Learning To Drive

Pedestrian Detection for Self Driving Cars

Simone Rossi & Matteo Romiti

January 14, 2017

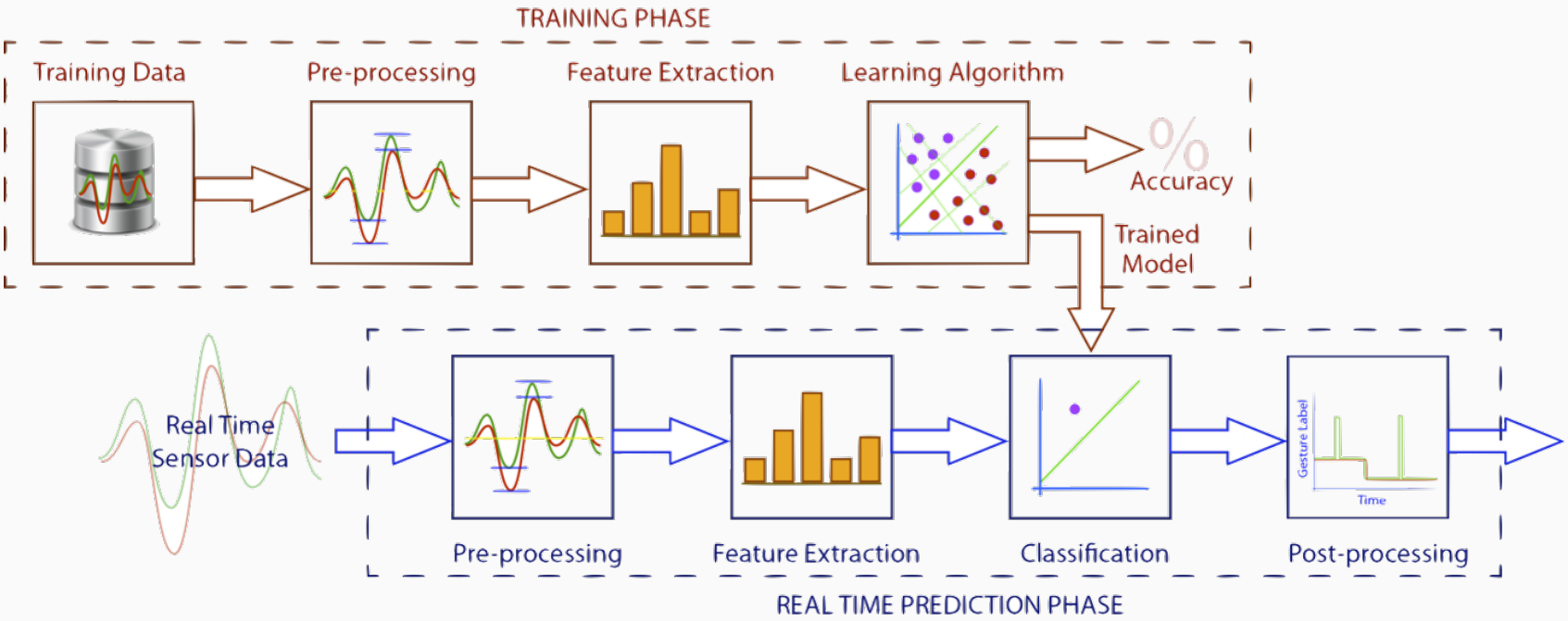
EURECOM, Ecole d'Ingénieur et Centre de Recherche en Telecommunications

Table of contents

Introduction


LEFT REARWARD VEHICLE CAMERA
MEDIUM RANGE VEHICLE CAMERA
RIGHT REARWARD VEHICLE CAMERA
ROAD FLOW
IN-PATH OBJECTS
ROAD LIGHTS
OBJECTS
ROAD SIGNS

Image Classification Pipeline



Classic pipeline for **Pedestrian Detection**

Histograms of Oriented Gradients

Histograms of Oriented Gradients

Why?

One of the most well known methods for feature extraction for human detection in computer vision, it is very versatile and it can be applied in different contexts. Ofter it is used as metrix to compare more sophisticated methods.

Histograms of Oriented Gradients

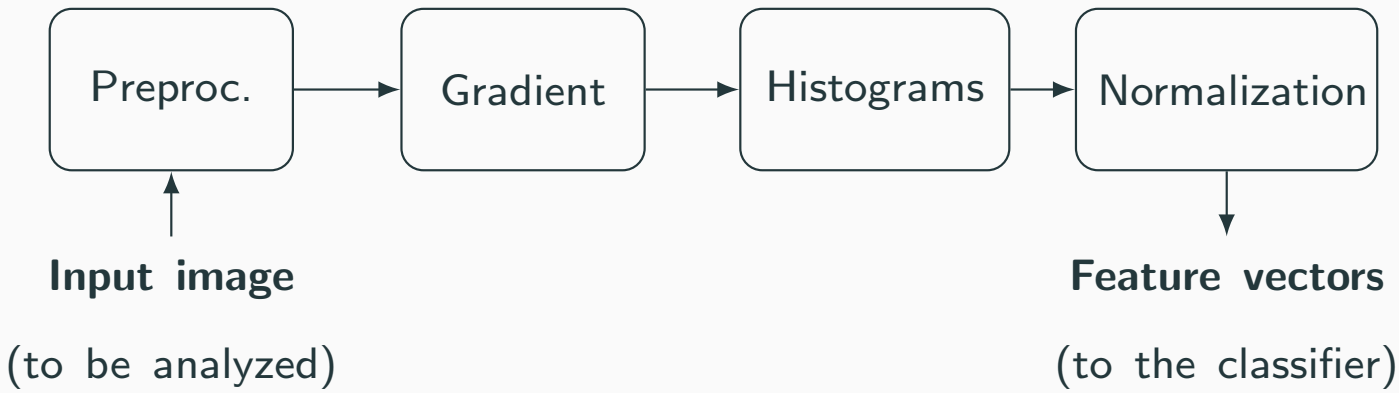
Idea?

Describe the local behaviour of the gradient applied to an image, in ordet to emphasize well defined geometric structures and edges. A local object appearance and shape can be characterized by the distribution of local intensity gradients or edge directions, even without knowledge of the corresponding gradient.

Histograms of Oriented Gradients

How?

Divide the image window into small spatial regions (*cells*), accumulate a 1D *histogram* of gradient directions over the pixels of the cell and normalize the result over a *block* of cells for better invariance to illumination.



Histograms

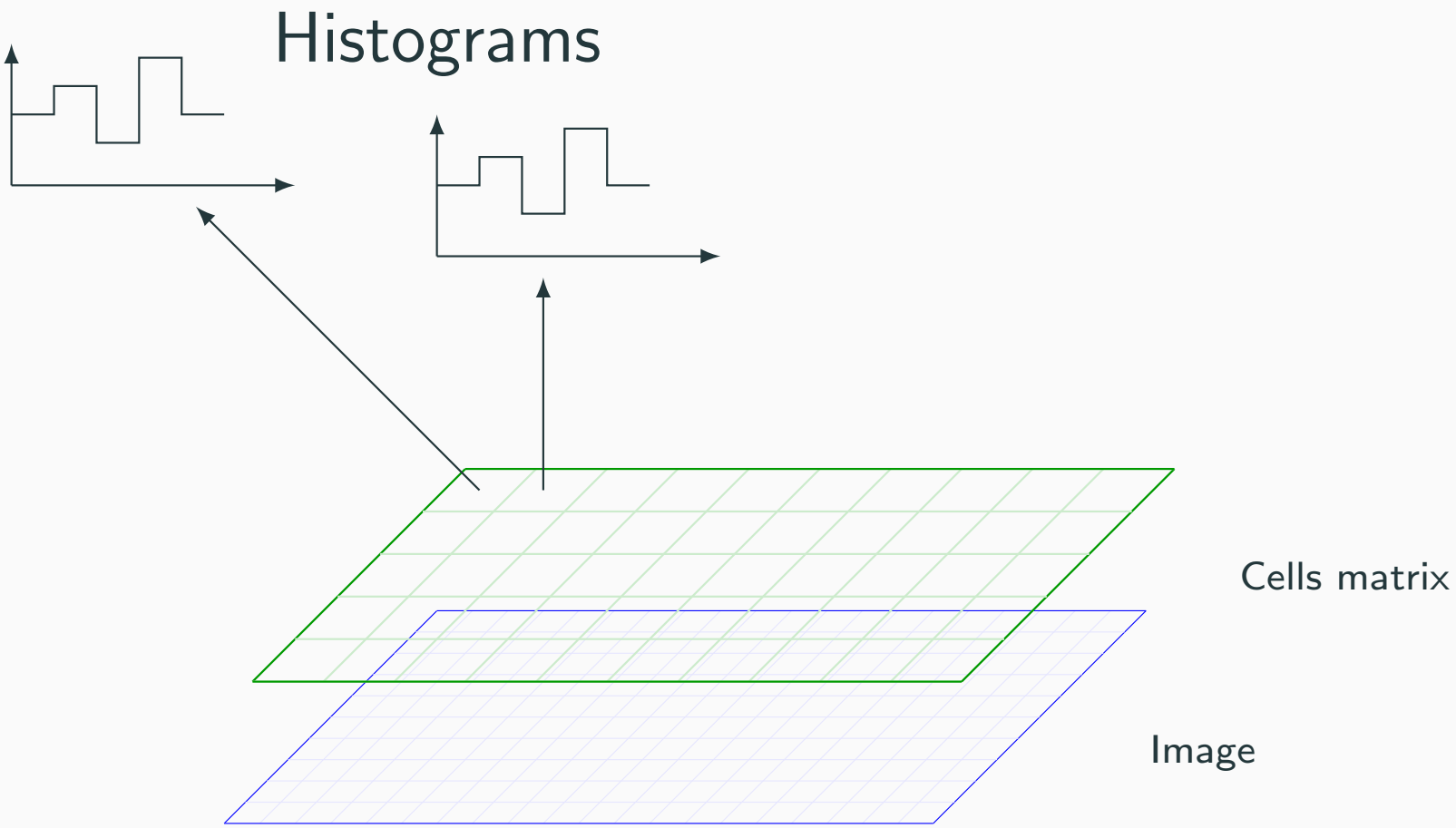


Figure 1: Spatial distribution of histograms and cells

Histograms

All the histograms are built as follows ($k = 1, \ldots, n_\theta$):

$$H_c(k) = \sum_i \sum_j f[G(i,j)]\delta(\phi_{k-1} < \theta(i,j) < \phi_k) \tag{1}$$

- n_θ *angle bins* uniformely distributed
- The vote is a function of the gradient magnitude at the pixel $f(G)$, either the magnitude itself, its square, its square root, ...

Normalization

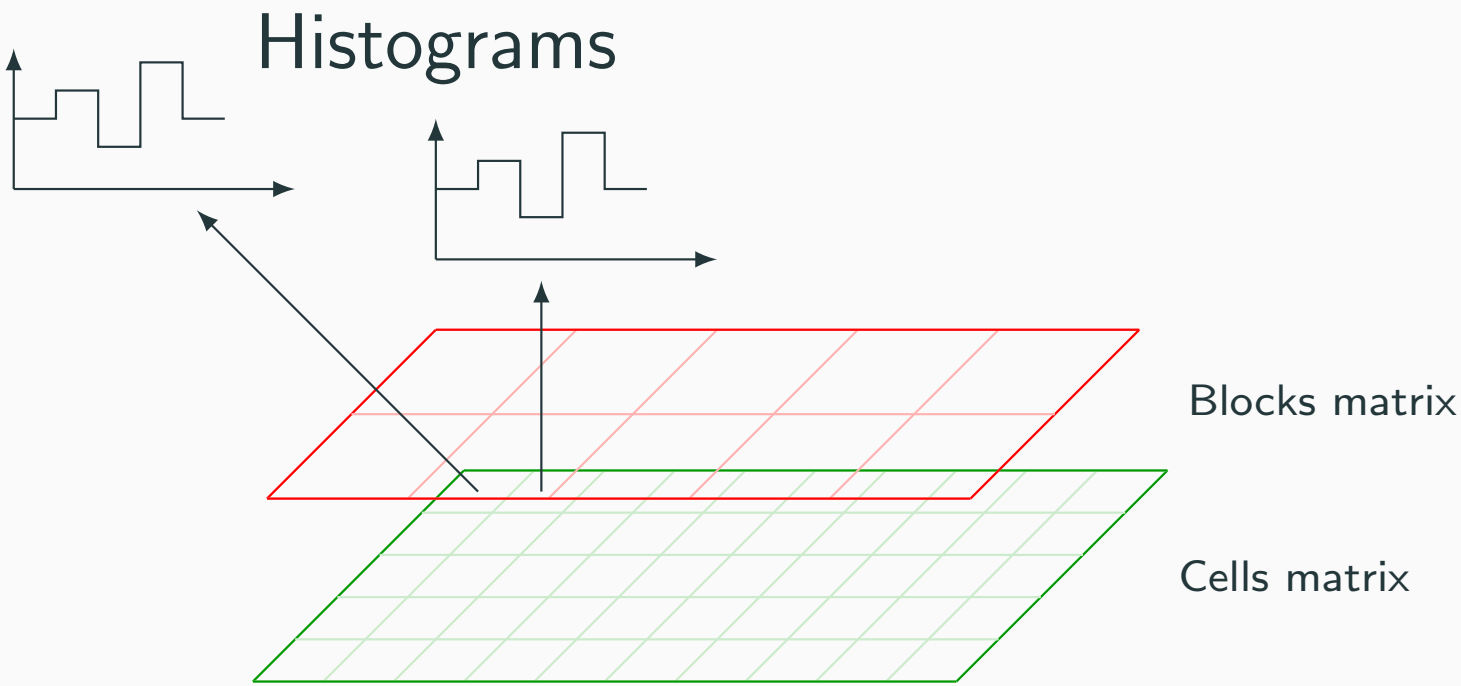


Figure 1: Spatial distribution of histograms, cells and blocks

Normalization

Why?

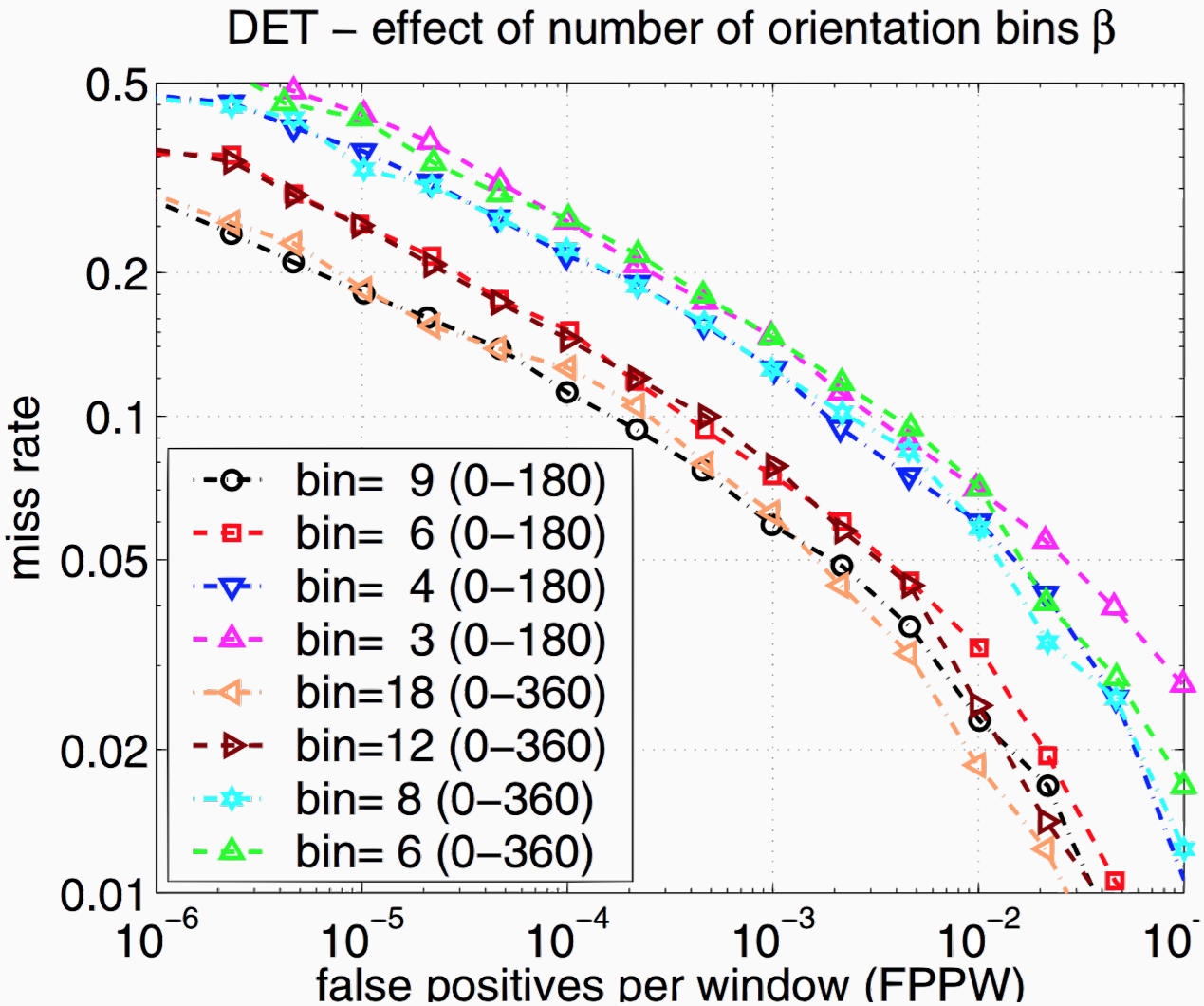
Gradient strengths vary over a wide range owing to local variations in illumination and foreground-background contrast, so effective local contrast normalization turns out to be essential for good performance.

Possible normalization scheme:

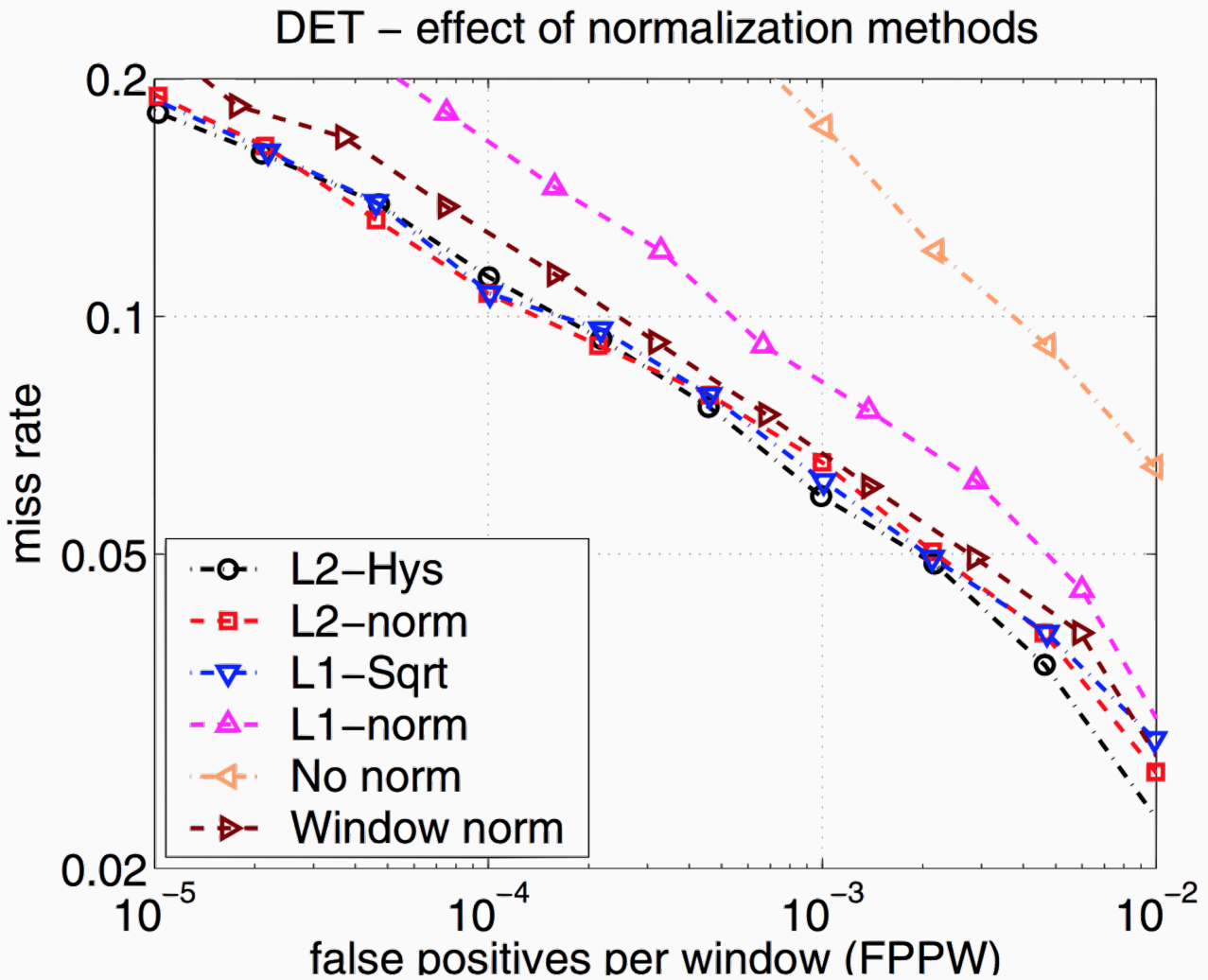
$$H(c,k) = \frac{H_1(c,k)}{\left(\sum_{c_i \in N_c} \sqrt{\|H_1(c_i)\|_2^2 + \varepsilon^2}\right)} \tag{2}$$

where N_c is the set of all cells in a block.

Results

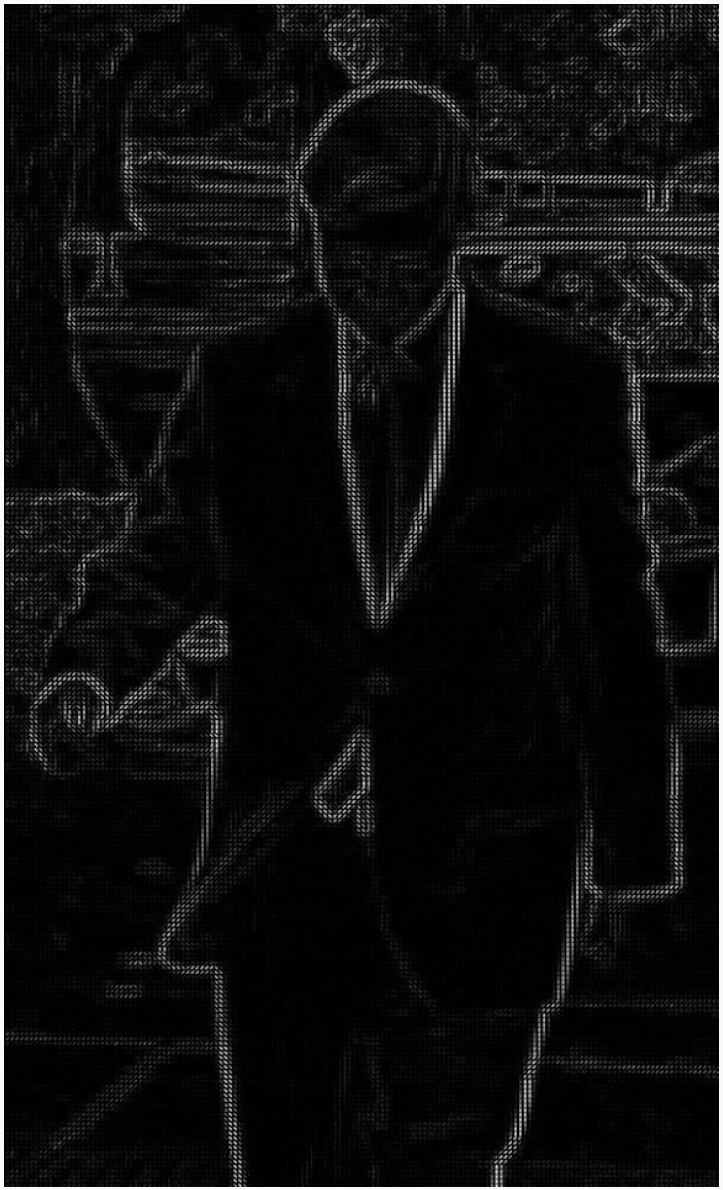

DET – effect of number of orientation bins β
miss rate
0.5
0.2
0.1
0.05
0.02
0.01
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
false positives per window (FPPW)
bin= 9 (0–180)
bin= 6 (0–180)
bin= 4 (0–180)
bin= 3 (0–180)
bin=18 (0–360)
bin=12 (0–360)
bin= 8 (0–360)
bin= 6 (0–360)

Results


DET – effect of normalization methods
miss rate
0.2
0.1
0.05
0.02
10^{-5}
10^{-4}
10^{-3}
10^{-2}
false positives per window (FPPW)
L2–Hys
L2–norm
L1–Sqrt
L1–norm
No norm
Window norm

Demo

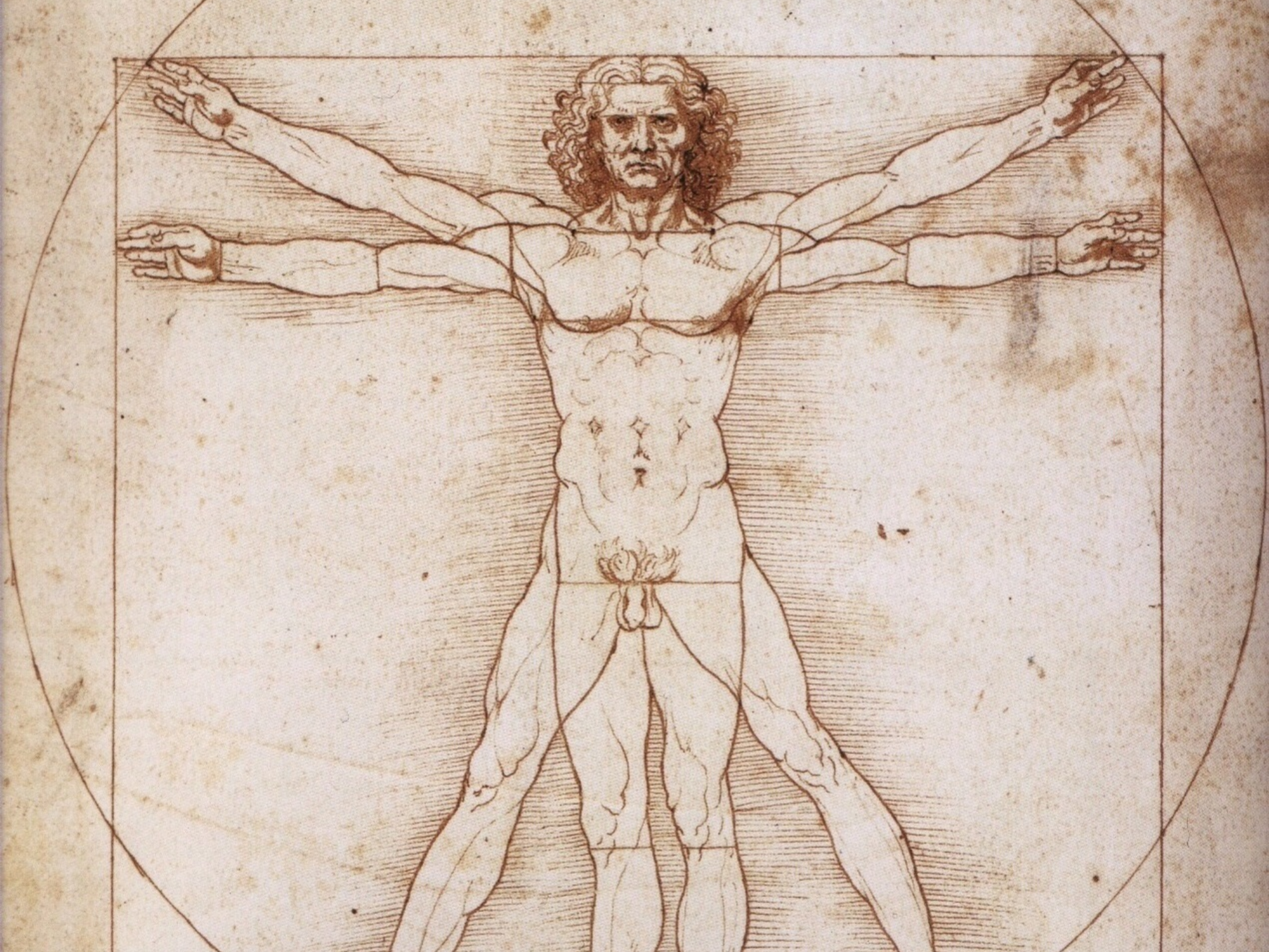
Demo



Summary

The HOG representation has several advantages. It captures gradient structure that is characteristic of local shape, with a representation invariant to local geometric and photometric transformations: translations or rotations make little difference if they are much smaller that the local spatial or orientation bin size.

For *human detection*, **fine orientation sampling** and **strong local photometric normalization** turns out to be the best strategy: it permits body segments to change appearance and move from side to side quite a lot provided that they maintain an upright orientation.



Part-Based Model for Object Detection

Part-Based Model

Idea

An object is made of a set of specific sub-blocks.

Part-Based Model

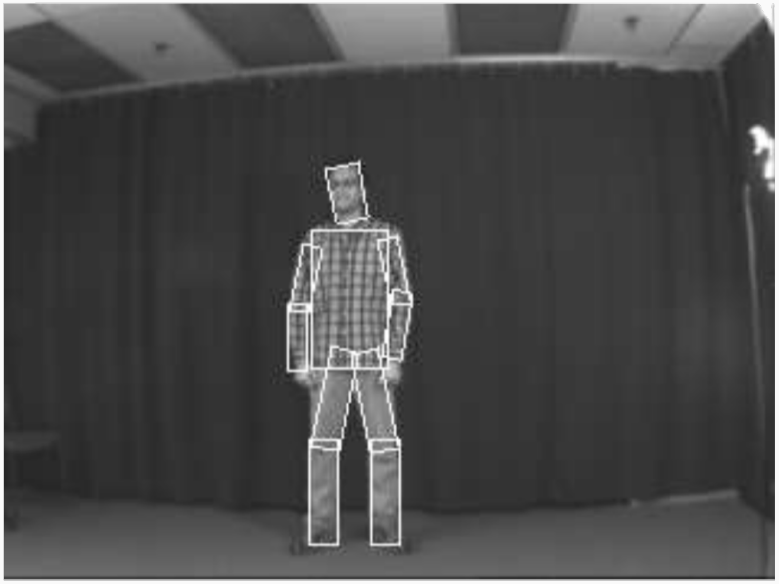
Idea

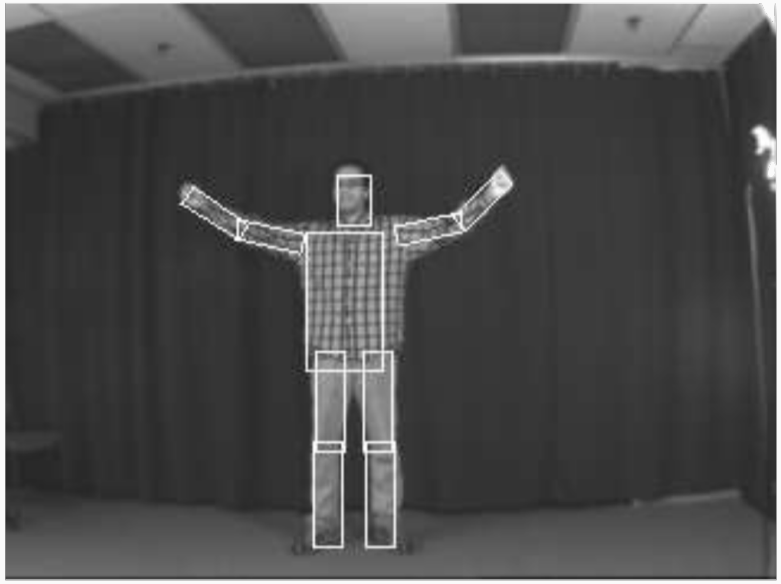
An object is made of a set of specific sub-blocks.

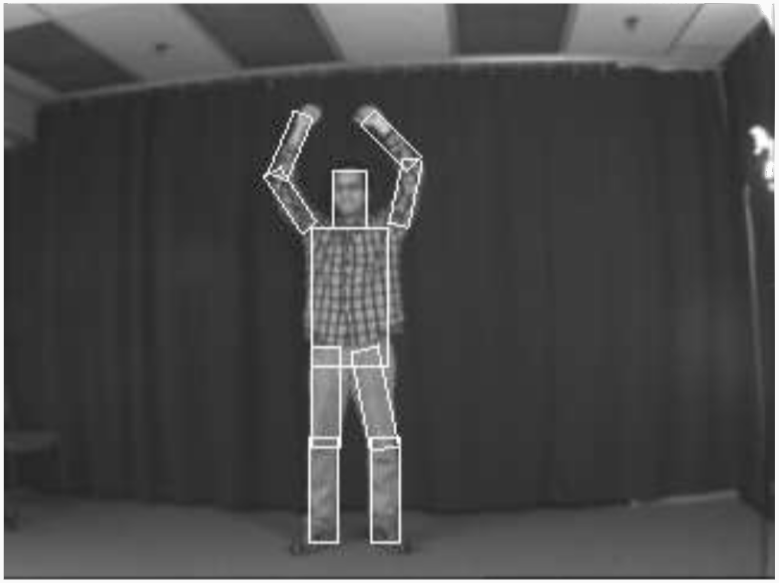
Pictorial structure

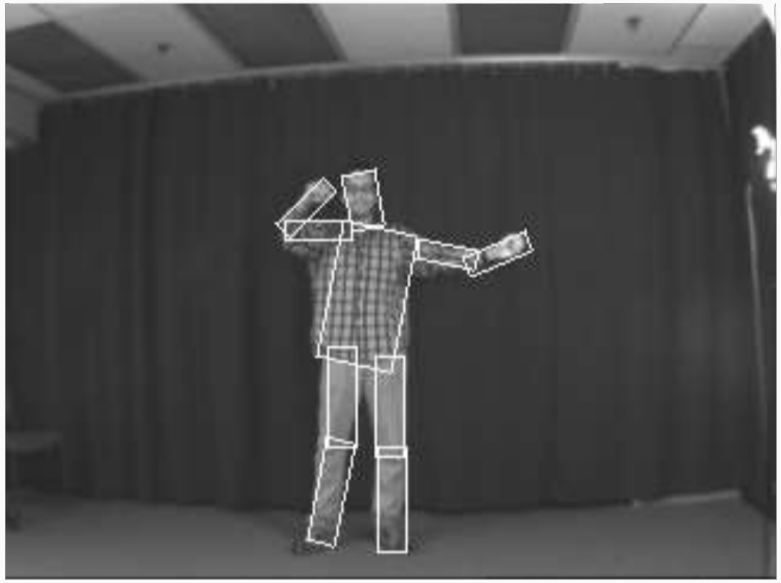
Pictorial structures represent objects by a collection of parts arranged in a deformable configuration.

Pictorial Structures









Part-based Models

Idea

The idea is to give a **score** for each part of the model (e.g. for humans could be arm, chest, legs, face, ...).

Part-based Models

Idea

The idea is to give a **score** for each part of the model (e.g. for humans could be arm, chest, legs, face, ...).

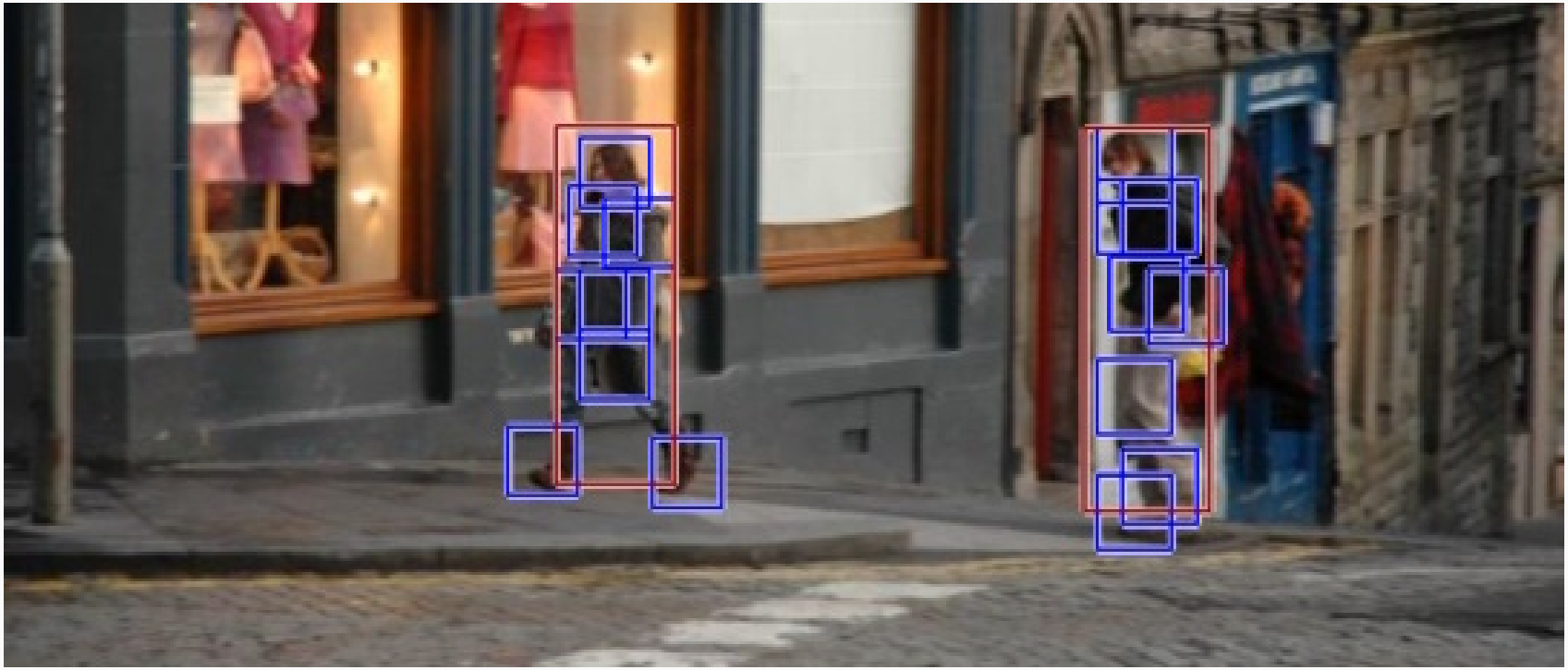
Starting point

We would like to define a score at different positions and scales in an image. This is done using a feature pyramid which specifies a feature map for a finite number of scales in a fixed range.

Scores

Given a model based on n parts, generally:

$$\text{score}(x_0, y_0) = R_{l_0}(x_0, y_0) + \sum_{i=1}^{n} D_{l_0-\lambda}(2(x_0, y_0) + v_i) + b$$



Partial Occlusion Handling

Partial Occlusion Handling

Why?

Generally, especially in crowded scenes, occlusions occur frequently. Nevertheless, generic detectors, such as HOG, assume that pedestrians are fully visible and their performance degrades when pedestrians are partially occluded.

Partial Occlusion Handling

Why?

Generally, especially in crowded scenes, occlusions occur frequently. Nevertheless, generic detectors, such as HOG, assume that pedestrians are fully visible and their performance degrades when pedestrians are partially occluded.

How?

The key to successful detection of partially occluded pedestrians is to use additional information about which body parts are occluded, for example *correlations among the visibilities of different parts* having different sizes.

Partial Occlusion Handling

Probabilistic Framework

It models correlations among the visibilities of parts as hidden variables

Partial Occlusion Handling

Probabilistic Framework

It models correlations among the visibilities of parts as hidden variables

Deep Model

The hierarchical structure of the deep model matches with the multilayers of the parts model well. Different from the other types of deep networks, whose hidden variables had no semantic meaning, this model considers each hidden variable as representing the visibility of a part.

Hidden variable framework

Let $\mathbf{x} \in \mathbb{R}^d$ and $y \in \{0, 1\}$ be respectively the *feature vector* and the *label* of a detection window.

Denote the detection scores of the P parts by $\mathbf{s} = [s_1, ..., s_P]^T = \gamma(\mathbf{x})$, where $\gamma(\mathbf{x})$ are part detectors.

Denote the visibilities of the P parts by $\mathbf{h} = [h_1, ..., h_P]^T \in \{0, 1\}^P$, with $h_i = 1$ meaning **visible** and $h_i = 0$ meaning **invisible**.

$$p(y|\mathbf{x}) = \sum_{\mathbf{h}} p(y, \mathbf{h}|\mathbf{x}) = \sum_{\mathbf{h}} p(y|\mathbf{h}, \mathbf{x}) p(\mathbf{h}|\mathbf{x}) \qquad (3)$$

Deep Model for Part Visibility Estimation

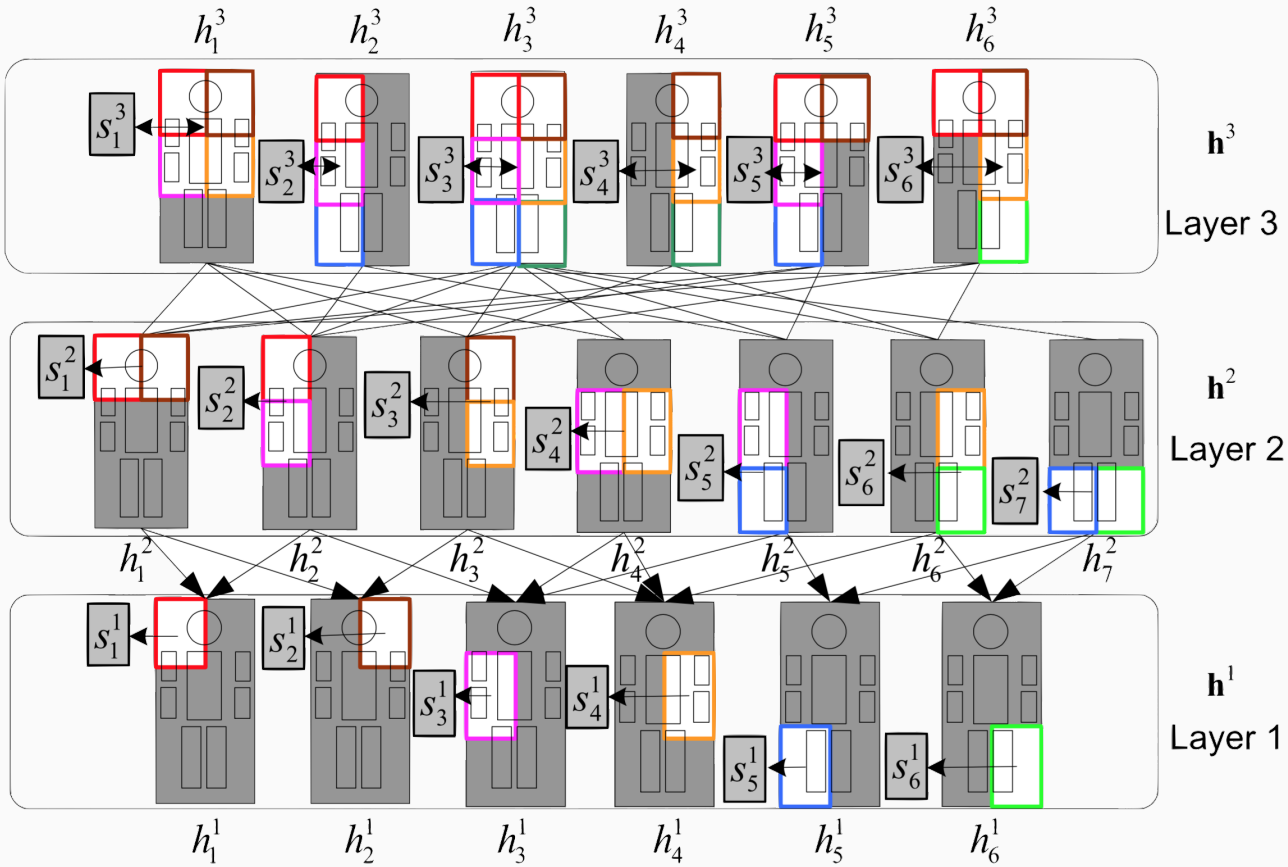
Objective

Build a deep model that learns the correlation of visibility relationship among parts

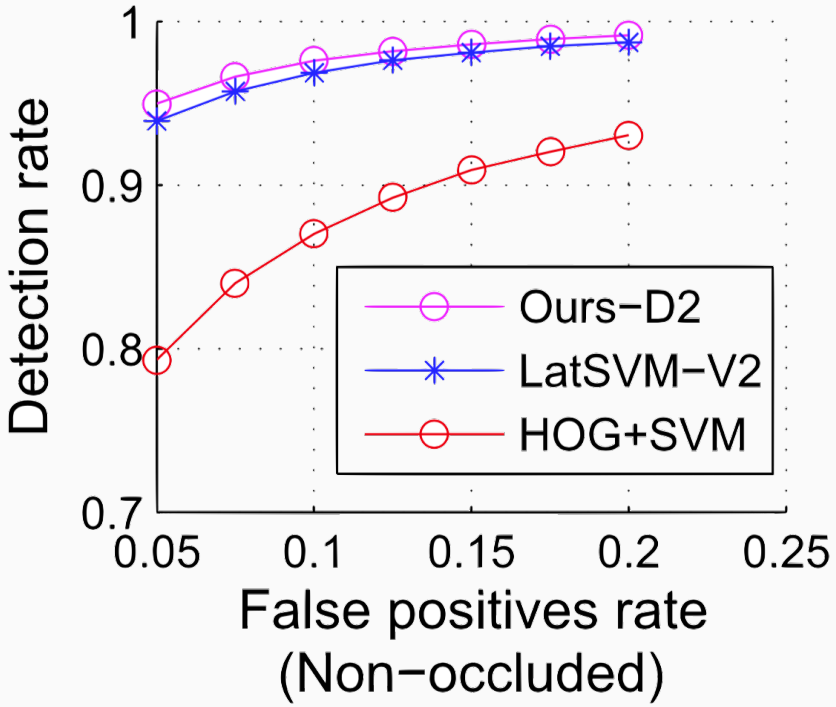
Deep Model for Part Visibility Estimation

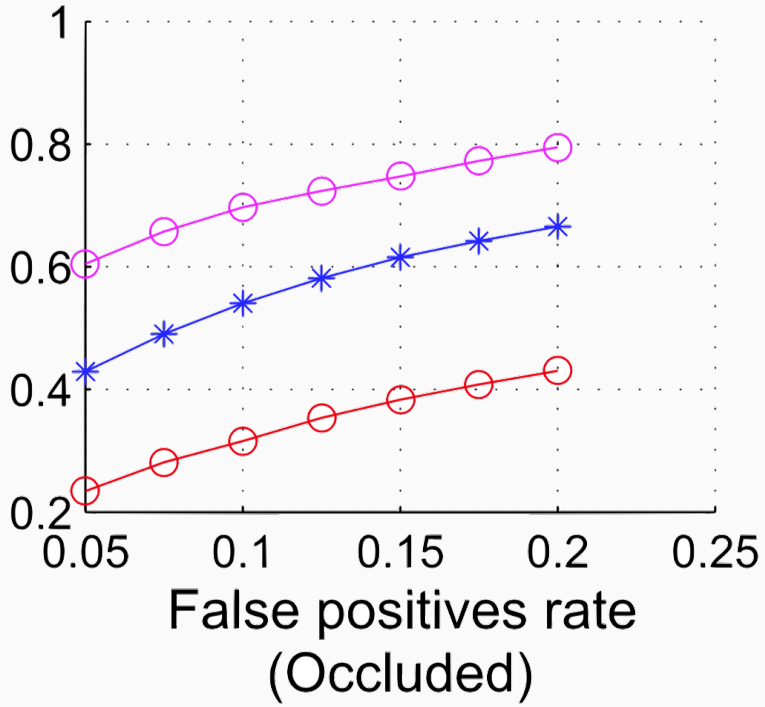
Objective

Build a deep model that learns the correlation of visibility relationship among parts



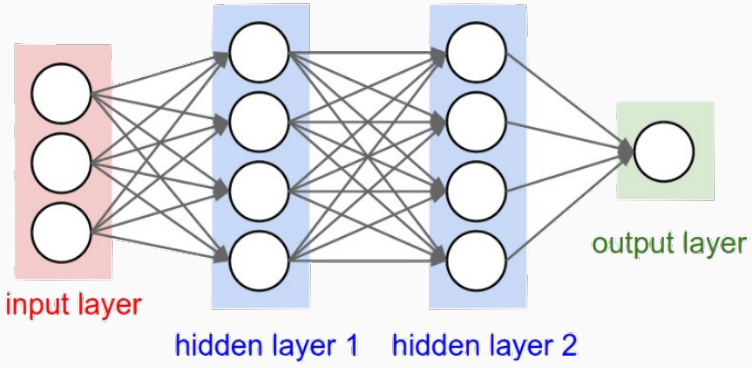
Result


Detection rate
1
0.9
0.8
0.7
0.05
0.1
0.15
0.2
0.25
False positives rate
(Non−occluded)
Ours−D2
LatSVM−V2
HOG+SVM

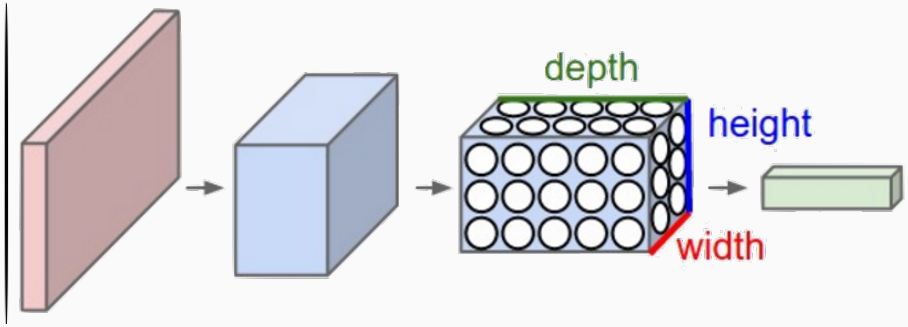

1
0.8
0.6
0.4
0.2
0.05
0.1
0.15
0.2
0.25
False positives rate
(Occluded)

Convolutional Neural Networks for Pedestrian Detection

Convolutional Neural Networks



Neural Network



Convolutional Neural Network

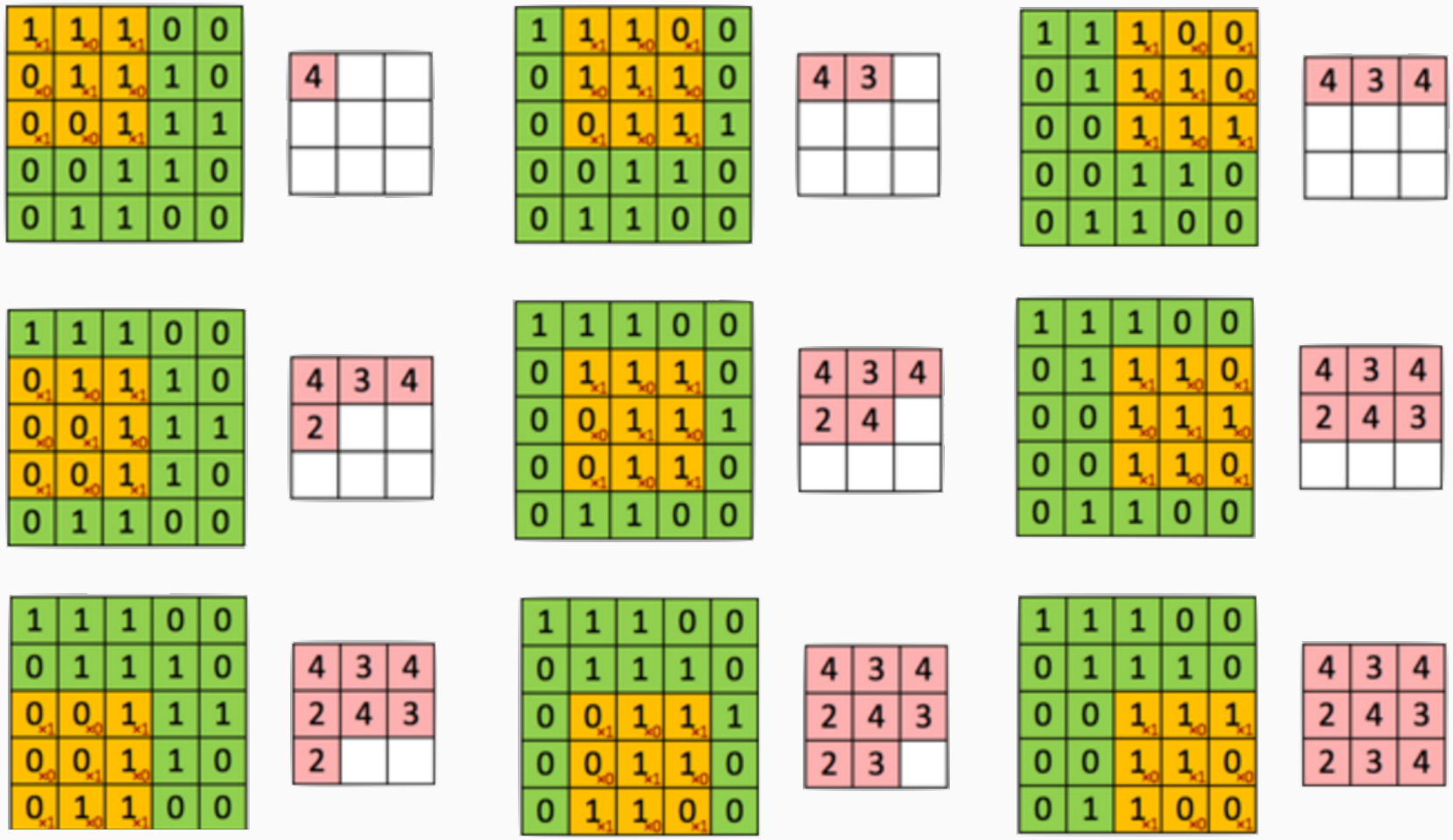
CNN Layers

A CNN is made of different layers:

- **Convolution**
- **Non linearity (ReLU)**
- **Pooling**
- **Fully-Connected**

Many combinations have been proposed

Convolution Layer



Convolution Layer

How many convolution matrices? $\rightarrow$ Number of filters or depth

Size of convolution matrices? $\rightarrow$ Depth

Step between each convolution? $\rightarrow$ Stride

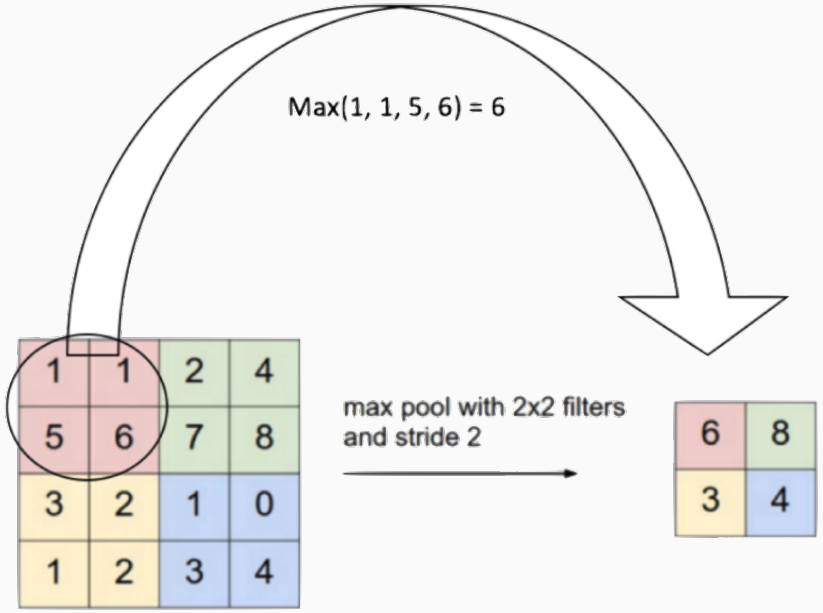
Zero-padding for convolution on the border?

Pooling Layer

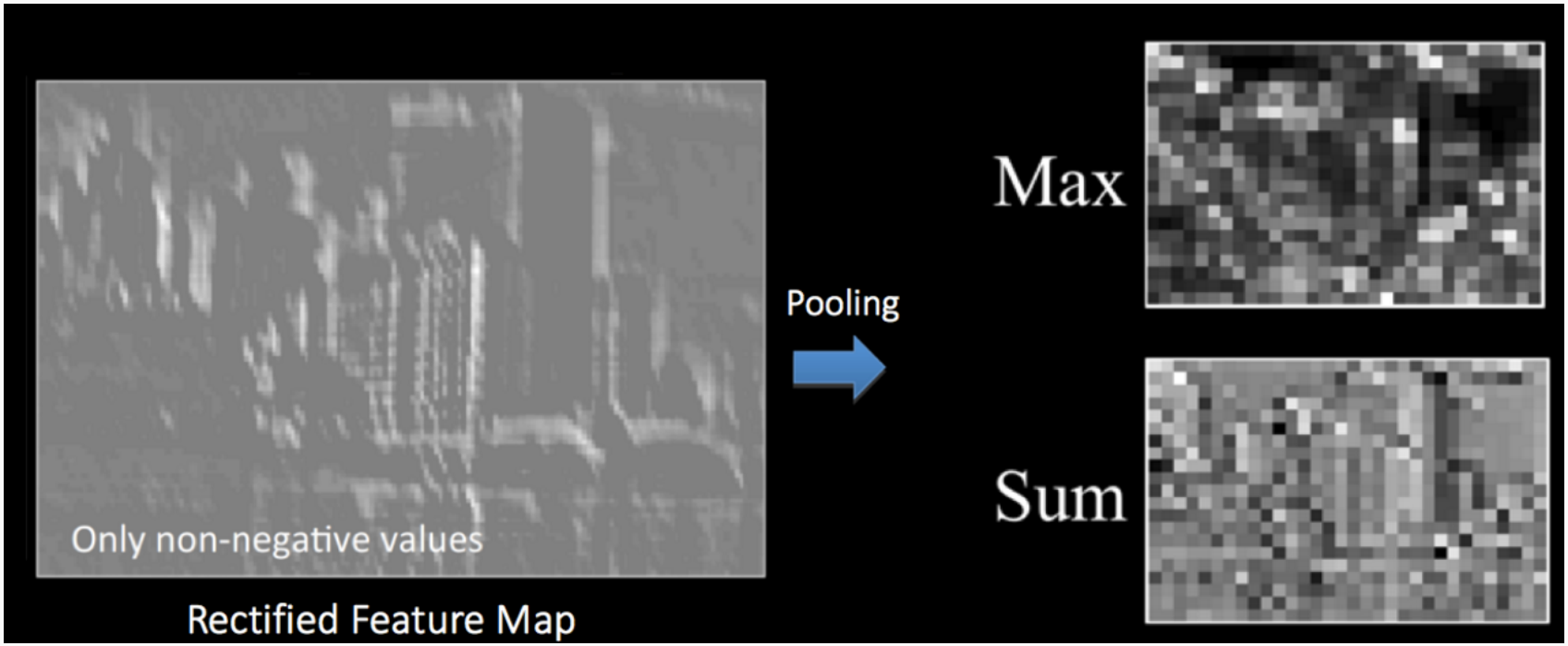
Downsampling operation (Max, Average, Sum etc.)

Reduces the dimensionality of each feature map

Retains the most important information



Pooling Layer



Fully-Connected Layer

Classification operation

Traditional Multi Layer Perceptron

Computes the class scores

DeepPed

Improving and adapting AlexNet to pedestrian detection

Low miss rate (MR), about 19.9%

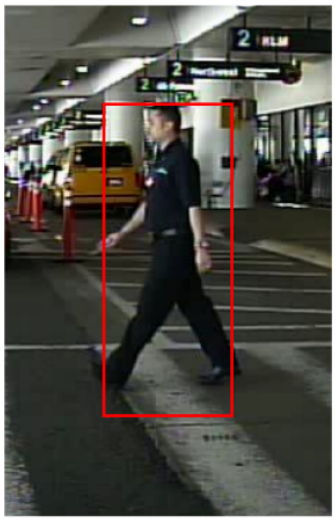
Not real-time, only 2 fps

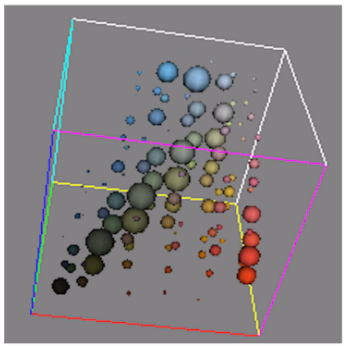
Model:

- Locally Decorrelated Channel Features (LDCF)
- Data preprocessing
- Fine-tuned CNN
- SVM (using LDCF scores)

Data Preprocessing

Padding to overcome the issue of imprecise proposed region
Negative Sample Decorrelation selects negative training examples that are as diverse as possible



DeepCascade

Improving and adapting AlexNet to pedestrian detection
Good accuracy: MR $= 26.2\%$
Real-time: 15 fps
Cascade model:

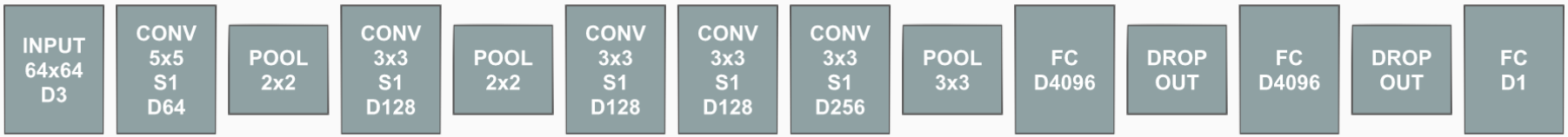
- frequently used method for **speeding up** classifiers
- divides classifiers into a sequence of **simpler classifiers**
- VeryFast cascade reduces recalls at each stage, increasing MR

DeepCascade - Model

1. Pretraining
    - pre-initialized weights based on Imagenet
    - easy to incorporate and increases accuracy
2. Soft-cascade
    - aborts the evaluation of non-promising detections
    - hybrid approach uses 10% of the stages of VeryFast
3. Tiny CNN classifier
4. Modified AlexNet
    - reduced depth and size
    - faster

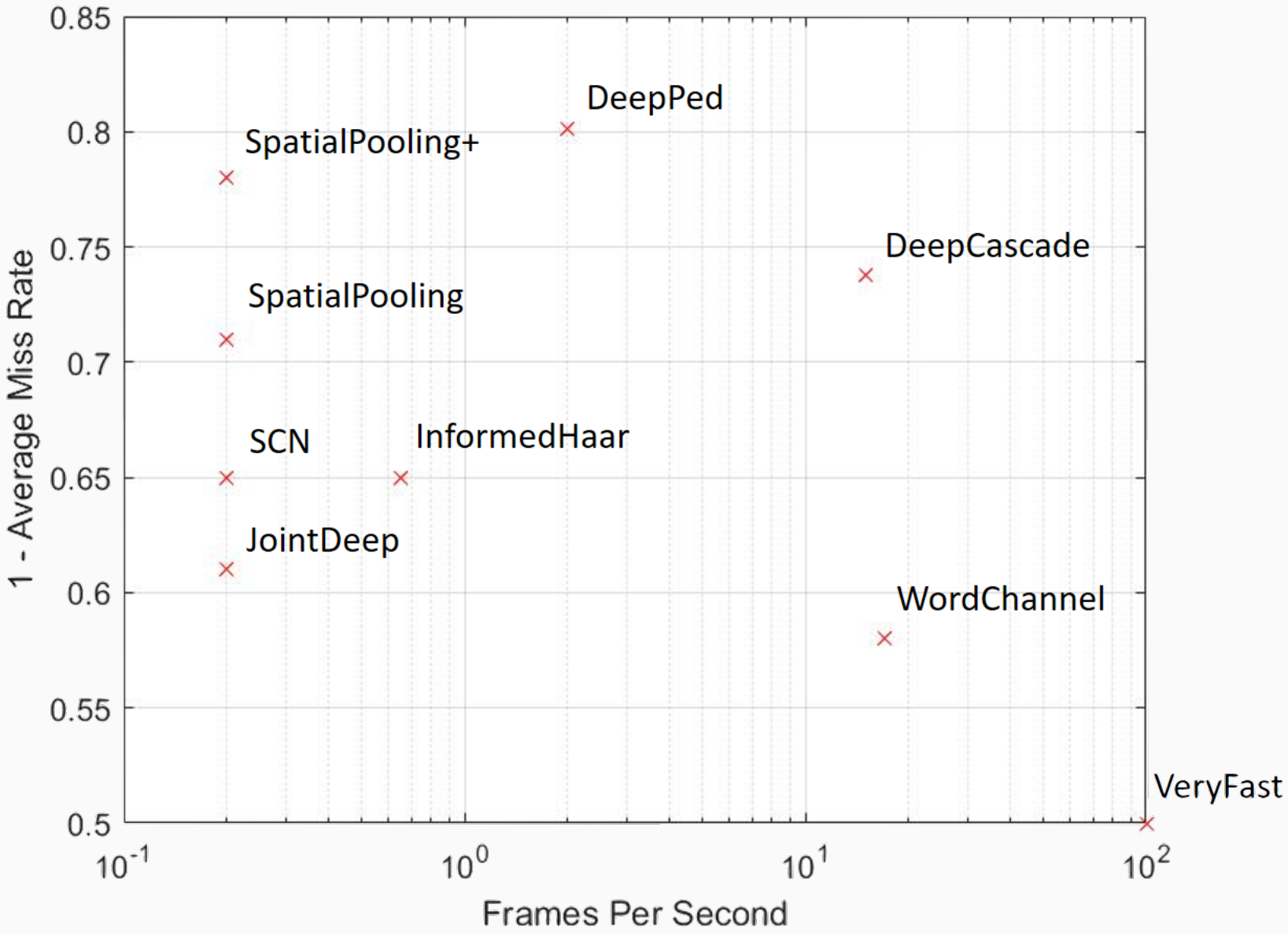
Classifier and Detector

The architecture of the tiny CNN classifier



The architecture of the modified AlexNet

Comparison


0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
1 - Average Miss Rate
10^{-1}
10^{0}
10^{1}
10^{2}
Frames Per Second
DeepPed
SpatialPooling+
DeepCascade
SpatialPooling
SCN
InformedHaar
JointDeep
WordChannel
VeryFast

Conclusions


LEFT REARWARD VEHICLE CAMERA
MEDIUM RANGE VEHICLE CAMERA
RIGHT REARWARD VEHICLE CAMERA
ROAD FLOW
IN-PATH OBJECTS
ROAD LIGHTS
OBJECTS
ROAD SIGNS

Conclusions

What have we seen so far?

Different methods for pedestrain detection capable of distinguish an human in a real world scenario.

Conclusions

What have we seen so far?

Different methods for pedestrain detection capable of distinguish an human in a real world scenario.

Are these result good enough?

NO

Conclusions

What have we seen so far?

Different methods for pedestrain detection capable of distinguish an human in a real world scenario.

Are these result good enough?

NO

Why?

High miss rate (15% to 25%) and difficulties on real-time implementations

Questions?

Bibliography i

Angelova, A., Krizhevsky, A., and Vanhoucke, V.

Real-time pedestrian detection with deep network cascades.

Paper from Google.

Bibliography ii

Dalal, N., and Triggs, B.

Histograms of oriented gradients for human detection.

Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2005).

Bibliography iii

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D.

Object detection with discriminatively trained part-based models.

IEEE transactions on pattern analysis and machine intelligence 32, 9 (September 2010), 1627–1645.

Bibliography iv

Ouyang, W., Zeng, X., and Wang, X.

Partial occlusion handling in pedestrian detection with a deep model.

IEEE transactions on circuits and systems for video technology 26, 11 (November 2016), 2123–2137.

Bibliography v

Tome, D., Monti, F., and Baroffio, L.

Deep convolutional neural networks for pedestrian detection.

Elsevier Journal of Signal Processing: Image Communication (2016).