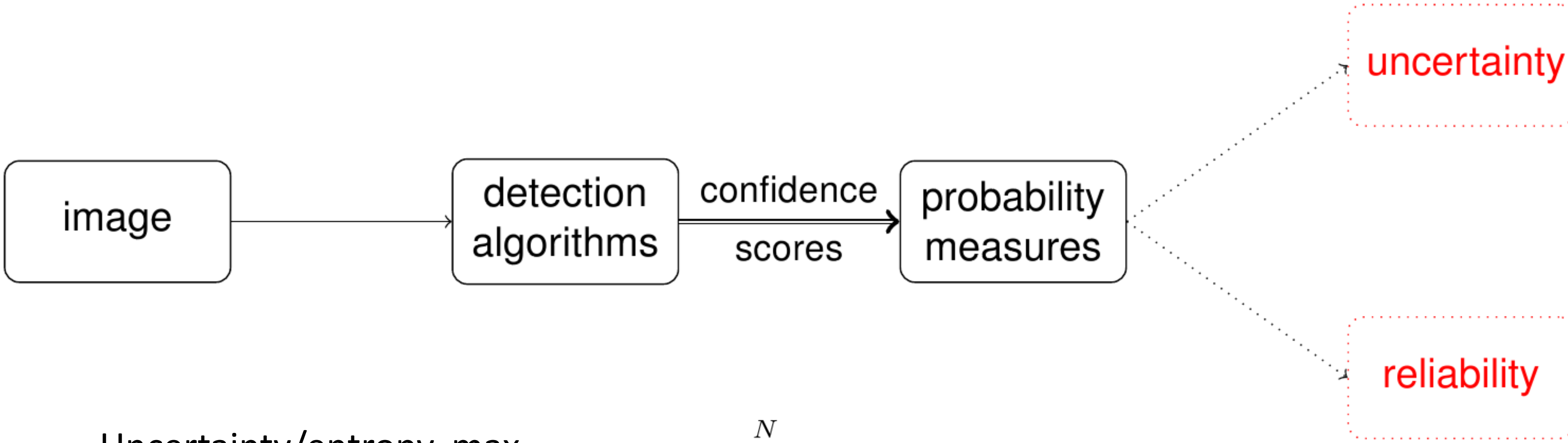
Misclassification performance

Name	True Pos.	False Neg.	False Pos.	F1 score	frame runtime (s)
Adaboost-sigmoid	**543**	**46**	326	0.745	**0.058**
Adaboost-platt	527	62	93	0.872	**0.058**
Adaboost-IR	521	68	**64**	**0.888**	0.061
Adaboost→SE-GP*	505	84	118	0.833	(variable)
Logitboost	541	48	341	0.735	0.06
SE-GP	529	60	1030	0.492	142
Lin-GP	500	89	4792	0.169	120
Lin-SVM	485	104	548	0.598	0.45
RVM	505	84	1811	0.348	8.68

*Run SE-GP *only* on window locations identified by adaboost



Good classifers here (low fppi and miss rate)



Uncertainty/entropy, max when all $p(C|x) = 1/N$:

$$H = -\sum_{k=1}^{N}[p(C_k|\mathbf{x})\log_N(p(C_k|\mathbf{x}))]$$

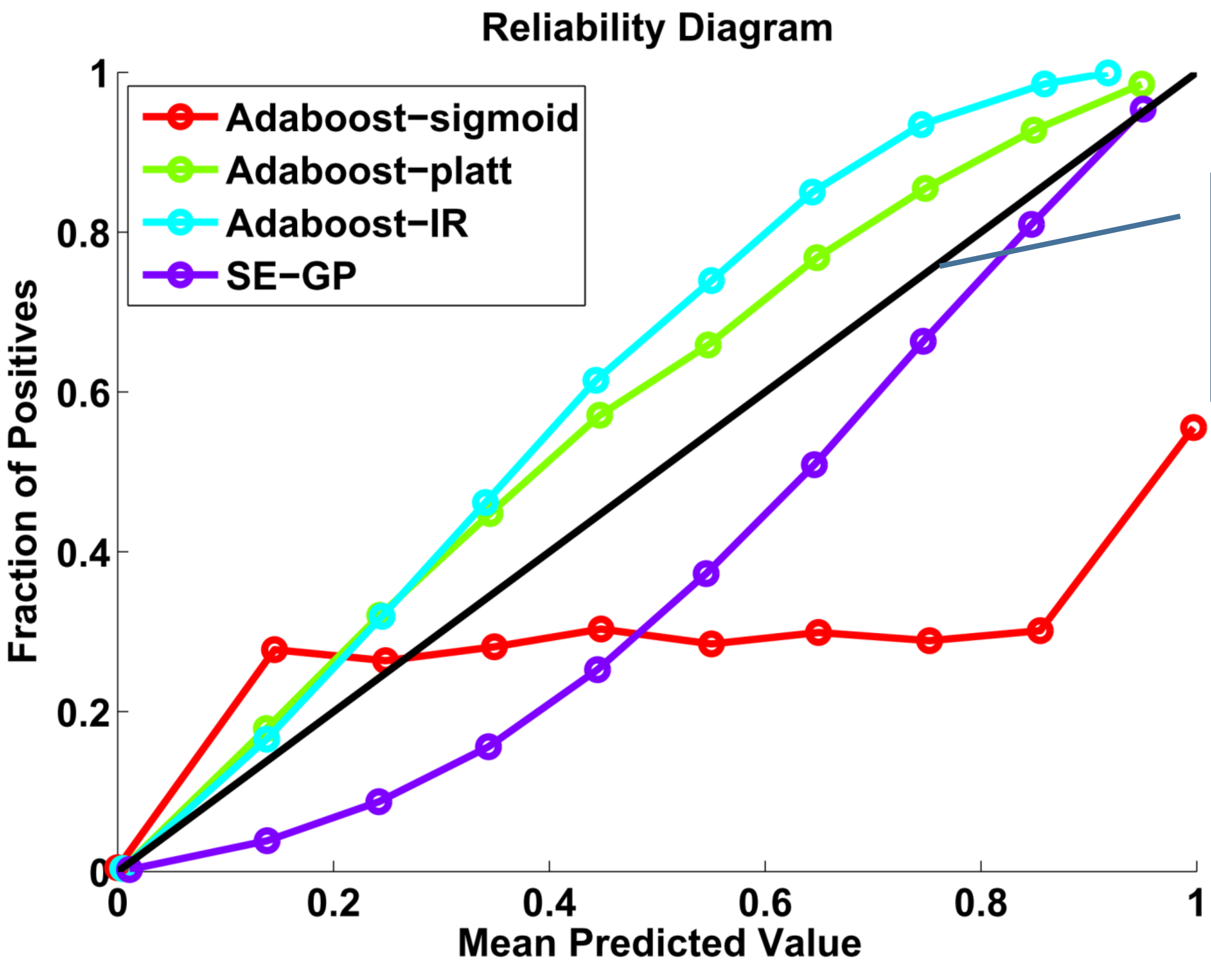
Mean-squared error penalises *confident* wrong classifications more than *uncertain* ones:

$$MSE = \frac{2}{N}\sum_{k=1}^{N}(C_k - p(1|\mathbf{x}_i))^2$$

Reliability results

Name	Area Under Curve	Mean Squared Error	frame runtime (s)
Adaboost-sigmoid	0.760	0.797	**0.058**
Adaboost-platt	0.834	**0.331**	**0.058**
Adaboost-IR	0.807	0.346	0.061
Adaboost→SE-GP	0.796	0.410	(variable)
Logitboost	0.749	0.813	0.06
SE-GP	**0.865**	0.595	142
Lin-GP	0.806	0.978	120
Lin-SVM	0.722	0.691	0.45
RVM	0.787	1.207	8.68

"How often did a prediction with e.g. p=0.8 result in a true positive?"
Modified to show discarded stages.



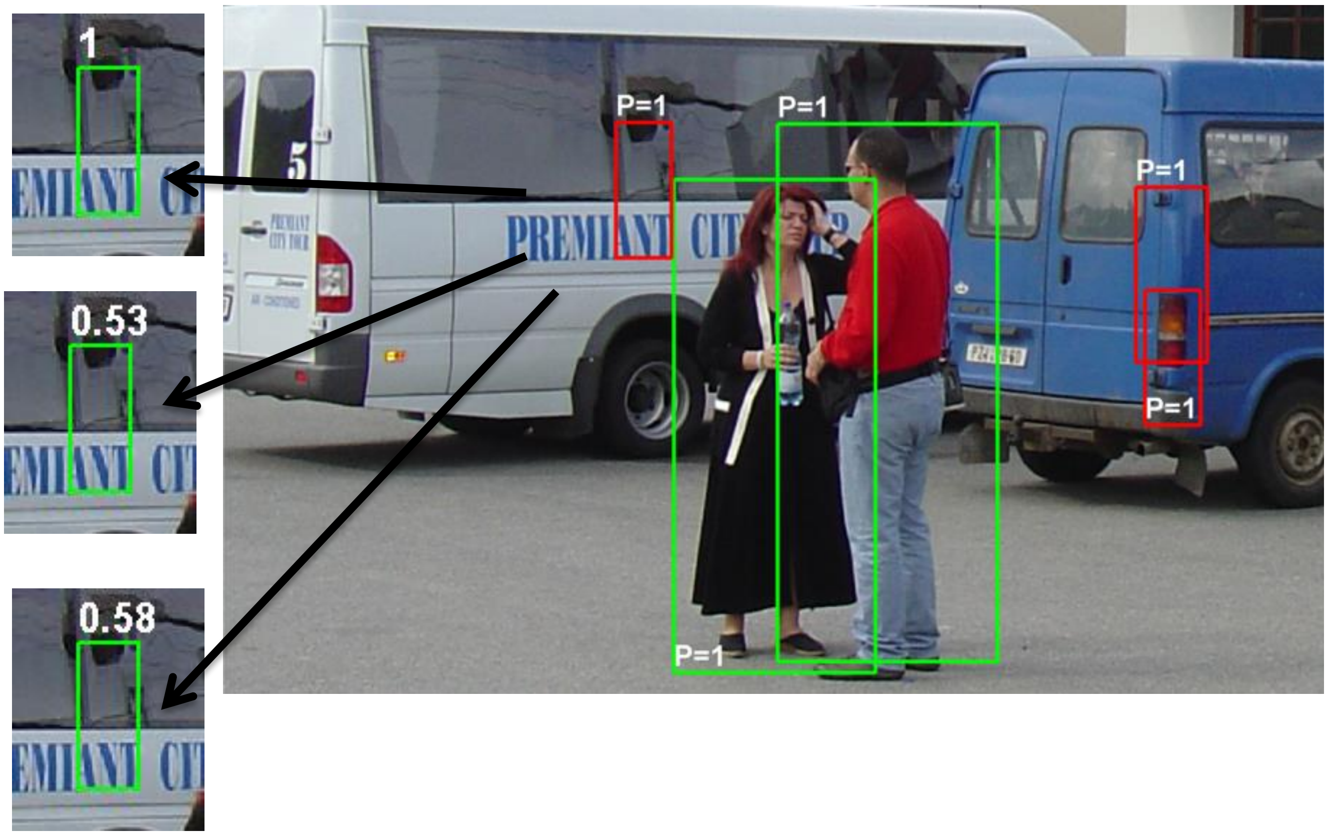
"Well-calibrated" classifier should lie on black line

A reminder…

Sigmoid:

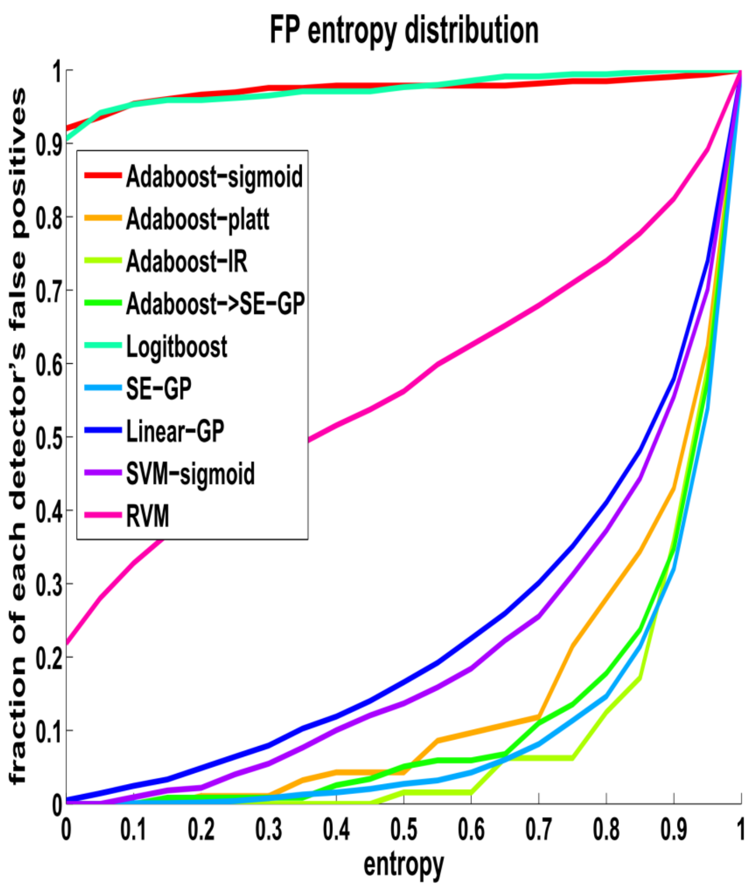
Adaboost → GP:

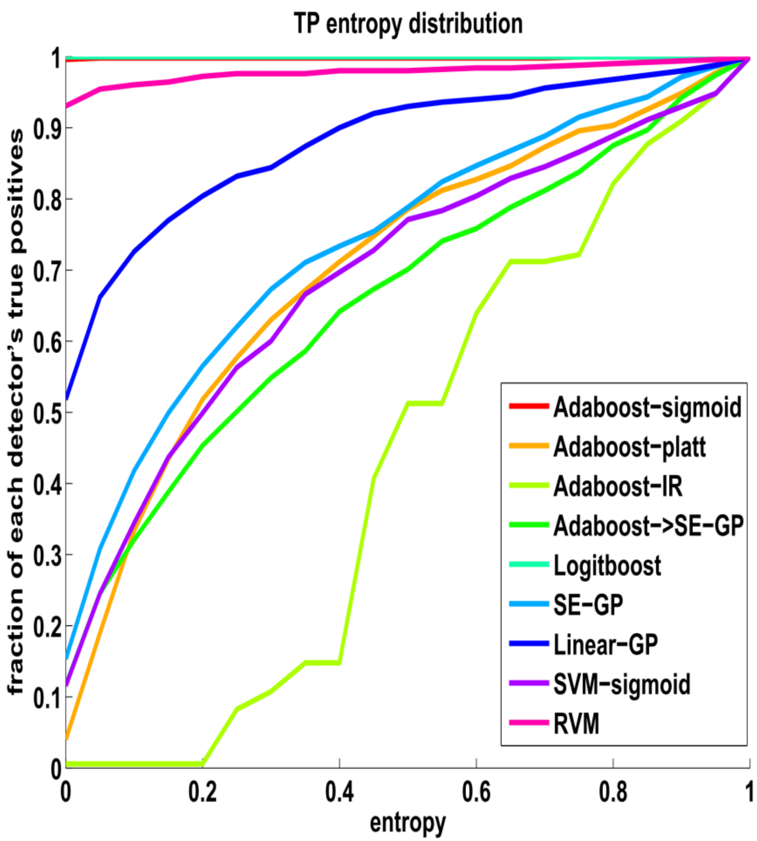
Platt:



- Active learning
- More expensive algorithms
- Unknown class identification

Uncertainty





[dstl]

EPSRC
Engineering and Physical Sciences Research Council

Summary

- GPs & probability generation techniques effective at separating uncertain from certain detections.
- GPs arguably not worth extra computational cost -- even when Adaboost used to filter negatives.
- Future work: apply to more classes/modalities

Detector performance in practice