GPU-based pedestrian detection for autonomous driving

V. Campmany[1,2,3], S. Silva[1,2,3], J.C. Moure[1,2], T. Espinosa[1,2], D. Vazquez[1,3], A. M. Lopez[1,3]
[1]Autonomous University of Barcelona   [2]Computer Architecture & Operating Systems
[3]Computer Vision Center

Abstract

Pedestrian detection for autonomous driving is one of the hardest tasks within computer vision, and involves huge computational costs. Obtaining acceptable real-time performance, measured in frames per second (fps), for the most advanced algorithms is nowadays a hard challenge. Taking the work in [1] as our baseline, we propose a CUDA implementation of a pedestrian detection system that includes LBP and HOG as feature descriptors and SVM and Random forest as classifiers. We introduce significant algorithmic adjustments and optimizations to adapt the problem to the NVIDIA GPU architecture. The aim is to deploy a real-time system providing reliable results.
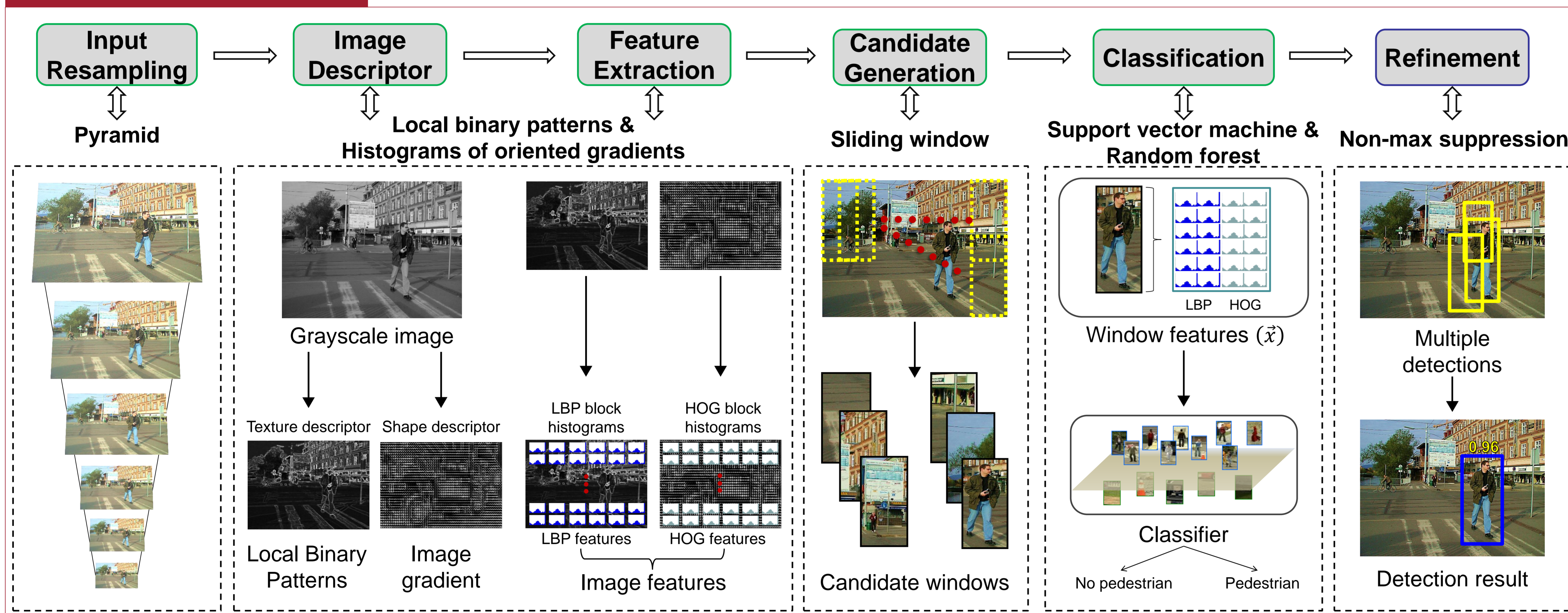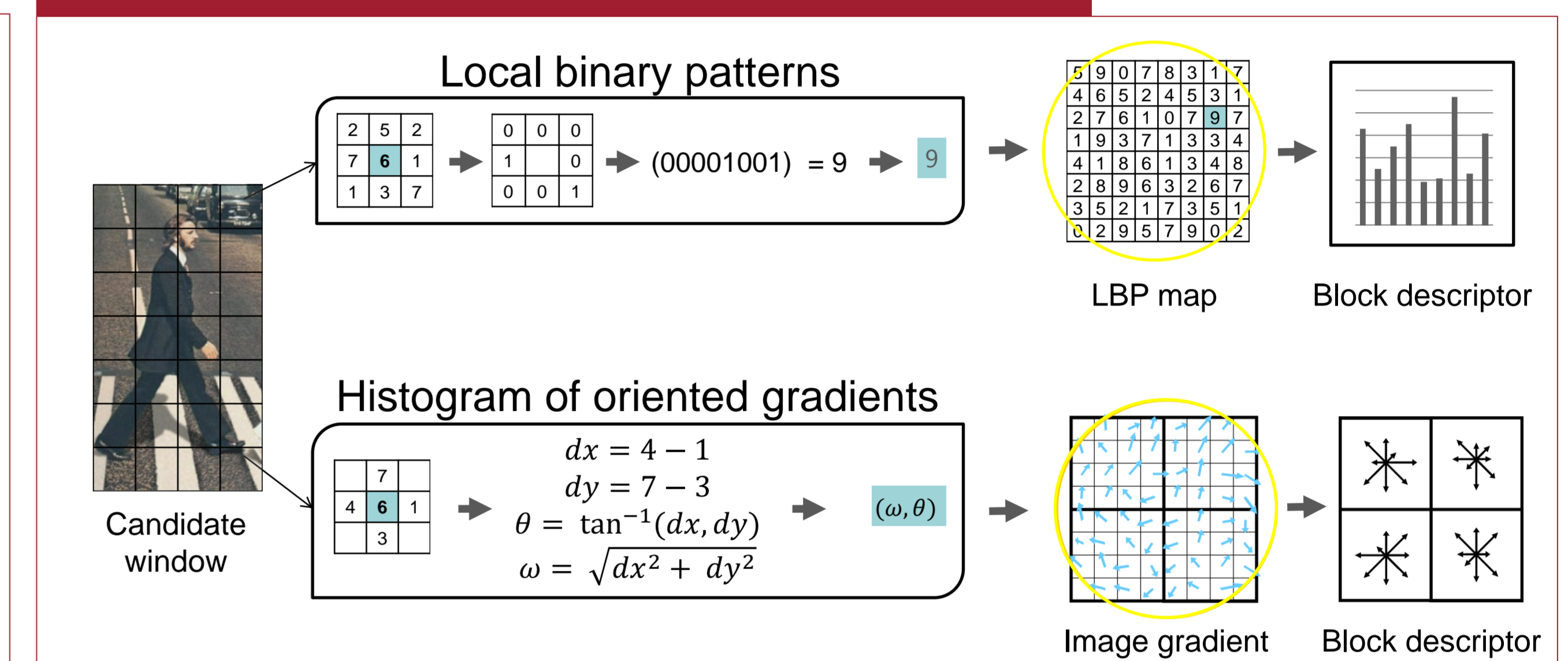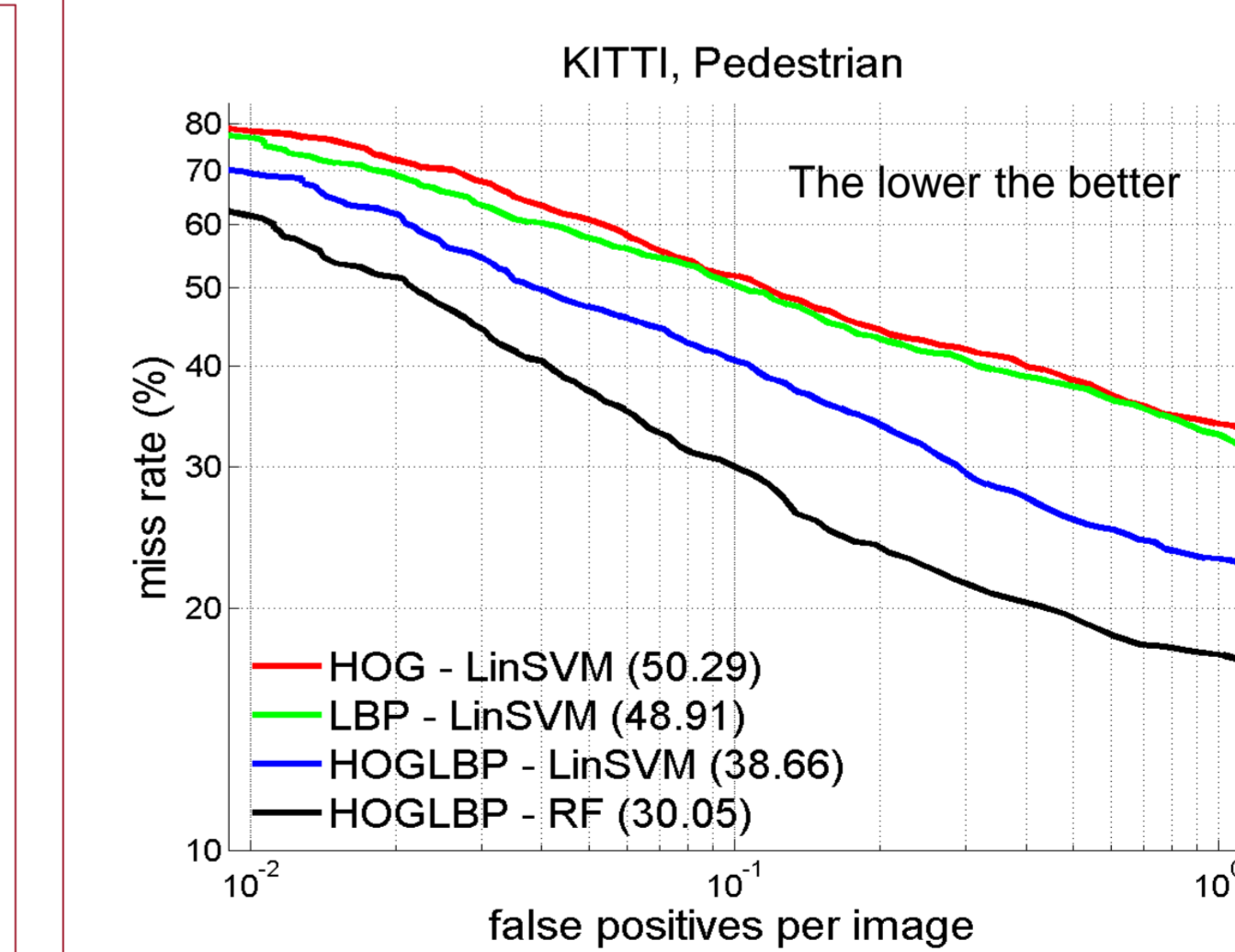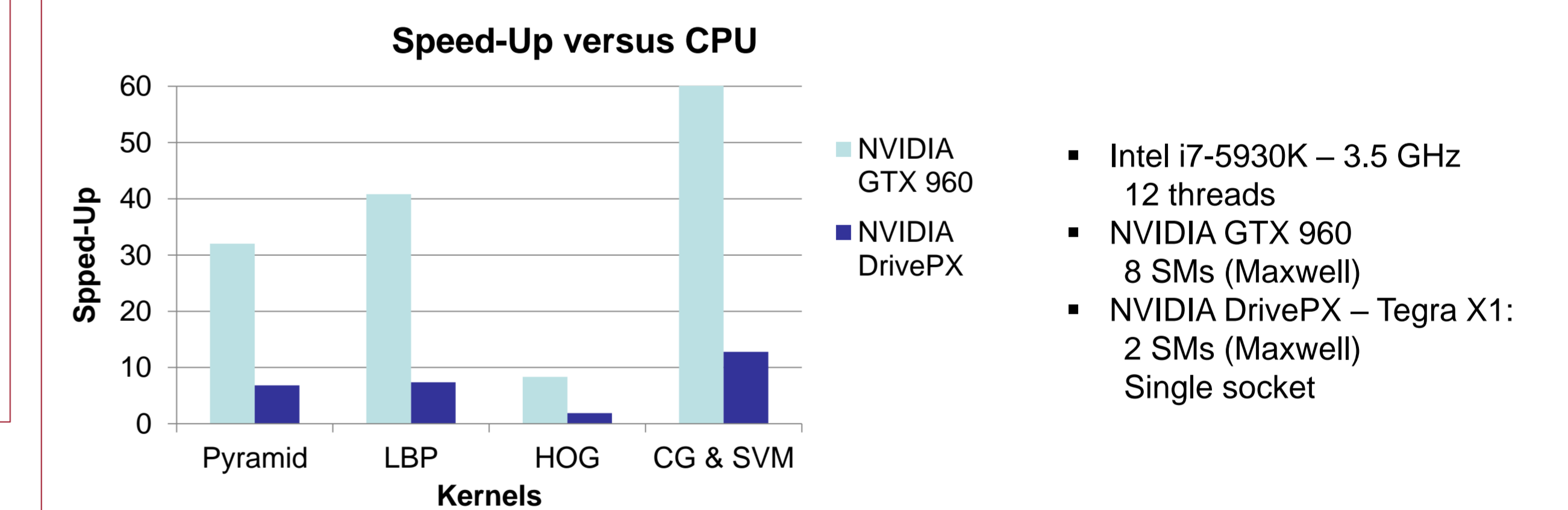
Object Detection Pipeline

Input Resampling → Image Descriptor → Feature Extraction → Candidate Generation → Classification → Refinement

Pyramid — Local binary patterns & Histograms of oriented gradients — Sliding window — Support vector machine & Random forest — Non-max suppression

Grayscale image

Texture descriptor — Shape descriptor

Local Binary Patterns — Image gradient

LBP block histograms — HOG block histograms

LBP features — HOG features — Image features

Candidate windows

Window features ($\vec{x}$)

LBP — HOG

Classifier → No pedestrian / Pedestrian
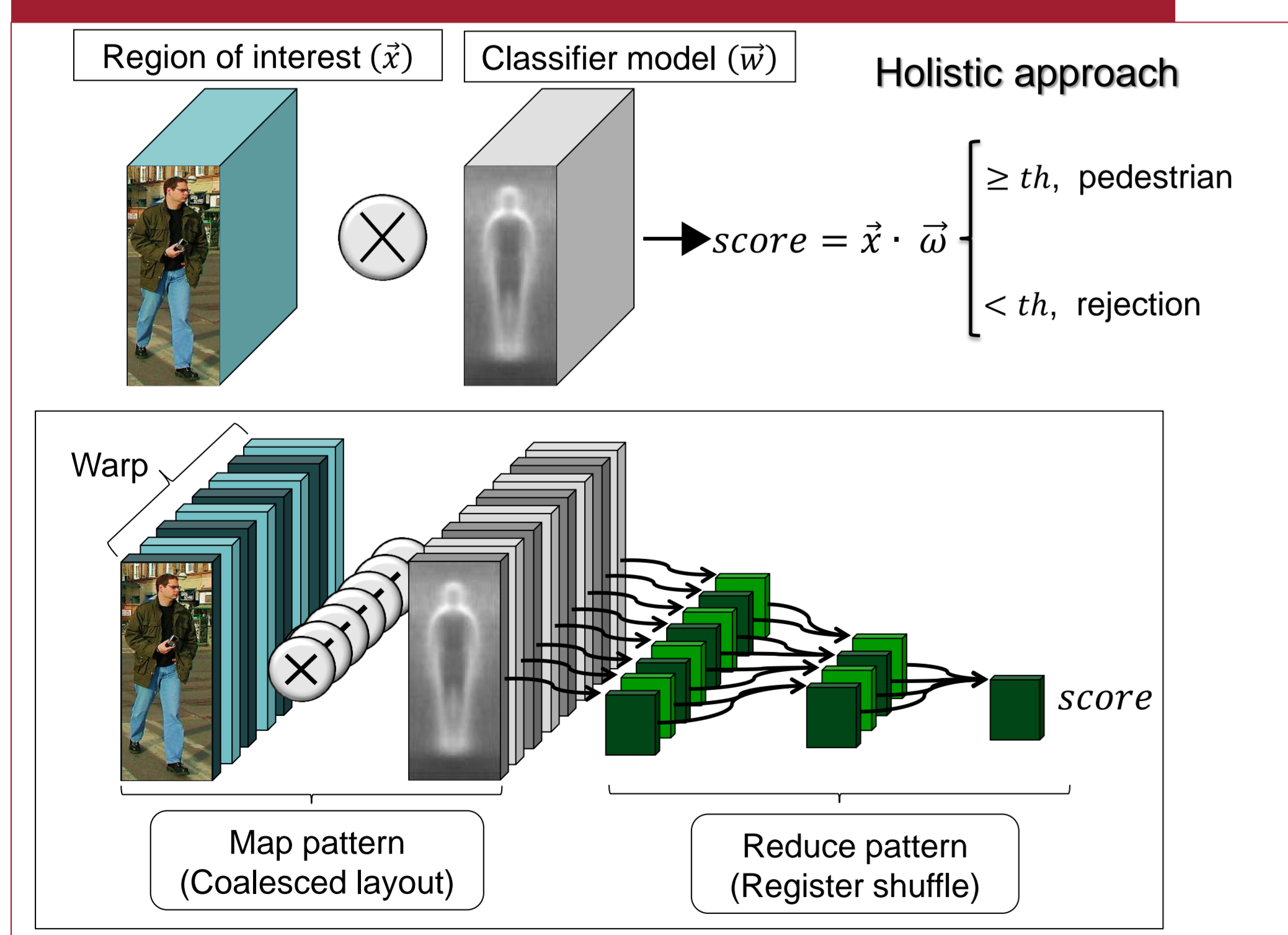
Multiple detections

Detection result

Image Descriptor & Feature Extraction

Local binary patterns

$(00001001) = 9$

LBP map — Block descriptor

Histogram of oriented gradients

$dx = 4 - 1$
$dy = 7 - 3$
$\theta = \tan^{-1}(dx, dy)$
$\omega = \sqrt{dx^2 + dy^2}$

(ω, θ)

Candidate window — Image gradient — Block descriptor

Classification: Support vector machine

Region of interest ($\vec{x}$) — Classifier model ($\vec{w}$)

Holistic approach

$$score = \vec{x} \cdot \vec{\omega}$$

$\geq th$, pedestrian
$< th$, rejection

Warp

Map pattern (Coalesced layout) — Reduce pattern (Register shuffle)

Classification: Random forest

Patch-based approach

Soft cascade

$score_1$ — $score_2$ — $score_n$

Rejected

$$score = \frac{1}{n}\sum_{i=1}^{n} conf_i$$

Pedestrian

Intelligent early rejection reduces total work

Each decision tree makes local guess. Weak predictions ensembled to get a reliable classifier.

Binary tree of local experts

($\vec{x}$)

Split nodes

($\vec{w}_1$)
($\vec{w}_2$)   ($\vec{w}_3$)

Leafs

$score_n$

Results

Speed-Up versus CPU

- NVIDIA GTX 960
- NVIDIA DrivePX

Kernels: Pyramid, LBP, HOG, CG & SVM

- Intel i7-5930K – 3.5 GHz 12 threads
- NVIDIA GTX 960 8 SMs (Maxwell)
- NVIDIA DrivePX – Tegra X1: 2 SMs (Maxwell) Single socket

KITTI, Pedestrian — The lower the better

- HOG - LinSVM (50.29)
- LBP - LinSVM (48.91)
- HOGLBP - LinSVM (38.66)
- HOGLBP - RF (30.05)

LBP + SVM	FPS	$FPS/Watt$
Intel i7-5930K	3,7	0,03
NVIDIA GTX 960	89	0,75
NVIDIA DrivePX	19	2,1

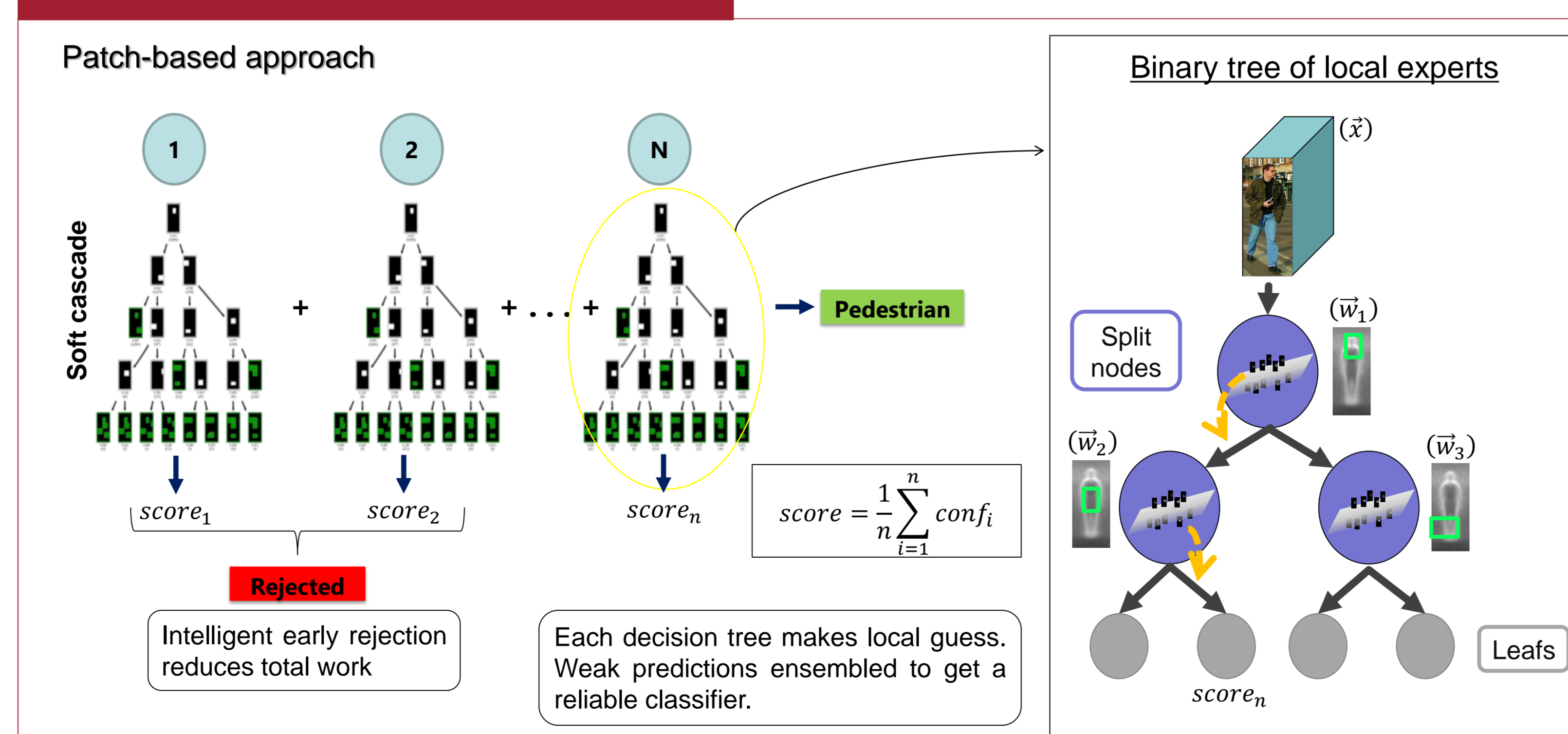| HOG + SVM	FPS	$FPS/Watt$
Intel i7-5930K	3,2	0,02
NVIDIA GTX 960	32	0,22
NVIDIA DrivePX	10	1,1

Conclusions & future work

- Our CUDA system is capable of running on real time obtaining accurate results.
- There is computational power left to integrate multiple cameras and additional ADAS systems such as lane departure warning, traffic signs recognition and 3D vision.

References:
[1] Marin, J., Vazquez, D., Lopez, A. M., Amores, J., & Leibe, B. (2013, December). Random forests of local experts for pedestrian detection. In Computer Vision (ICCV), 2013 IEEE International Conference on (pp.2592-2599).