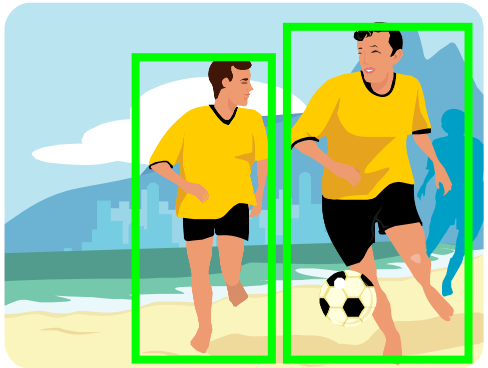
A Boosting Approach for Human Detection from Images

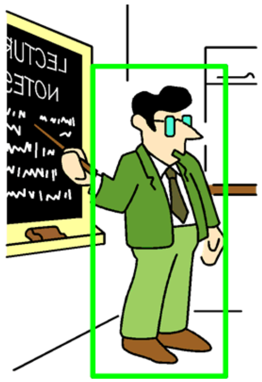
Yu-Ting Chen and Chu-Song Chen

Presented by Chu-Song Chen 陳祝嵩
Research Center for Information Technology Innovation
Academia Sinica
中央研究院 資訊科技創新研究中心

Introduction

- Goal:
  - Detect and localize upright people in static images.





- Applications of human detection in images
  - Video surveillance
  - Content-based image/video retrieval
  - Intelligent transportation systems

Introduction (cont.)

- Challenges of human detection
  - Wide variety of articulated poses
  - Variable appearance/clothing
  - Complex backgrounds
  - Unconstrained illumination
  - Occlusions, different scales

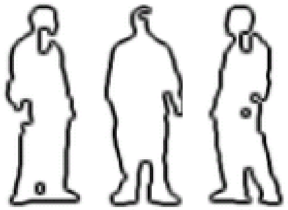
Basic Knowledge for Human Detection

- Three components are essential for learning a human detector.
  - Collecting training data

Positive training data

Negative training data

  - Extracting features from training data
    - color, edge, gradient, silhouette, etc.



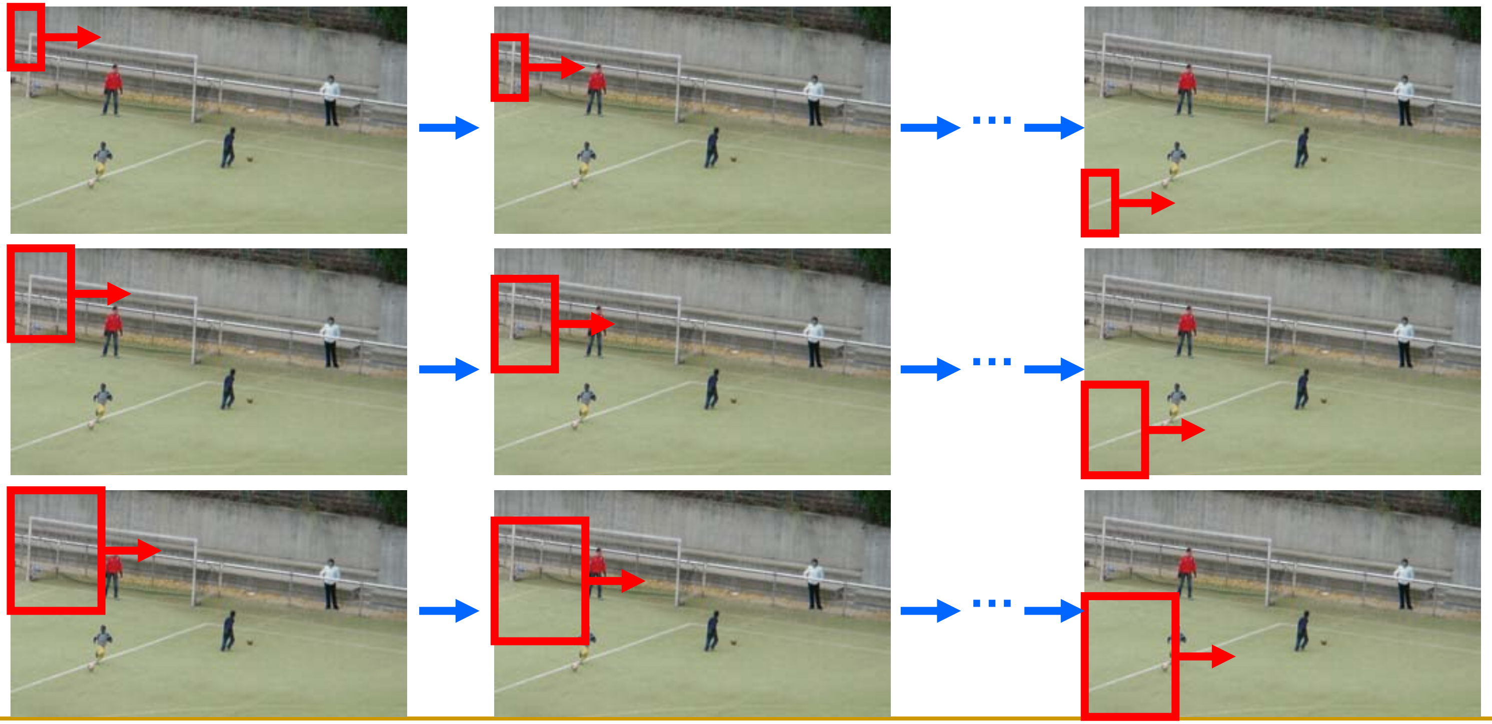
  - Learning an object detector
    - SVM, Boosting, Cascaded AdaBoost, Manifold learning, etc.

Basic Knowledge for Human Detection (cont.)

- After learning a human detector
  - A detection window is usually used to scan the testing image along *x* and *y* directions for human detection.

Basic Knowledge for Human Detection (cont.)

- Human detection
  - Detection windows with different sizes are used to detect humans with different scales.



Related Works

- Rectangle features + Boosted Cascade (Viola & Jones, CVPR'01)
  - Fast face detection (CVPR'01) and pedestrian detection (ICCV'03)
  - Feature: Intensity-based rectangle features
  - Drawback:
    - Rectangle feature is not sufficient for encoding the variety of human appearance.
- Histograms of Oriented Gradients (HOG) feature + SVM (Dalal and Triggs, CVPR'05)
  - Feature: Gradient-based HOG features
  - HOG captures local and global gradient-orientation structures of human images.
  - Drawback:
    - High-dimensional feature vector $\Rightarrow$ High computation cost

Overview

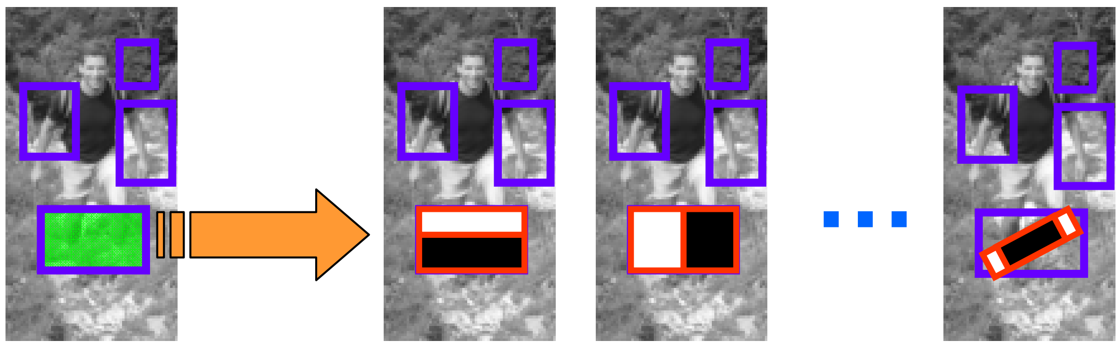
- What kinds of features are appropriate for human detection?
  - Intensity-based features
  - Gradient-based features
  - Hybrids of them
- In this research
  - Both intensity- and gradient-based features are employed:
    - Intensity-based features:
      - Rectangle features
    - Gradient-based features: Instead of using high-dimensional HOG features, 1-D gradient-based features are adopted.
      - EOH (Edge-Orientation Histogram) features
      - ED (Edge Density) features
  - A new boosted cascading structure is employed for learning a human detector.

Feature Pool (cont.)

- Intensity-Based Rectangle Features
  - Ten types of rectangle features are used.

Non-tilted rectangle features Tilted rectangle features

  - Inside a detection window, sub-blocks with different positions and aspect ratios can be produced.
  - Each sub-block B_i contains ten rectangle features.

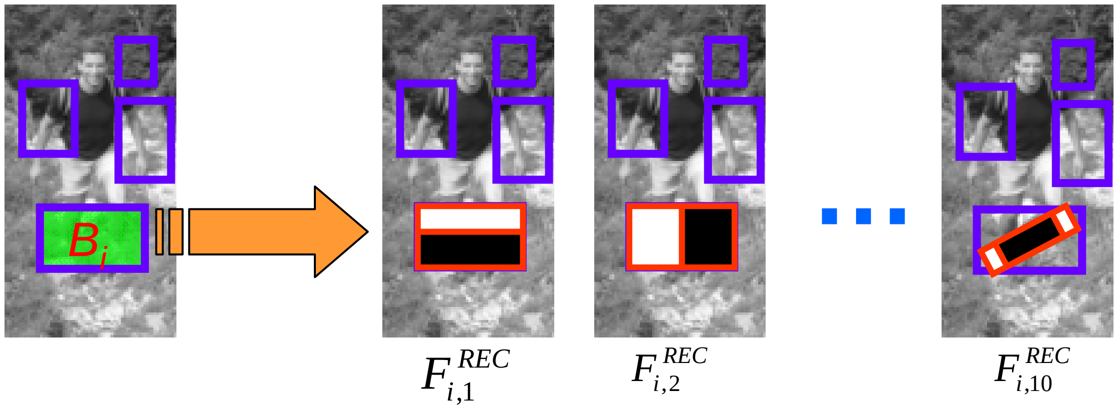


Feature Pool (cont.)

- Intensity-Based Rectangle Features (cont.)
  - A rectangle feature $F_{i,m}^{REC}$ is defined as

    $$F_{i,m}^{REC} = White(B_i, type_m) - Black(B_i, type_m)$$

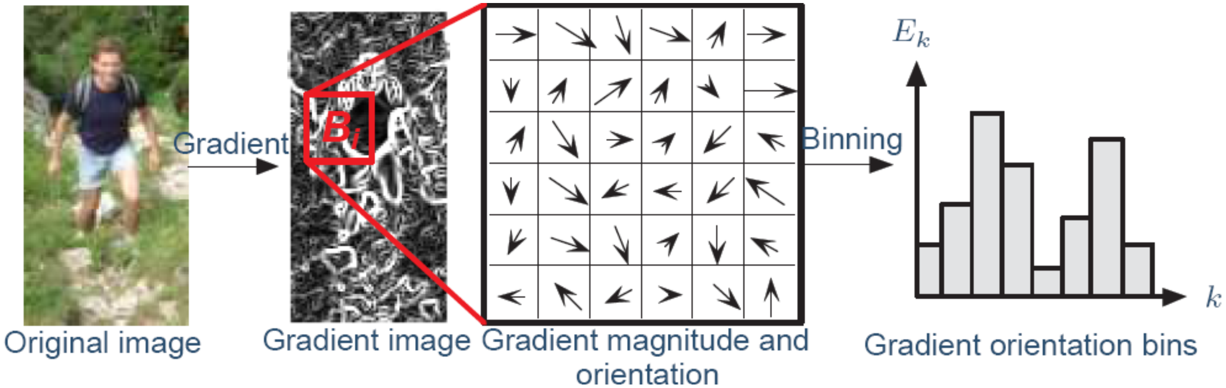
    - $type_m$ denotes m-th type rectangle feature
    - $White(B_i, type_m)$ and $Black(B_i, type_m)$ are the illumination summations in white and black regions.



  - The rectangle feature is represented by a real value.

Feature Pool (cont.)

- Gradient-Based EOH Features (Edge Orientation Histogram; Levi and Weiss, CVPR'04)
  - Similarly, there are several sub-blocks inside a detection window.
  - For a sub-block B_i, the pixel gradient magnitude m and gradient orientation θ are calculated.
  - Then, gradient orientation histogram E (with K bins over $0^o \sim 180^o$) is measured.



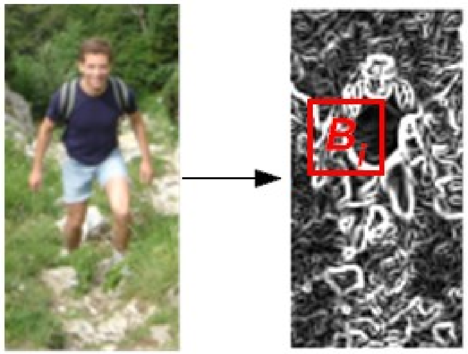
  - The EOH features $F_{i,k}^{EOH}$ is defined as: $F_{i,k}^{EOH} = \frac{E_k}{\sum_{j=1}^{K} E_j}$

Feature Pool (cont.)

- Gradient-Based EOH Features (cont.)
  - Properties of EOH features:
    - EOH feature can only characterize one gradient orientation at a time.
    - EOH feature is represented by a real-value.

Feature Pool (cont.)

- Gradient-Based ED Features (Edge Density; Phung and Bouzerdoum, ICASSP'07)
  - Similarly, there are several sub-blocks inside a detection window.
  - For a sub-block B_i, the ED feature F_i^{ED} is defined as the averaged gradient magnitude in B_i.



$$F_i^{ED} = \frac{1}{area_i} \times \sum_{(x,y)\in B_i} m(x,y)$$

where $m(x,y)$ is the gradient magnitude at location (x,y) and $area_i$ is the area of B_i.

  - ED feature is represented by a real value.

Feature Pool (cont.)

- In summary
  - The feature pool $\mathcal{F}$ employed in our approach contains both intensity- and gradient-based features:

$$\mathcal{F} = \{F_{i,m}^{REC}, F_{i,k}^{EOH}, F_{i}^{ED}\}$$

  - What features in $\mathcal{F}$ are discriminative for classifying positive and negative training data?
    - AdaBoost is useful for selecting discriminative weak classifiers (constructed by the features in $\mathcal{F}$).
    - Besides, all features in $\mathcal{F}$ are represented by a real value.
      - These features are suitable for integration into the AdaBoost algorithm.

The Idea of AdaBoost

- *"Find many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule,"* *~Schapire 2002*
  - 三個臭皮匠勝過一個諸葛亮
  - Boosting is a powerful technique for combining multiple "base" classifiers to form a committee whose performance can be significantly better than any of the base classifiers.
  - AdaBoost can give good results even if base classifier performance is only slightly better than random (>50%).
    - Hence base classifiers are known as weak learners.

The Idea of AdaBoost (cont.)

- Weak classifiers are trained and selected in sequence.
- Each weak classifier is trained based on weighted training data.
  - Weights of training data depend on the performance of the selected weak classifiers.
    - Training data misclassified by one of the classifiers is given more weight when used to train next classifier.
  - The weak learner with lowest weighted error over all weak learners is selected.
- Decisions are combined using a weighted majority weighting scheme and the final classification is given by

$$H = \sum_{t=1}^{T} a_t h_t$$

Features as Weak Classifiers

- In this research, features in the feature pool $\mathcal{F}$ are used to construct weak classifier *h*.

- After selecting *T* weak classifiers, the strong classifier can be expressed as:

$$H(z) = sign(\sum_{t=1}^{T} a_t h_t(z) - \alpha)$$

  where *z* is the input example and α is a threshold.

- The confidence value of *H* is defined as:

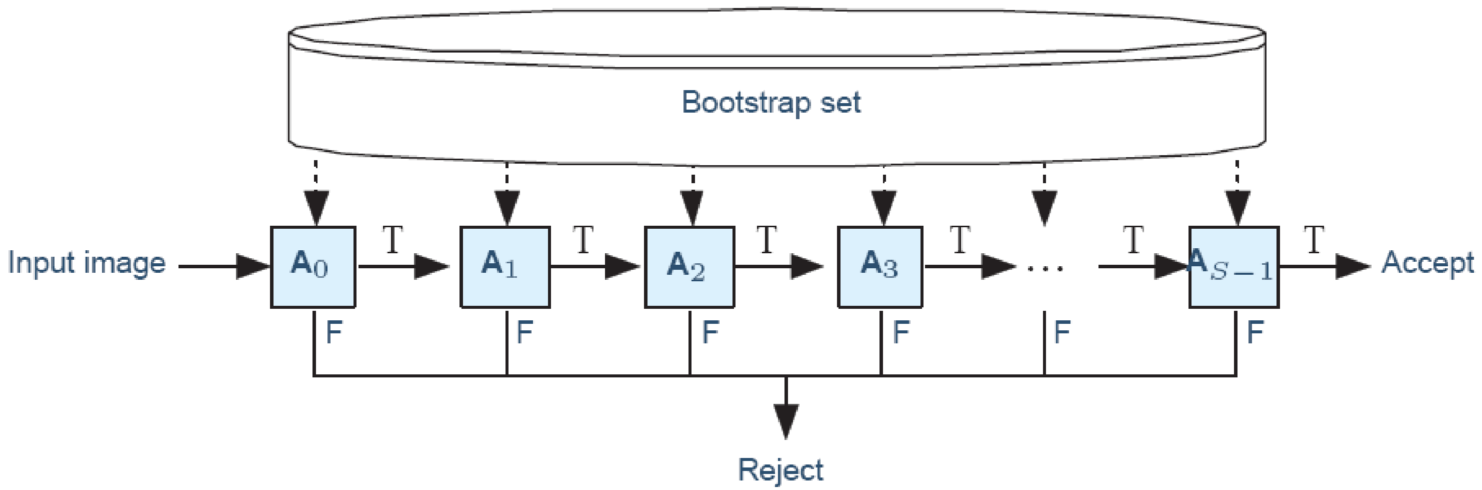
$$conf_A(H; z) = \sum_{t=1}^{T} a_t h_t(z) - \alpha$$

The Idea of Cascaded AdaBoost

- To find humans in an image usually involves a full search of all possible positions and scales in the image.
- Observe that the negative windows are far more than positive windows.
  - AdaBoost classifier uses same computation time to accept or reject a window.
  - Therefore, many computation power is wasted on rejecting negative windows.

The Idea of Cascaded AdaBoost (cont.)

- Viola and Jones (Viola & Jones, CVPR'01) proposed a boosted cascade to fast reject negative windows.
  - Basic cascaded structure (*S* stages)



where A_i is an AdaBoost classifier in the *i*-th stage.

  - Negative example can leave the cascade in some early stages.
  - Only detection windows that pass all the stages are deemed to be positive ones.

The Idea of Cascaded AdaBoost (cont.)

- Properties of cascaded AdaBoost:
  - The decision times for negative and positive windows are un-equal.
    - Negative examples < Positive examples
  - The negative windows are far more than positive windows to detect in the image.
    - Saving on the detection time of the negative windows increases the overall efficiency of the object detector.

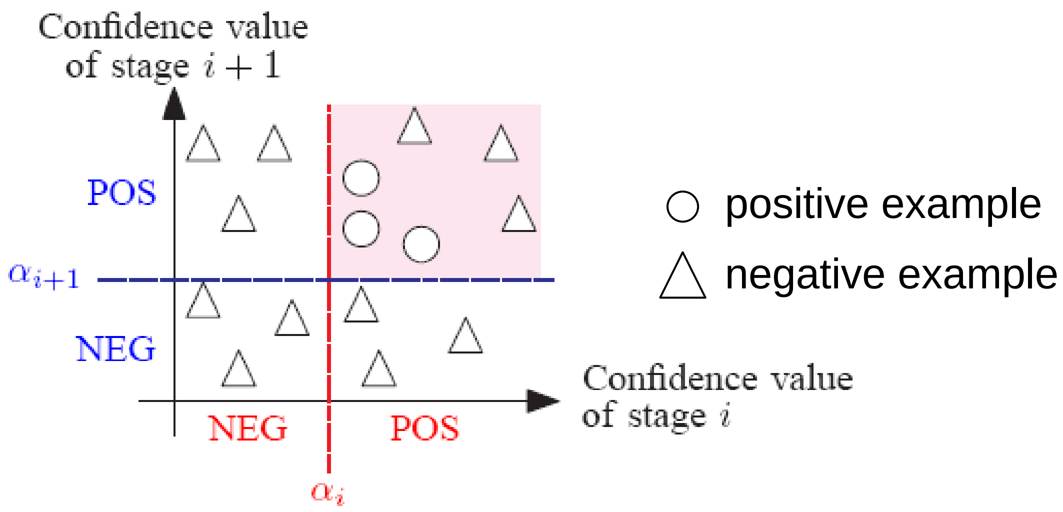
The Idea of Cascaded AdaBoost (cont.)

- To train a cascaded structure, two goals are set for each stage A_i.
  - d_i: Minimum detection rate of positive training examples.
    - Ex. $d_i = 99.95\%$
  - f_i: Maximum false-positive rate of negative training examples.
    - Ex. $f_i = 50\%$
- In each stage, the AdaBoost algorithm is employed to select a set of weak learners from $\mathcal{F}$ to achieve the goals, d_i and f_i.
- The overall detection rate D and the false-positive rate F for the training data can be estimated as:

$$
\begin{aligned}
D &= \prod_{i=1}^{S} d_i \\
F &= \prod_{i=1}^{S} f_i
\end{aligned}
$$

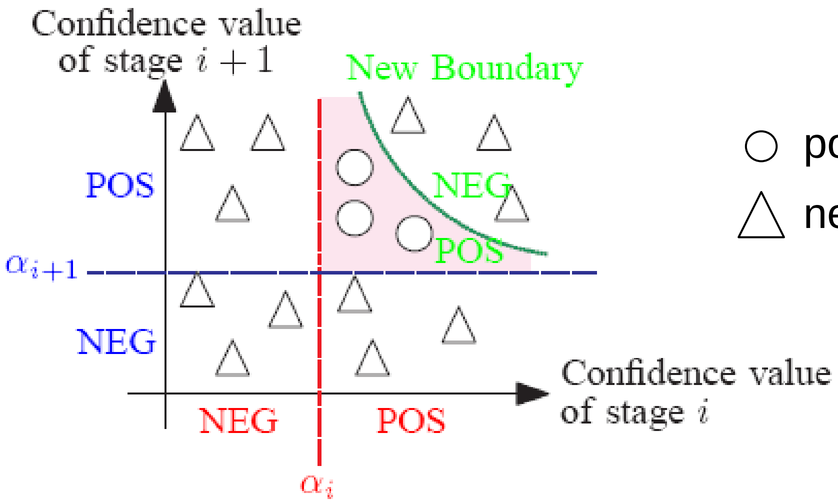
The Idea of Cascaded AdaBoost (cont.)

- In basic cascaded structure
  - The confidence value used to make a yes/no decision in current stage is not used in later stages.
  - The stages are independent to each other.
  - The classification boundaries are parallel to the axes of the stages.



Exploit Inter-Stage Information

- Exploit the inter-stage information
  - It is possible to enhance the classification performance.
    - composite the confidence values of multiple stages (say, *d*-stage) into a vector
    - make a decision in the *d*-dimensional space
    - The classification boundaries can be hyper-planes of general forms.



○ positive example

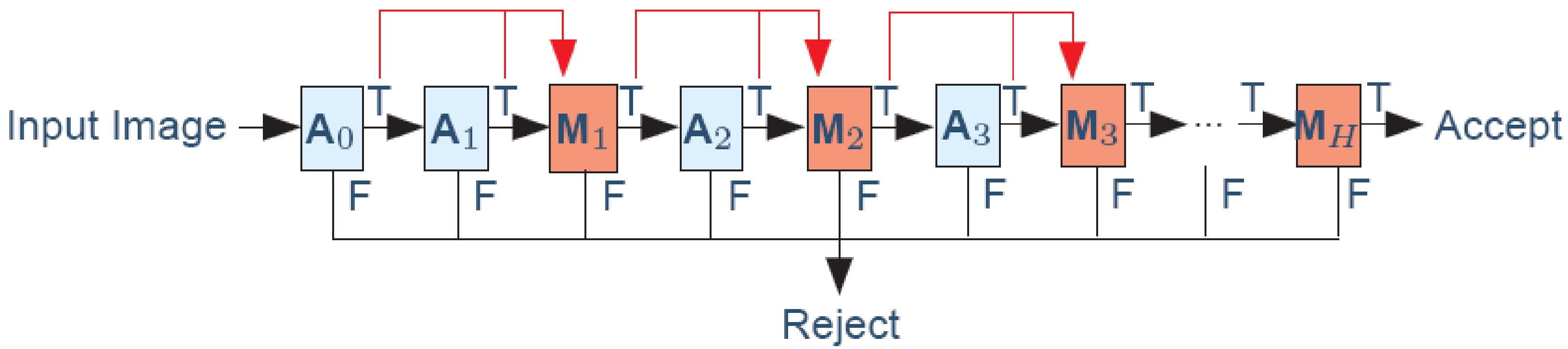
△ negative example

Exploit Inter-Stage Information (cont.)

- One possible way to exploit the inter-stage information
  - delay the decisions of all *S* stages
  - perform a post-classification
  - Drawback: Negative example has no chance to leave the cascade early.

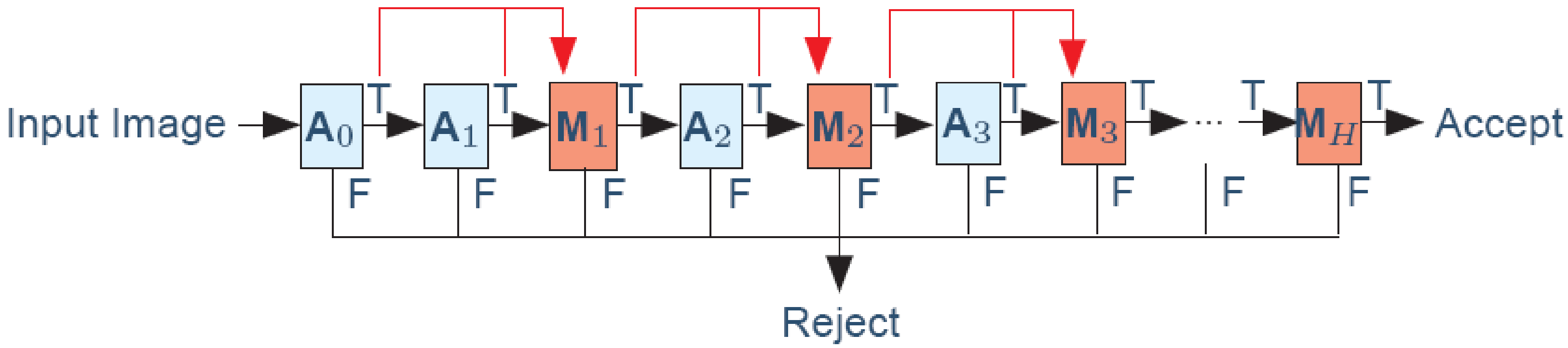
Adding Meta-Stages (IEEE TIP'08)

- Adding meta-stages to the basic cascaded structure



- Properties of meta-stages:
  - A meta-stage is a classifier that uses the inter-stage information (represented by the confidence value) of some previous stages for learning (as shown in red arrows).
  - The meta-stage also outputs a confidence value according to the adopted classification method.

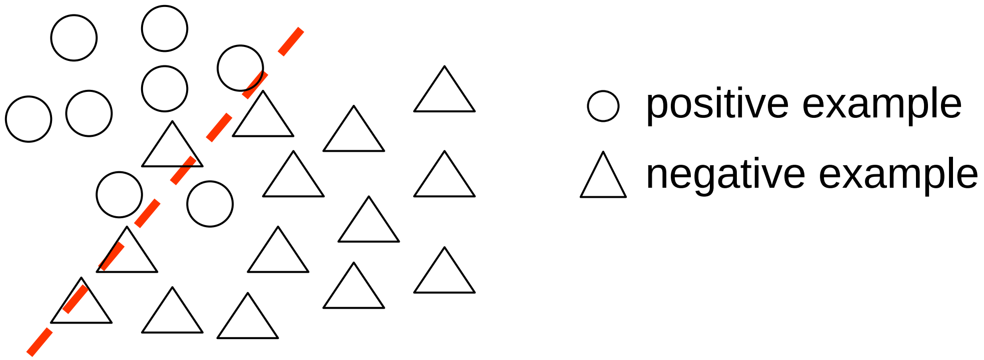
Adding Meta-Stages (cont.)



- Meta-stage of above structure
  - It is a light-weight computation.
  - It can further reject the negative examples.
  - It contains information of all previous stages.
  - In our implementation, the goals of meta-stage are usually set as:
    - Detection rate of positive training examples: d_i = 100%
    - Find the classifier with the lowest false-positive rate f_i of the negative training examples under d_i = 100%.

Meta-Stage Classifier

- The linear SVM (LSVM) is adopted as the meta-stage classifier because of its high generalization ability and efficiency in evaluation.
  - 3-fold cross-validation is applied to select the best penalty parameter *C* of the LSVM.
  - A maximum-margin hyper-plane that separate positive and negative training data can be learned based on *C*.

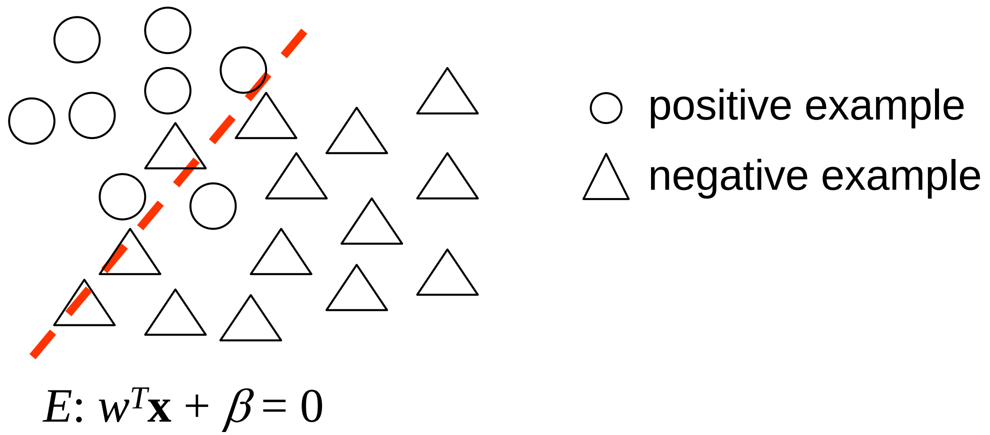


  - The hyper-plane is formulated as E: $w^T\mathbf{x} + \beta = 0$

    where w is the normal vector of the plane and β is the offset from the origin.

Meta-Stage Classifier (cont.)

- To satisfy the goals of the meta-stage:
  - move the hyper-plane along its normal direction (w) by applying different thresholds (β).
  - find a threshold with the lowest f_i (under the assumption that d_i = 100%).

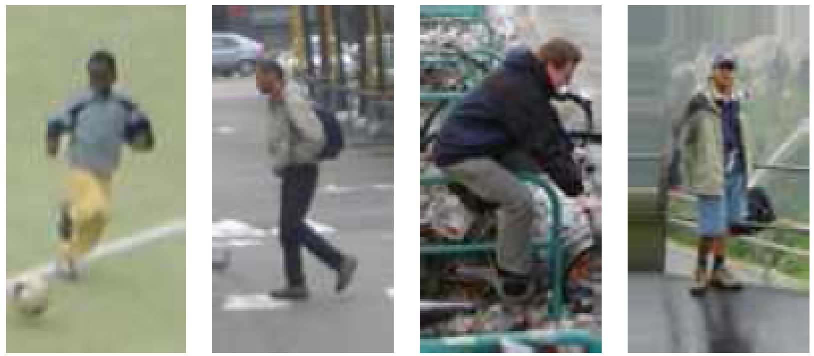


- The confidence value of the meta-stage for data **x'** is defined as:

$$conf_M(\mathbf{x}') = w^T\mathbf{x}' + \beta$$

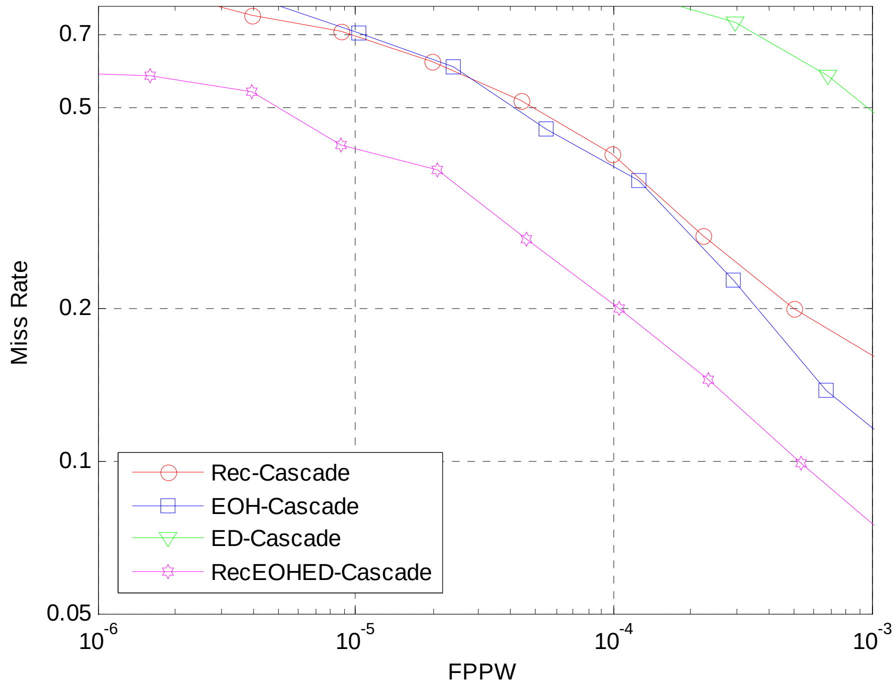
Experiment Results

- INRIA person data set is used.
  - This data set contains people standing in different positions with different orientations and poses.
  - Some training and testing human image of the INRIA data set are shown

Training	Testing
	

Experiment Results (cont.)

- What kinds of features are appropriate for human detection?
  - Intensity-based feature: Rec-Cascade (Rectangle Features)
  - Gradient-based feature: EOH-Cascade (EOH Features)
  - Gradient-based feature: ED-Cascade (ED Features)
  - Hybrid feature: RecEOHED-Cascade (Rectangle + EOH + ED Features)



Detection Efficiency:

Rec-Cascade	47.75 FPS
EOH-Cascade	6.67 FPS
ED-Cascade	20.78 FPS
RecEOHED-Cascade	5.99 FPS

Performance of Inserting Meta-Stages

- Because there are many alternatives in combining AdaBoost stages and meta-stages, we can not test all of them.
- In our experiments, we consider the following three forms:
  - 2D meta-stages (MetaCascade-2D):

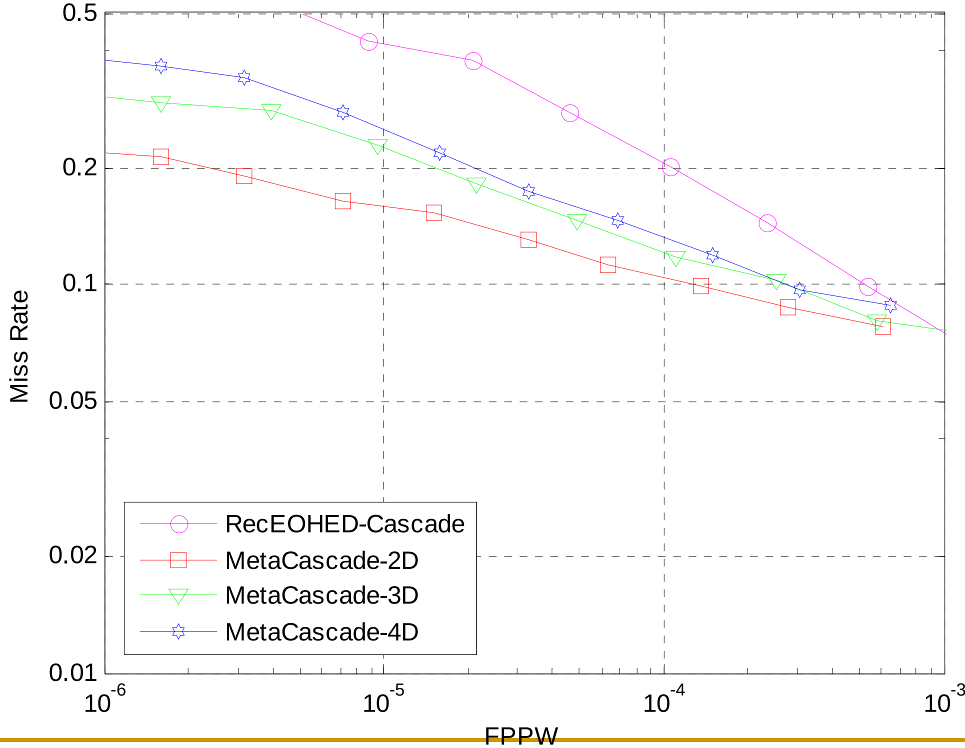
    AAMAMAM…AM
  - 3D meta-stages (MetaCascade-3D):

    AAAMAAMAAM…AAM
  - 4D meta-stages (MetaCascade-4D):

    AAAAMAAAMAAAM…AAAM

Performance of Inserting Meta-Stages (cont.)

- We evaluate how inserting several meta-stages affects the performance of the detector.

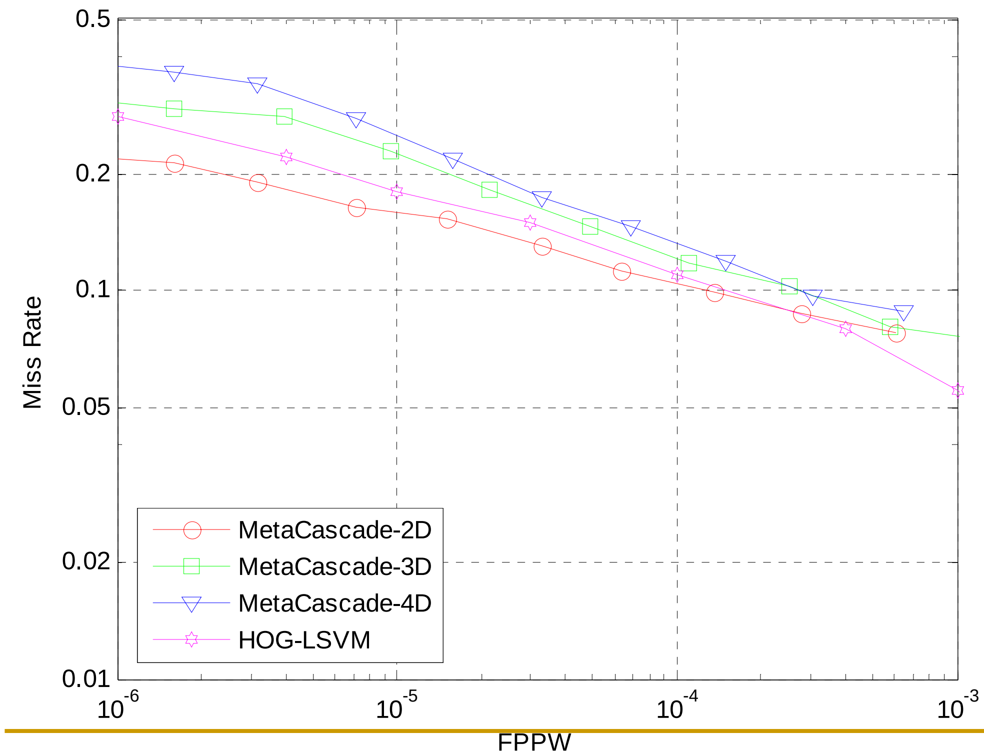


Detection Efficiency:

RecEOHED-Cascade	5.99 FPS
MetaCascade-2D	8.61 FPS
MetaCascade-3D	8.52 FPS
MetaCascade-4D	8.44 FPS

Efficiency and Accuracy Comparison with HOG

- We compare the proposed cascaded method with meta-stage to the HOG-based method (Dalal and Triggs, CVPR'05).



Detection Efficiency:

MetaCascade-2D	8.61 FPS
MetaCascade-3D	8.52 FPS
MetaCascade-4D	8.44 FPS
HOG-LSVM	0.91 FPS

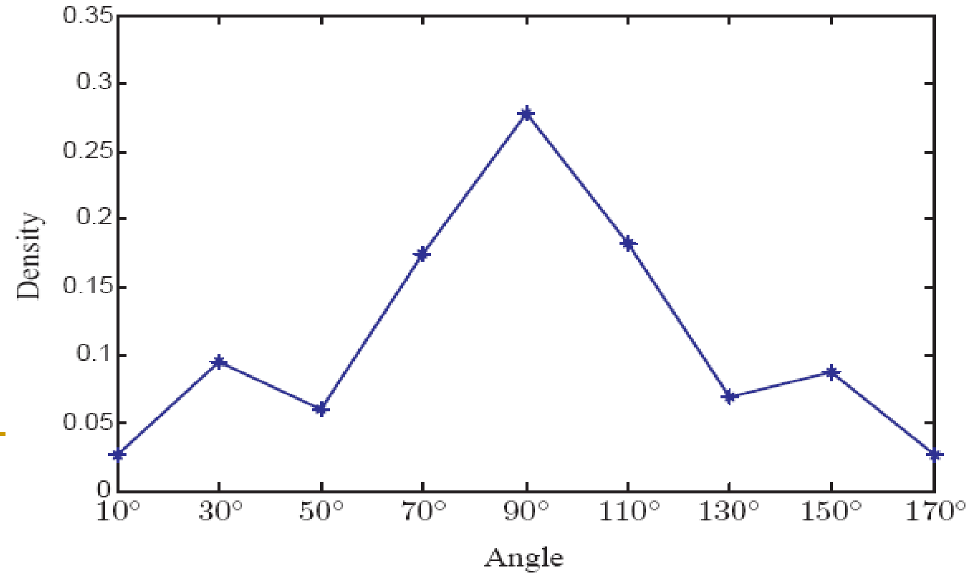
Visualization of Selected Features

- Four features selected in the first AdaBoost stage A_0 of the MetaCascade-2D method.
  - These four features are effective for fast rejecting non-human images.

EOH ED Rectangle Rectangle

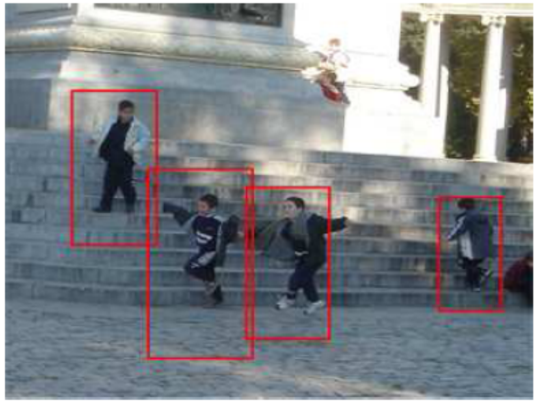
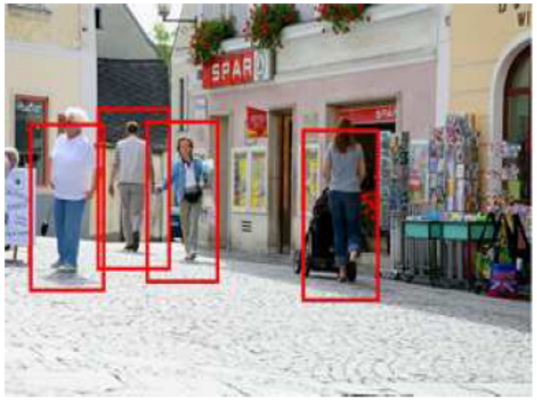
Relative Frequency of Selected EOH Features

- We show a relative frequency distribution of the angles that were selected for the EOH features in the MetaCascade-2D method.
  - The distribution peaks at 90^o (vertical angle).
    - This reveals that the vertical edge is a main characteristic for human detection.
  - Note that the curve is roughly symmetric.
    - It is because that standing humans are approximately symmetric about the vertical axis.



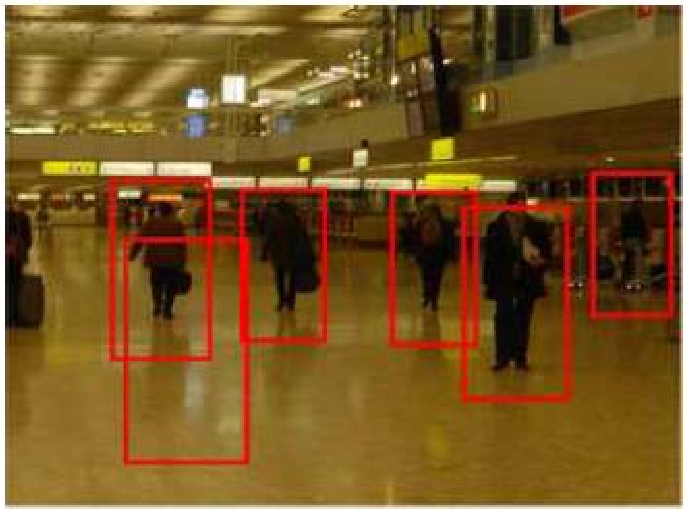
Detection Results

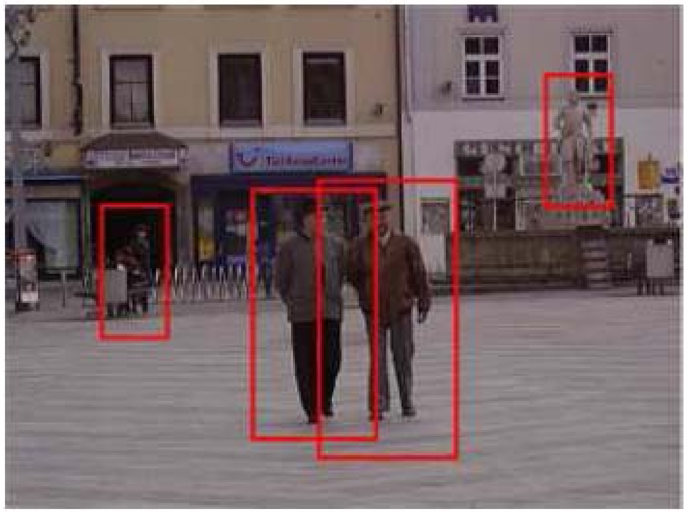
- This slide shows some detection results on INRIA person data set and Google images downloaded from the Internet.

INRIA person data set	Google Image Search
	
	

False-Positive Results

- We show two examples of false-alarms, where a human shadow and a statue are mis-detected.





Conclusions

- A combination of heterogeneous features can result in a more effective detector.
  - A better detection accuracy can be achieved by integrating both intensity- and gradient-based features.
- We proposed a novel cascaded structure that incorporates meta-stages.
  - The detection accuracy can be enhanced without sacrificing the detection efficiency.

Thank You!