Training with corrupted labels to reinforce a probably correct team-sport player detector

Berk Sevilmiş, Pascaline Parisot, Christophe De Vleeschouwer

Université catholique de Louvain

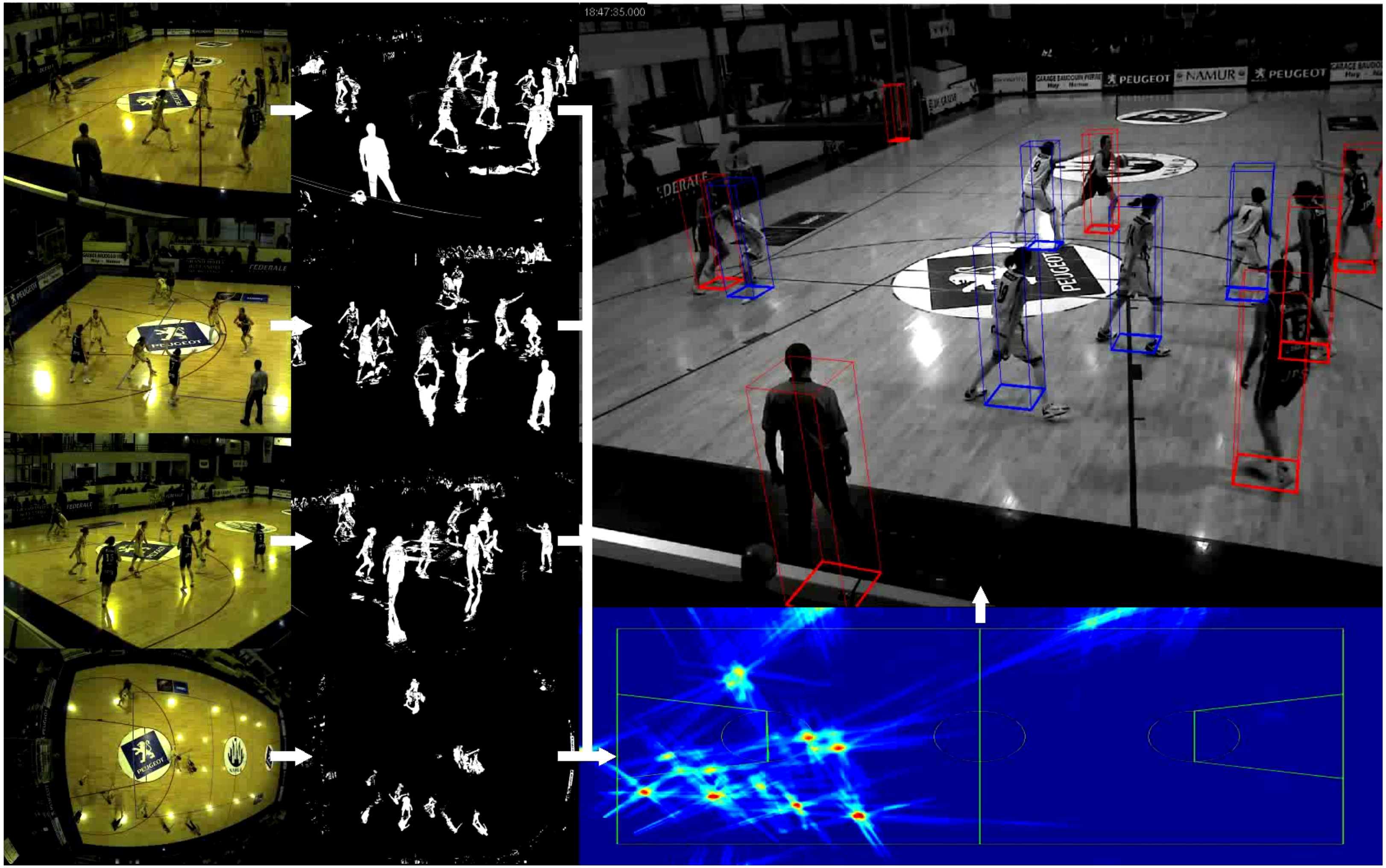
ISPS

June 5, 2013

Player detection reinforcement

- Goal
  - Exploit visual features to improve foreground-based detection.
- Main ideas
  - Use a classifier to select true positives among the candidates identified based on foreground detector.
  - Originality: train with samples labeled by the initial detector → potentially corrupted labels.
- Constraints
  - Appearance of the object and of the scene changes from one game to the other.
  - Player => large range of deformation.

Multi-view foreground-based detection



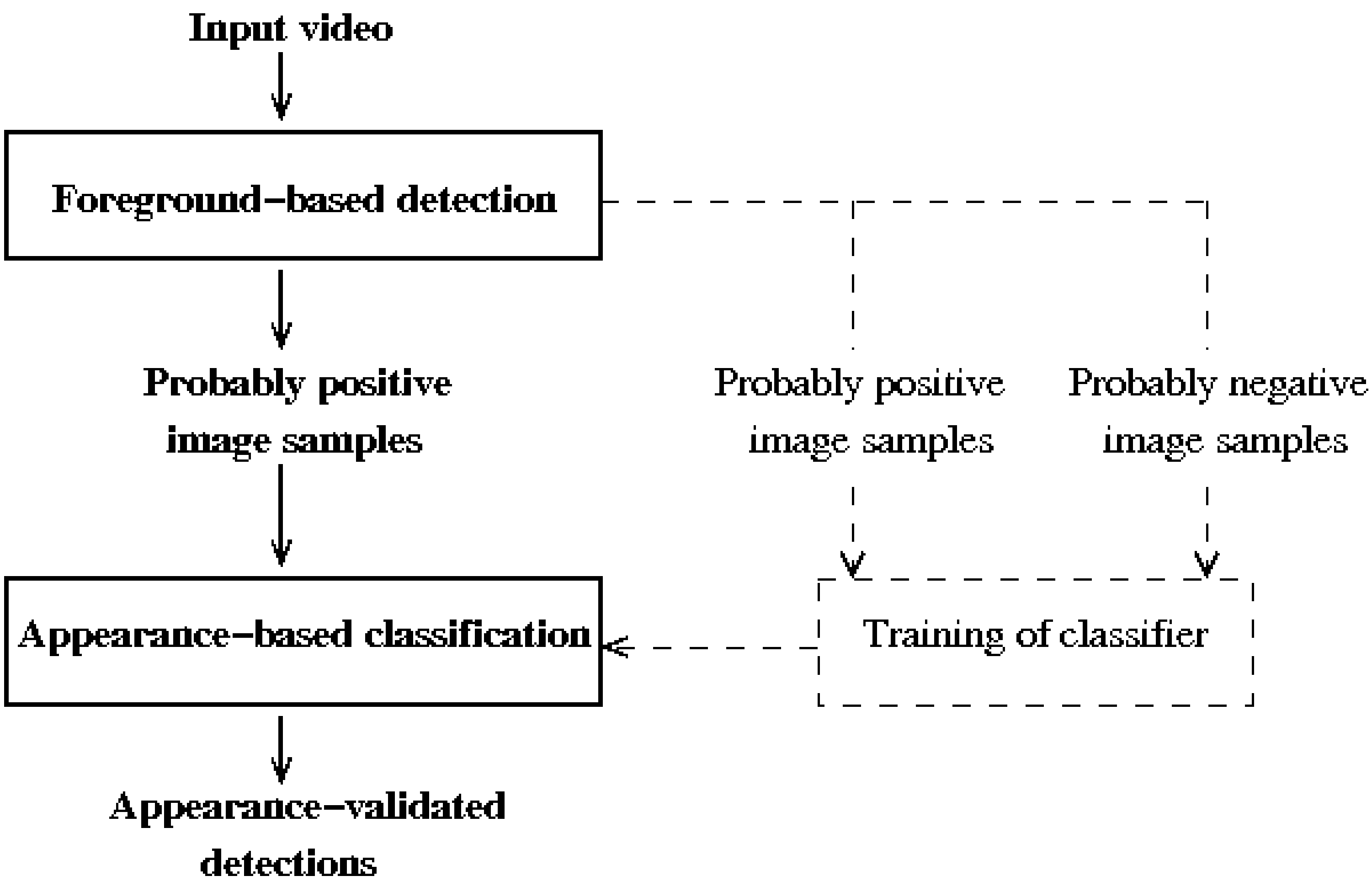
Single-view foreground-based detection

Co-training

- Goal: decrease the number of manually labeled samples (semi-supervised approach)
- Inputs:
  - 2 sets of samples:
    - A small one with labeled samples
    - A bigger one with unlabeled samples
  - 2 complementary classifiers based on independent features
- Train both classifiers based on the labeled samples (supervised approach)
- Reinforce the classifiers based on the unlabeled samples (unsupervised approach)

Blum, A., Mitchell, T. Combining labeled and unlabeled data with co-training.
COLT: Proceedings of the Workshop on Computational Learning Theory, Morgan Kaufmann, 1998, p. 92-100.

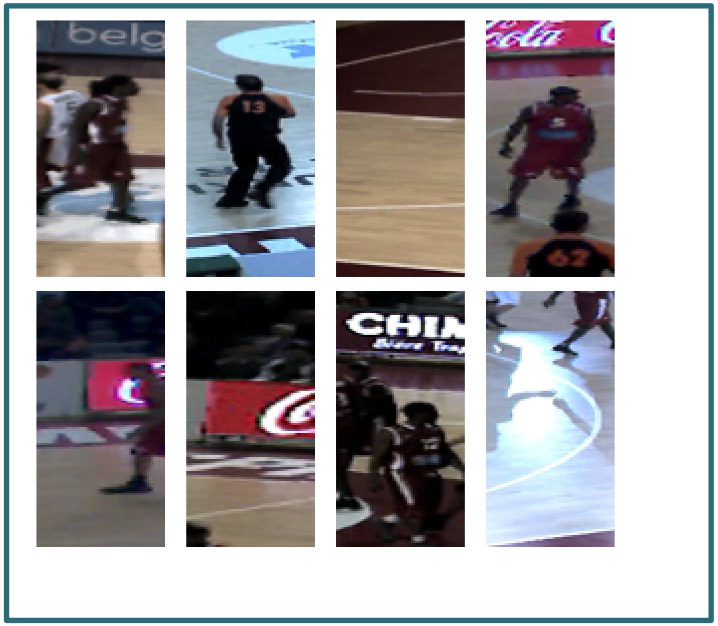
Framework


Input video
Foreground−based detection
Probably positive
image samples
Probably positive
image samples
Probably negative
image samples
Appearance−based classification
Training of classifier
Appearance−validated
detections

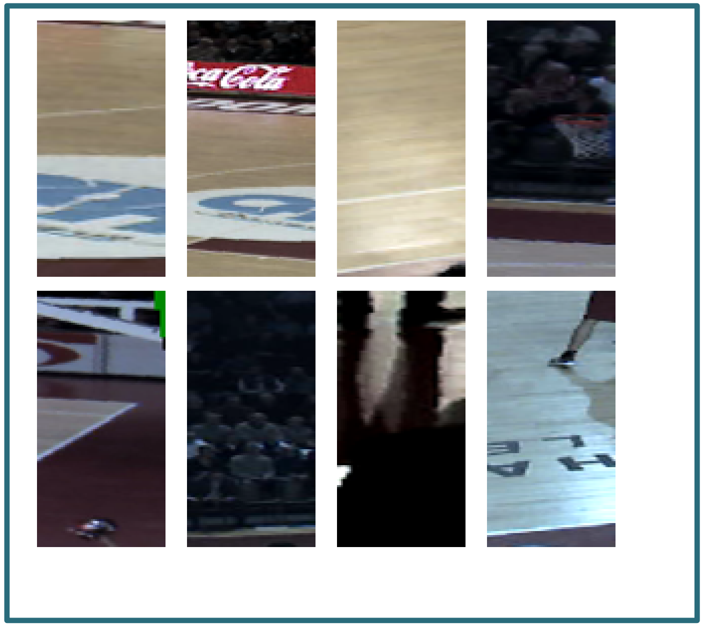
Dataset

Image samples in detector output set (corrupted with negative samples)

Image samples in negative set (randomly extracted from video)



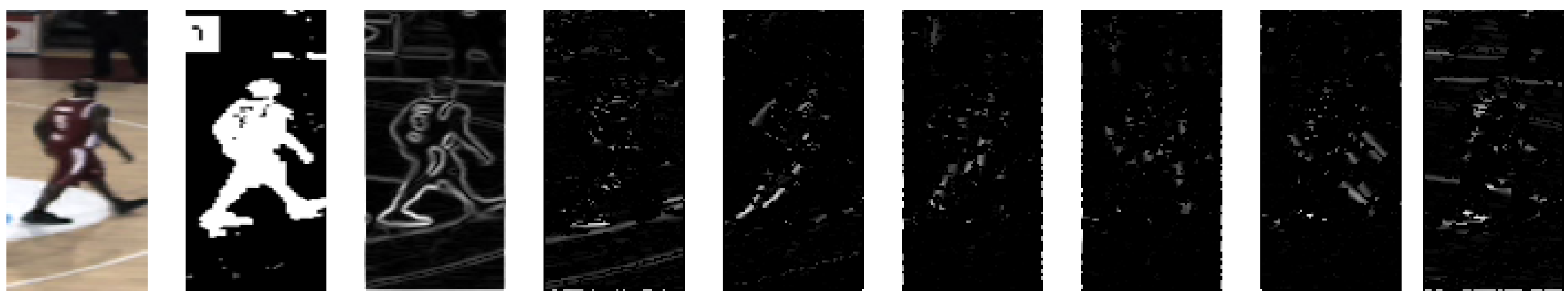
Detector output set
(Players + Referees + Negative samples)



Random window set

Visual features

- Channels:

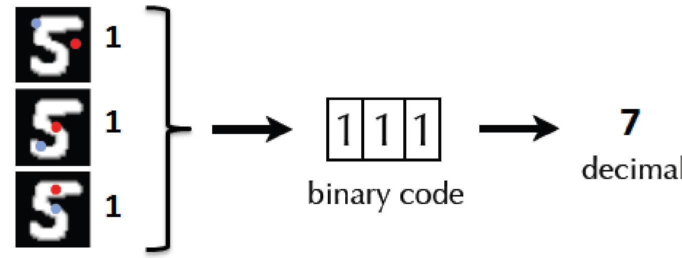


- Raw pixel values, or integral images

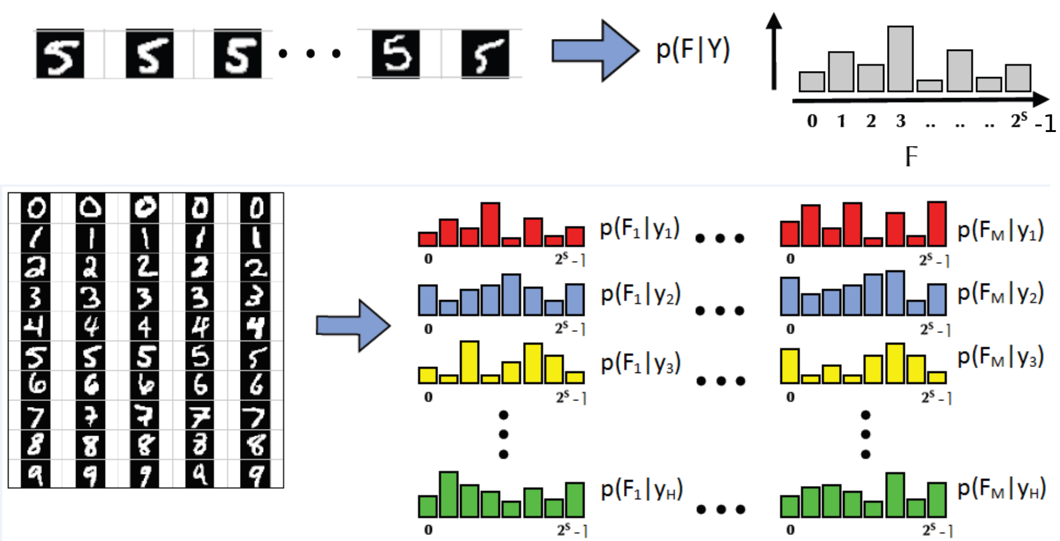
Classifiers: Random Ferns

Ensemble of binary tests:

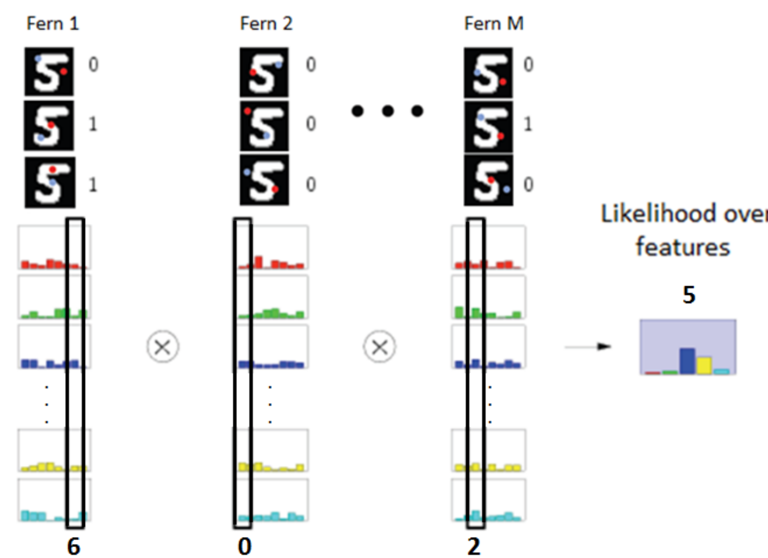
$$b_i = \begin{cases} 1 & if\ I(\bullet) < I(\bullet) \\ 0 & else \end{cases}$$



Training

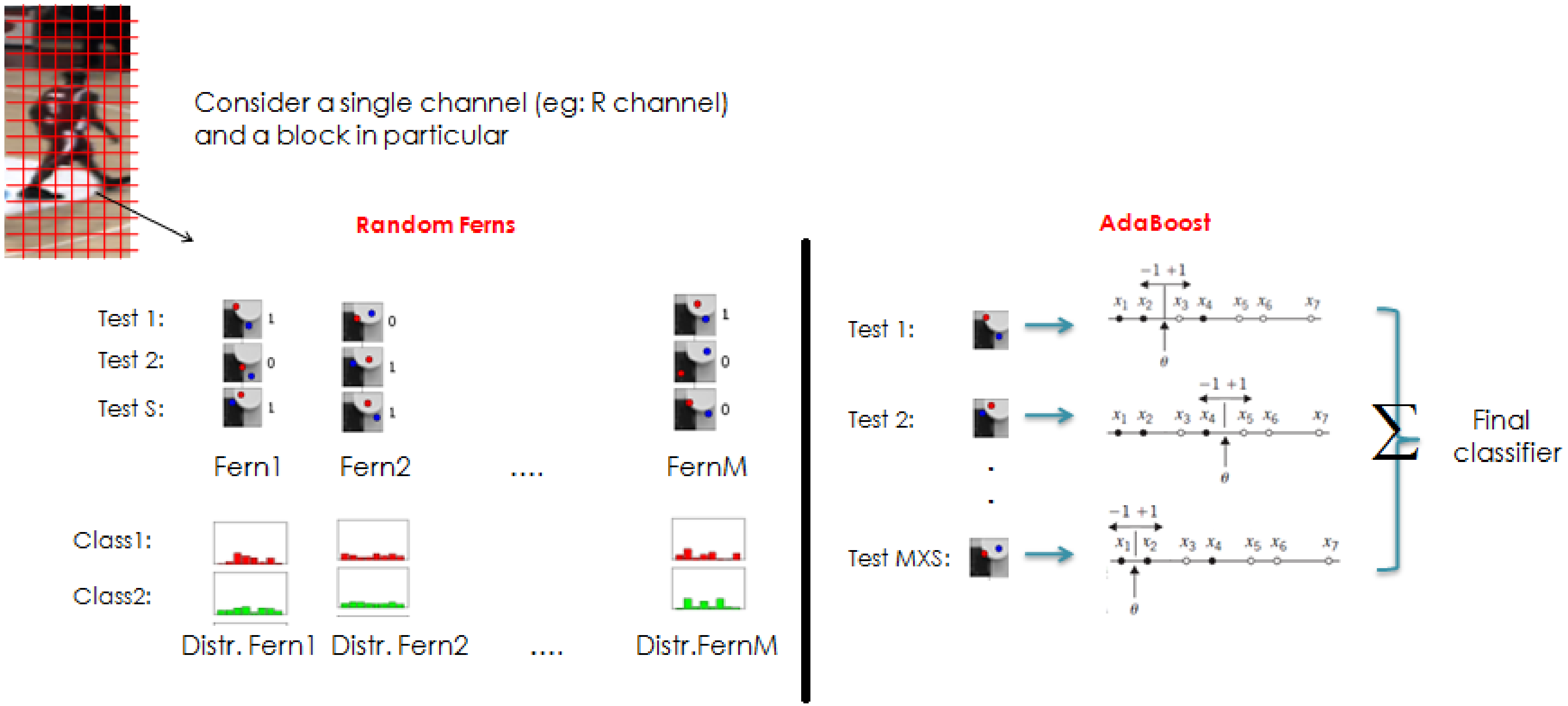


Test



Classifiers

- Ensemble of binary tests: $b_i = \begin{cases} 1 & if\ I(\color{red}{\bullet}) < I(\color{blue}{\bullet}) \\ 0 & else \end{cases}$

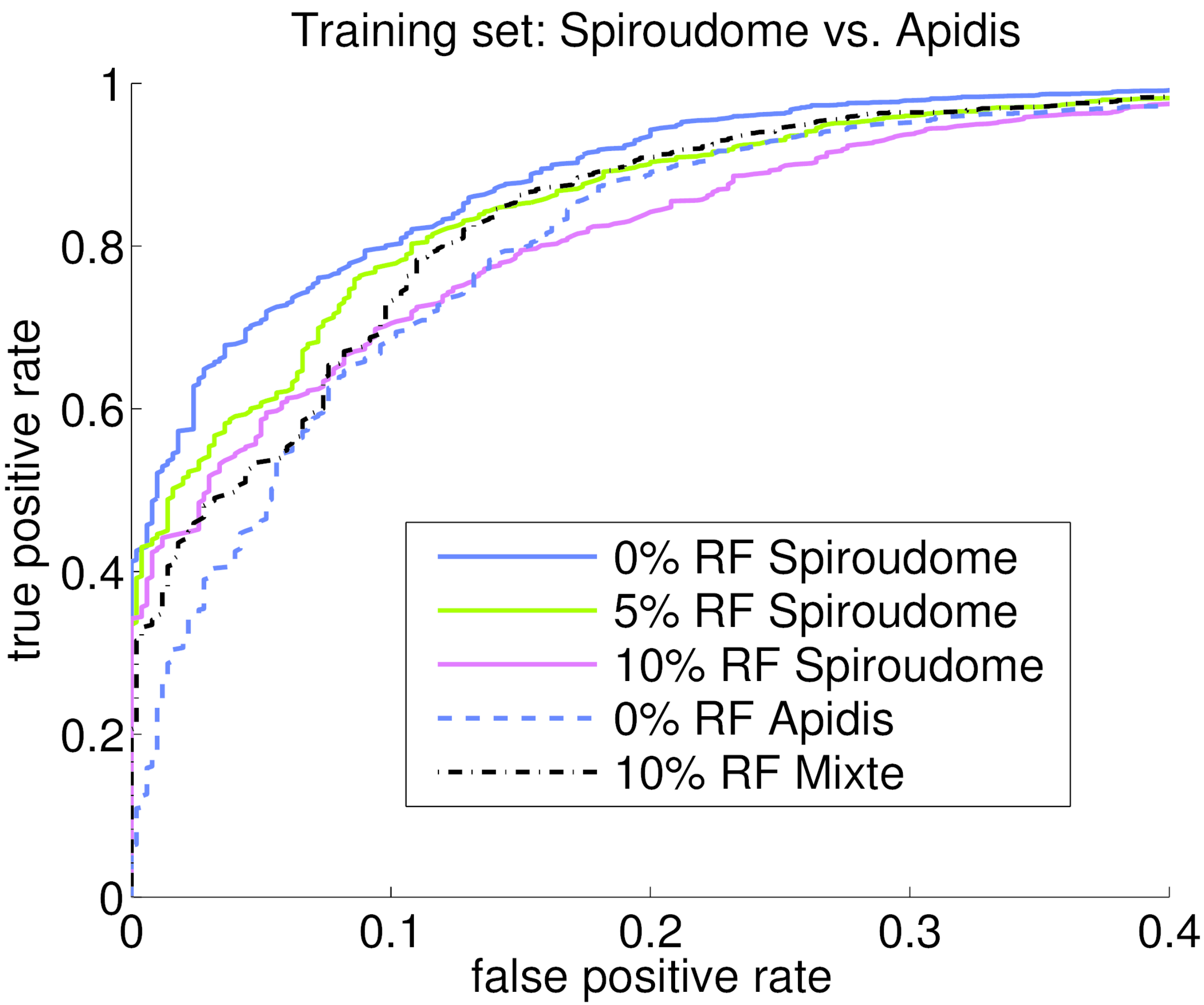


Results

- Data
  - Corrupted data with n% of corruption
    (n=0 => no corruption)
    - Learning:
      - Positive learning set: 1000 samples (n = 5 => 950 positives, 50 negatives)
      - Negative learning set: 1000 samples (probably negative)
    - Testing:
      - Testing set: 1000 samples with 10% of corrupted labeled
      - Test based on 5 learning sets
      - 5 tests by fern, 200 ferns by block, 16x16 blocks by image
- Methods
  - AdaBoost vs. Random Ferns
  - Integral image vs. pixel comparisons

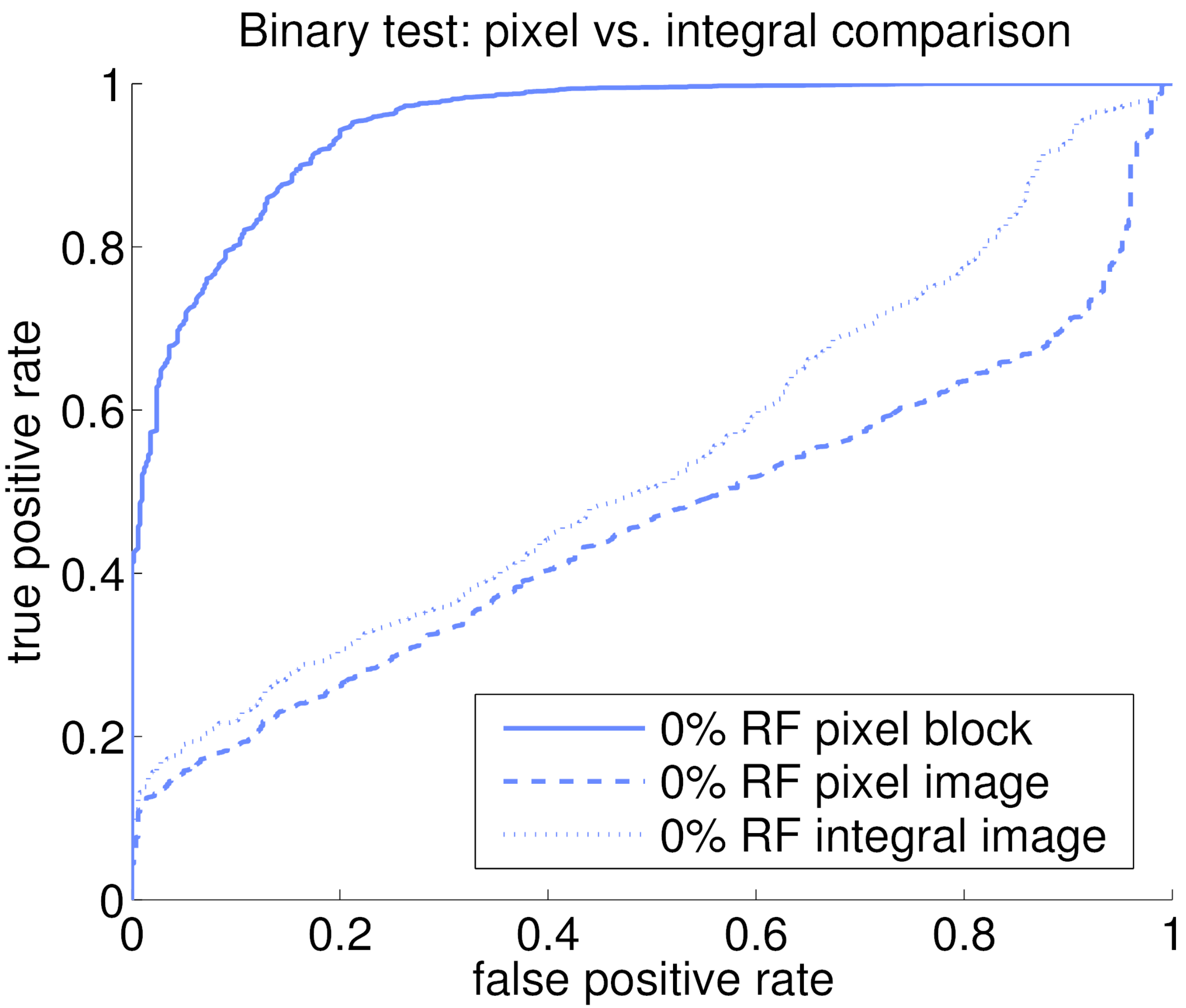
Single-view foreground-based detection

Is-it necessary to train on-line ?



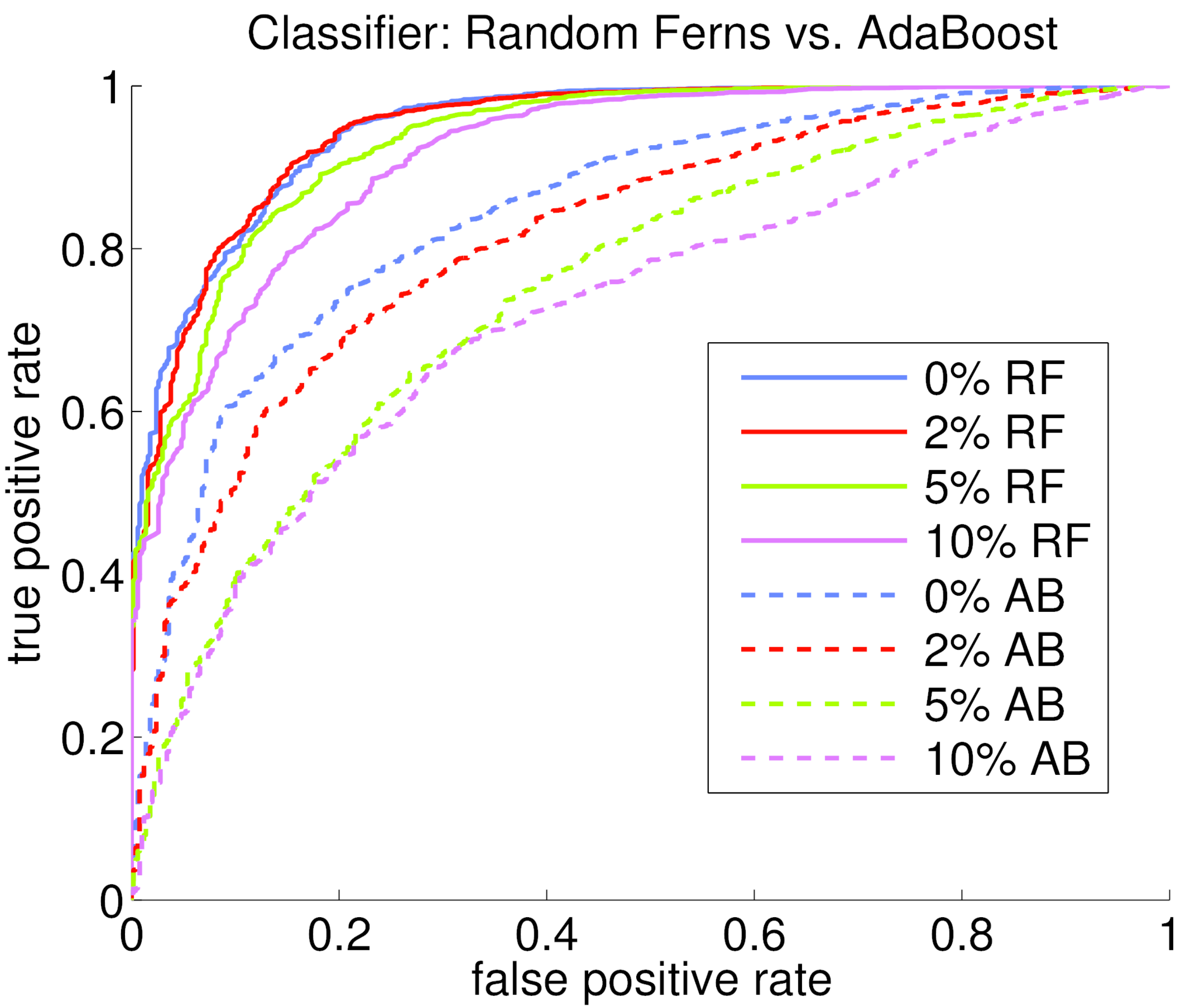
On-line training helps even in case of corrupted labels.

Which kind of binary test ?



Binary test based on the comparison of pixels within a block outperforms the two other kinds of binary test.

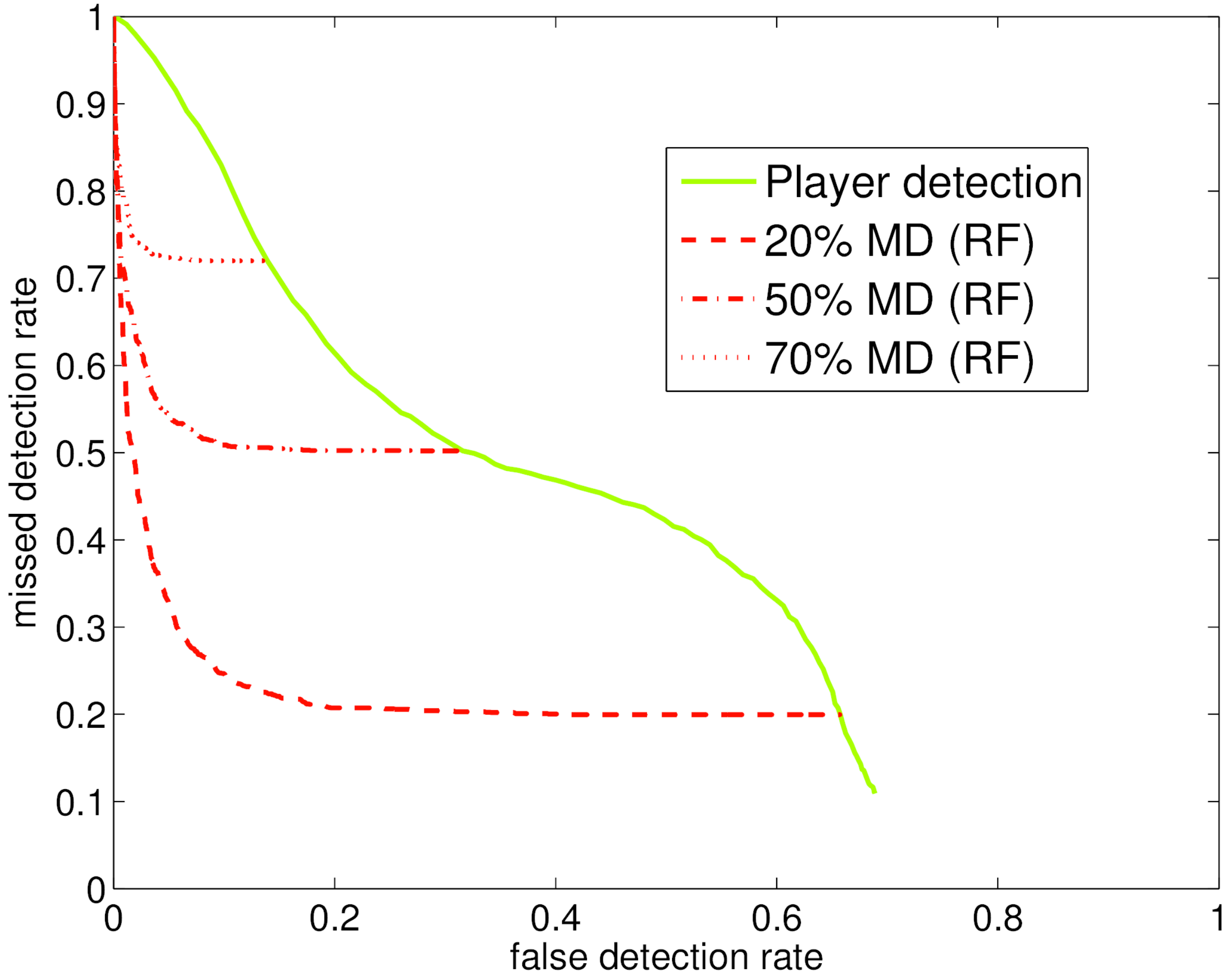
Which classifier with corrupted labels ?



Random ferns classifier outperforms AdaBoost ones and is less sensitive to corrupted labels.

Improvement of the detector

280 regularly spaced frames with ground truth



The classifier significantly improves the operating trades-off compared to the ones obtained based on foreground detection only.

Conclusion

- Co-training
- Random Ferns outperform AdaBoost.
- On-line training is necessary.
- The classifier improves the player detection algorithm.

Thank you for your attention !