University of Information Technology
Faculty of Computer Science

Introduction to Computer Vision

Pedestrian Detection

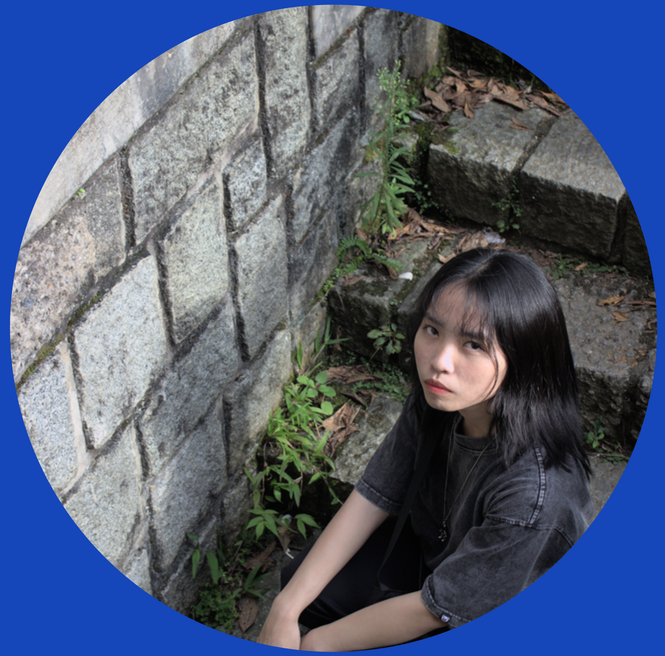
Lecturer

TS. Mai Tiến Dũng

Members

Nguyễn Như Hà
21522028

Nguyễn Thị Thanh Lan
21521065



Bùi Lê Khánh Linh
21522284

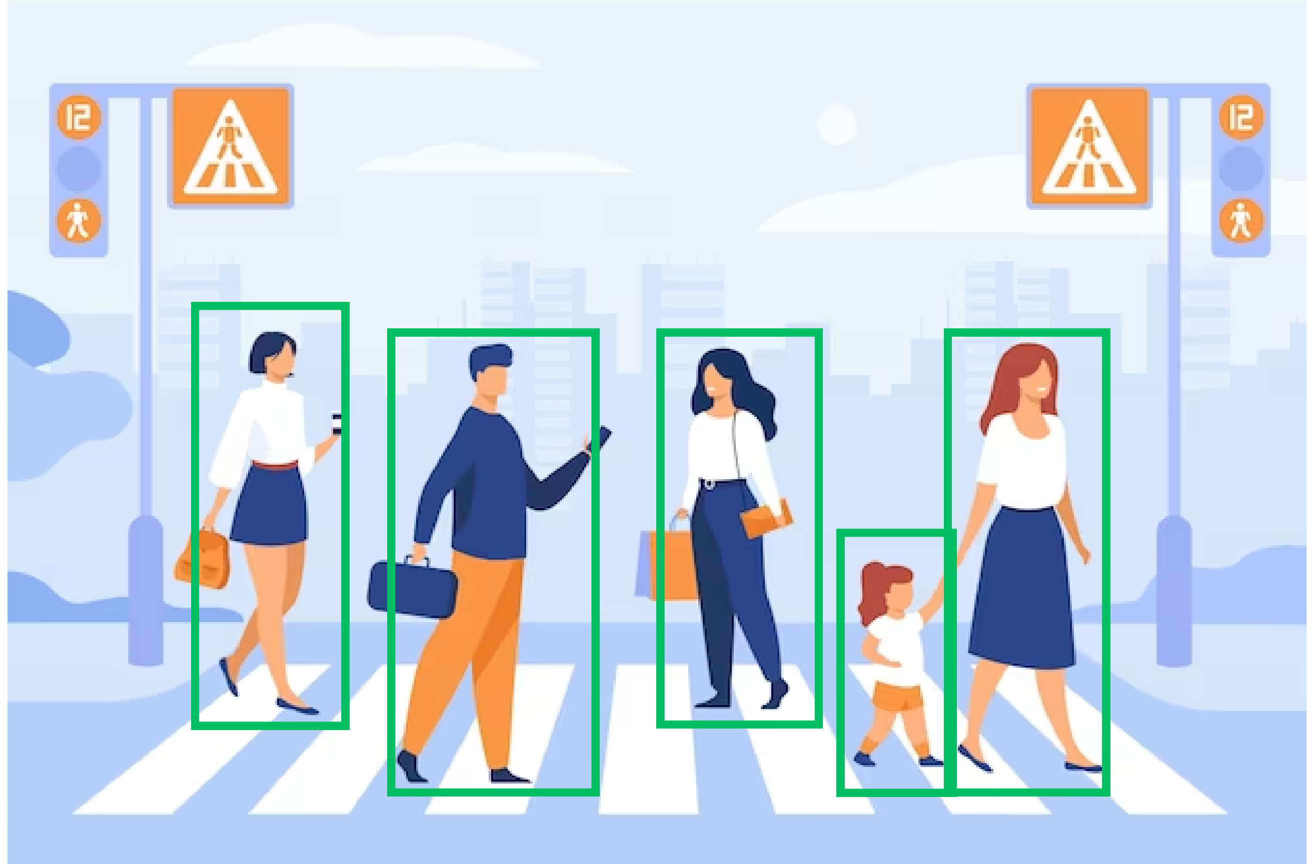
Table of Contents

Pedestrian Detection problem

Generalize the problem: input, output, application in real life

I. Pedestrian Detection problem

Input and Output



Input

An image (single frame).

Output

Image includes coordinates (bounding box) of each detected pedestrian (if any)

I. Pedestrian Detection problem

Application in real life

Autopilot

Driver assistance system, recognise and stop for pedestrians in crosswalks

https://www.tesla.com/autopilot

BLAXTAIR

Pedestrian & Obstacle Proximity Detection

https://blaxtair.com/en/products/blaxtair-pedestrian-obstacle-detection-camera

Implementation

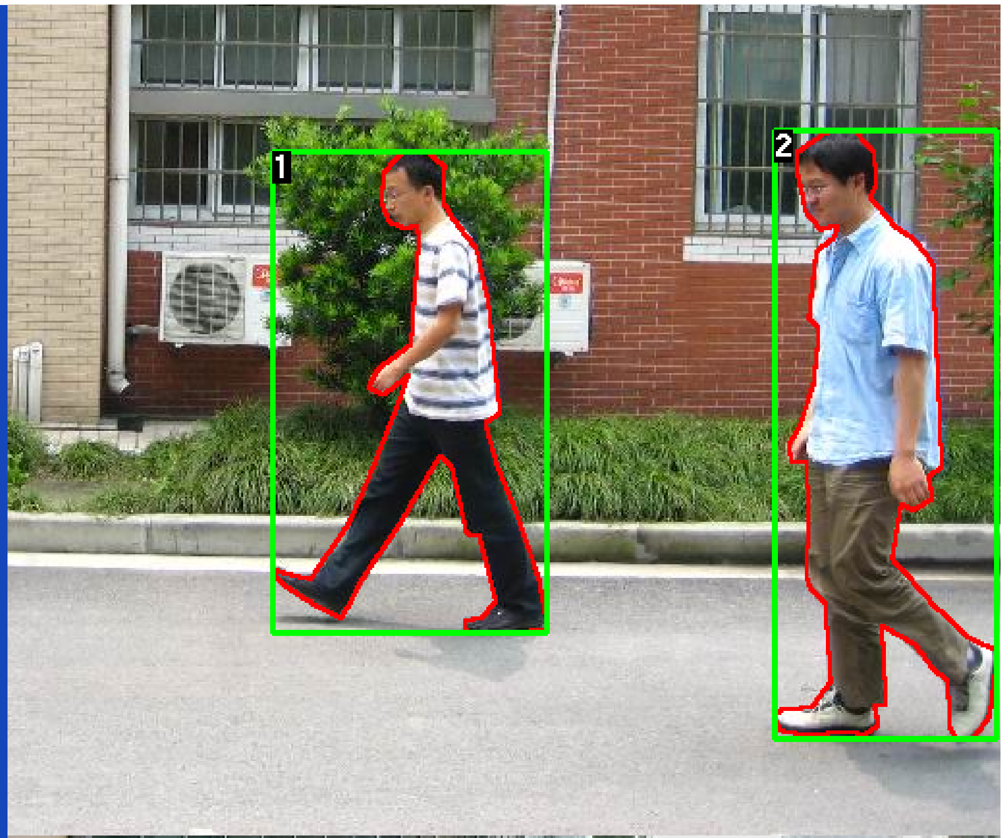
Dataset, Metric, HOG+SVM, Faster RCNN, Demo, Conclusion

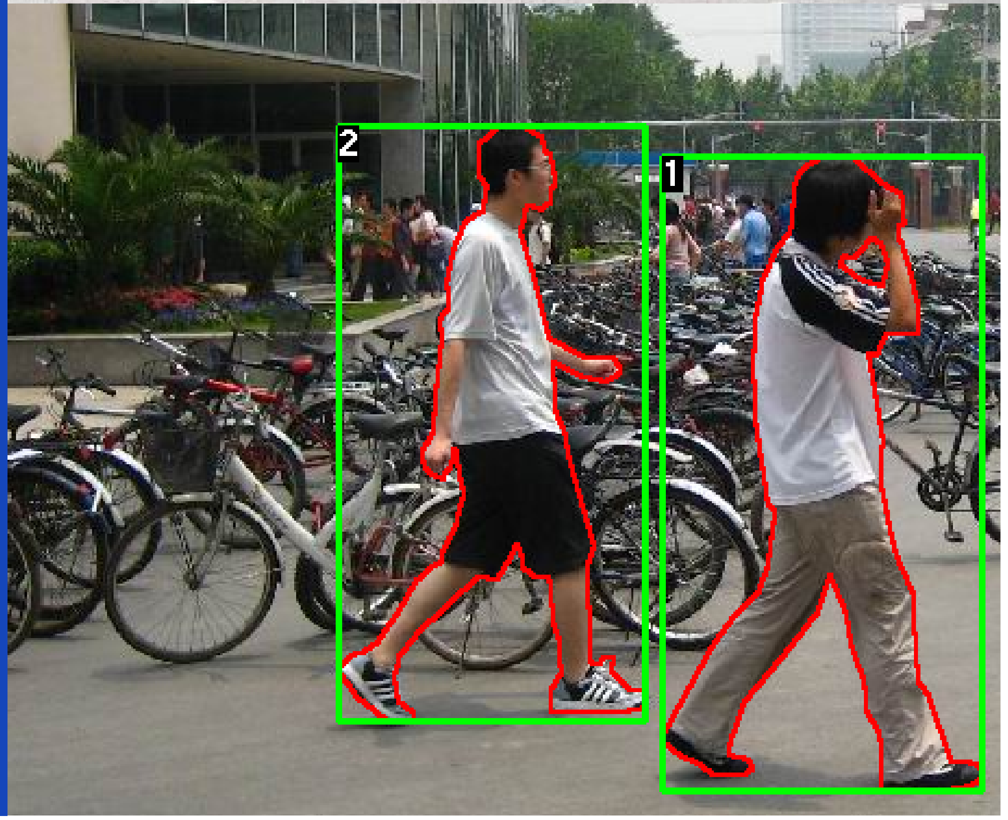
1. Dataset

Penn-Fudan Database
for Pedestrian Detection and Segmentation

Overall:

- Dataset includes 170 images with 345 labeled pedestrians
- The main subject of the images is one or many pedestrians.
- The height of pedestrians is about (180x390) pixels
- All detected pedestrians are upright.





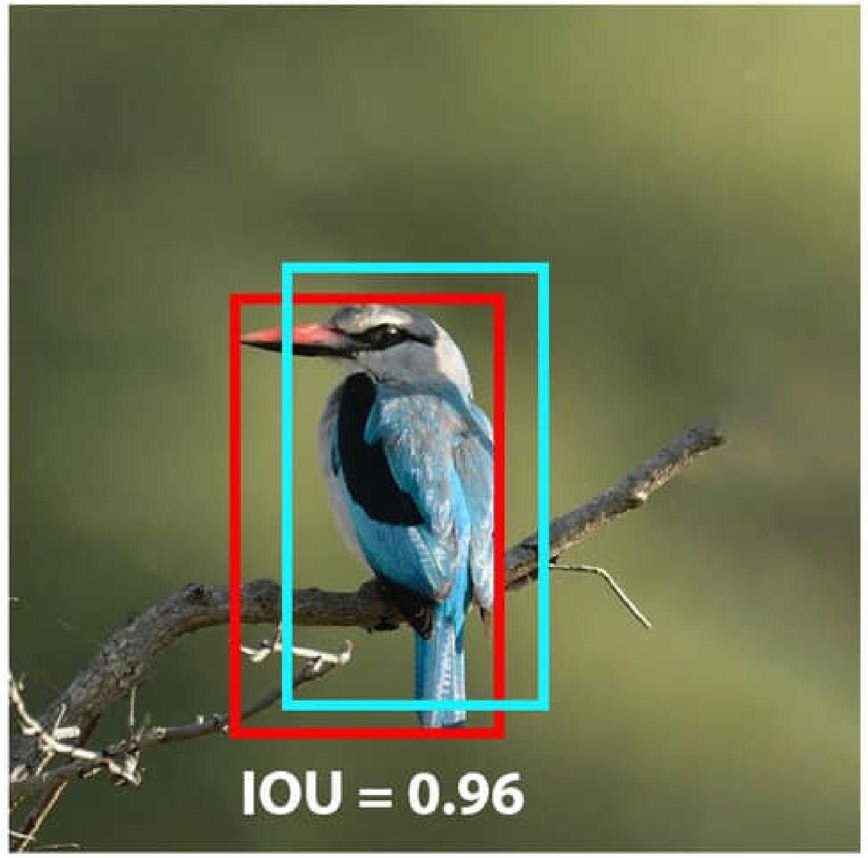
2. Metrics

Intersection over Union (IoU)

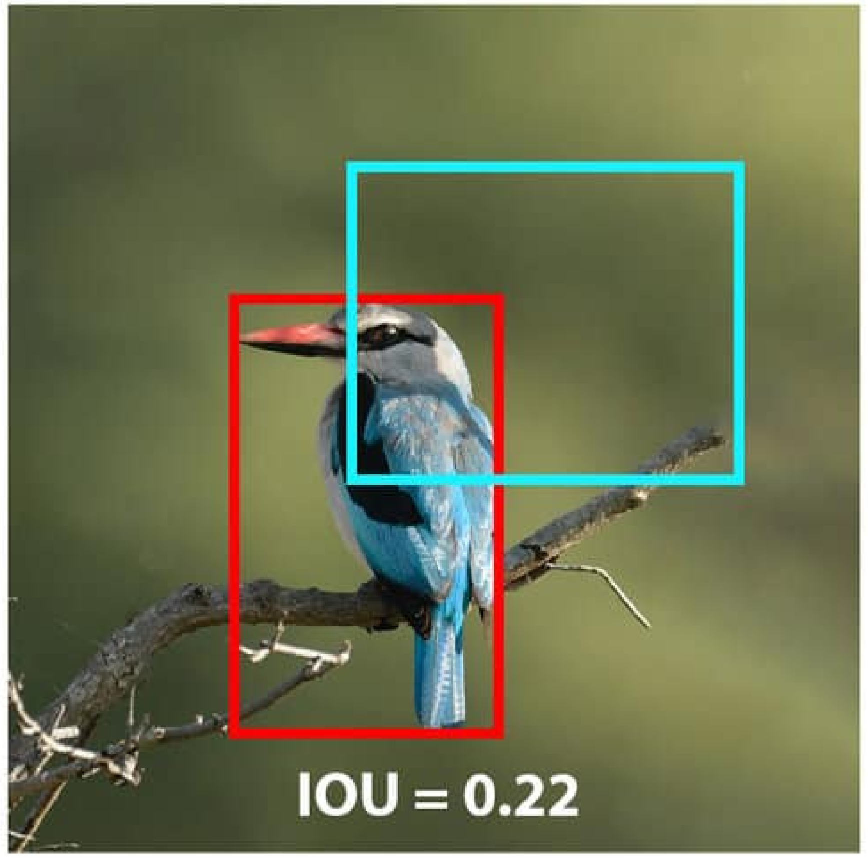
$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

2. Metrics

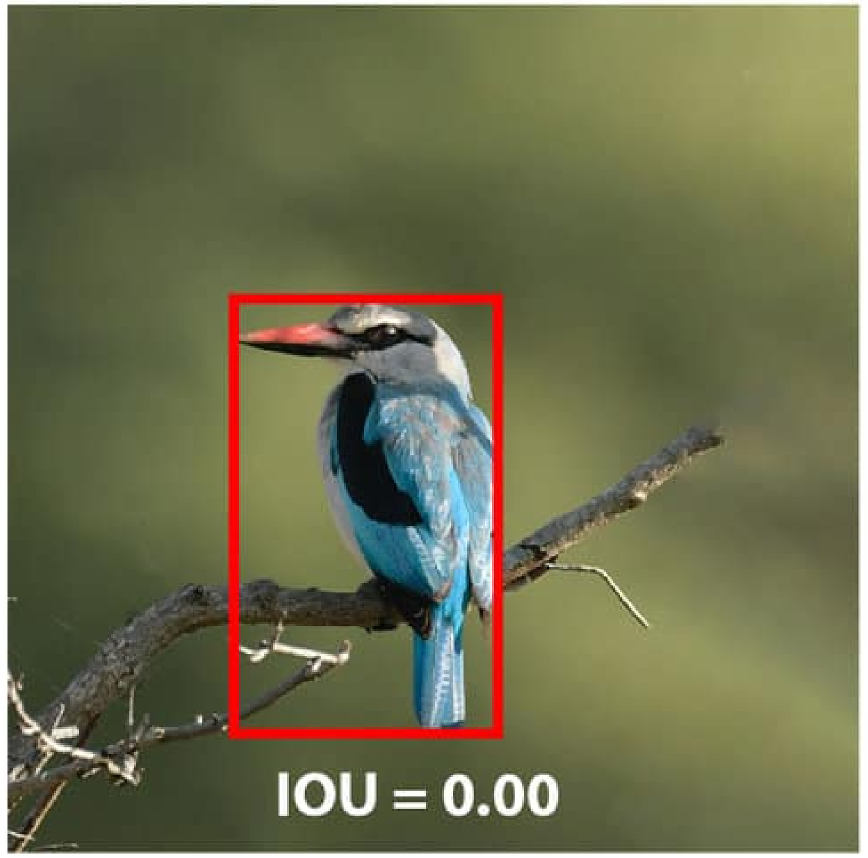
Intersection over Union (IoU)



True Positive



False Positive



False Negative

2. Metrics

Precision & Recall

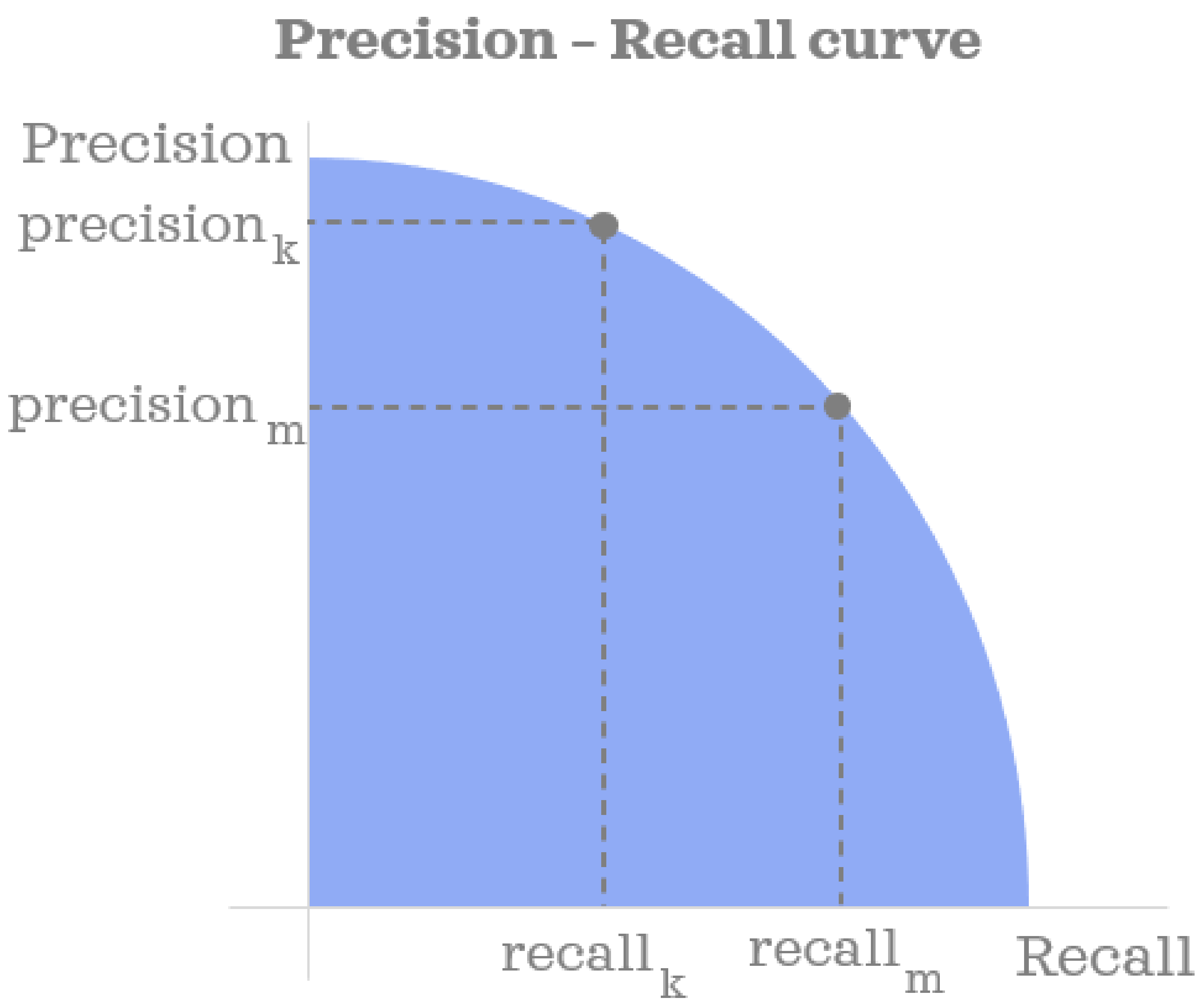
		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

2. Metrics

AP (Average Precision)



2. Metrics

mAP (mean Average Precision)

pyimagesearch

**Mean Average Precision (mAP)
Using the COCO Evaluator**

$$mAP = \frac{1}{|classes|} \sum_{c \in classes} \frac{|TP_c|}{|FP_c| + |TP_c|}$$

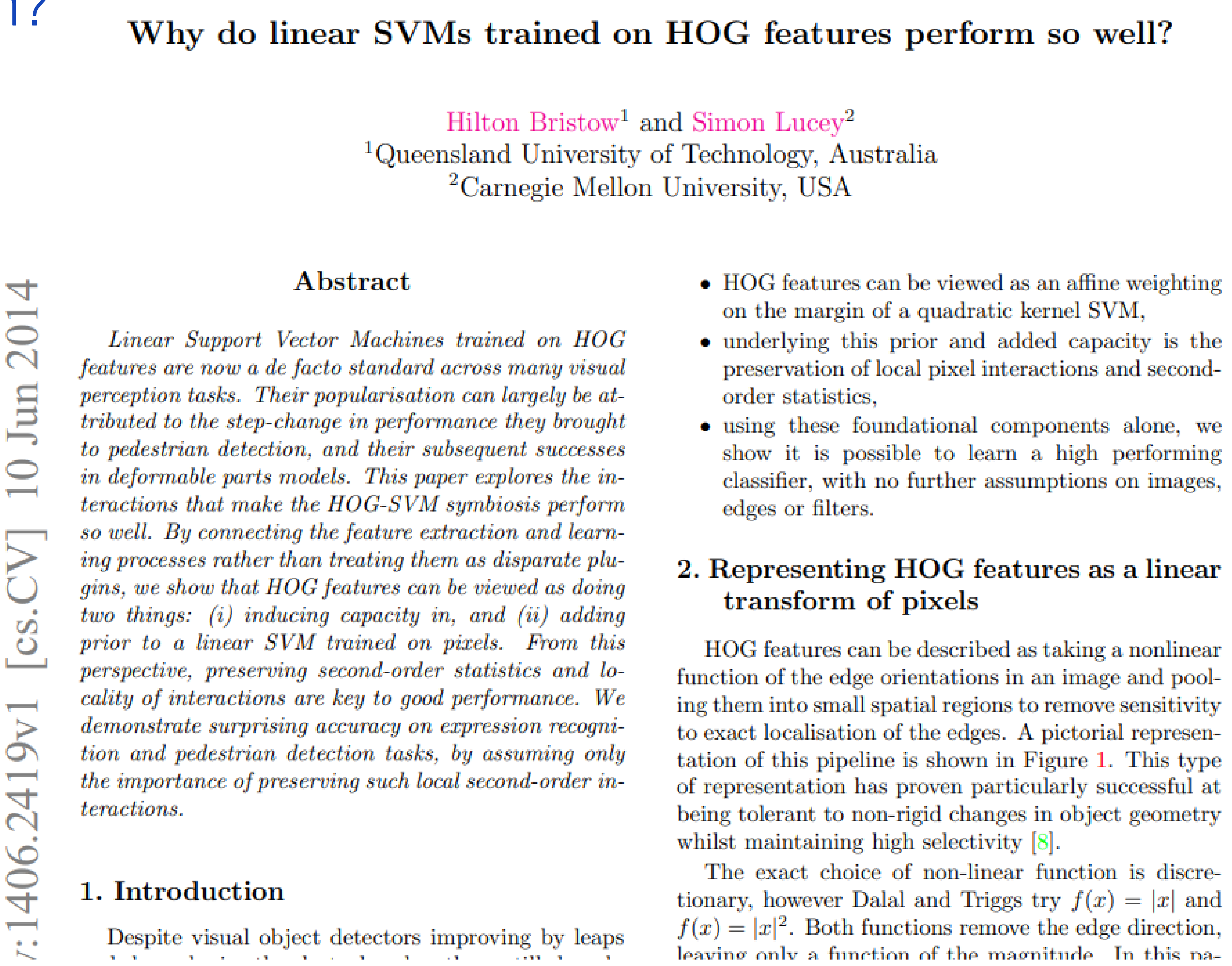
HOG + SVM

Why using HOG + SVM, Introduction, Implementation, Evaluation, Demo

3. HOG + SVM

Why using HOG+SVM for pedestrian detection?

- a commonly used method
- one of the most basic algorithm for detecting pedestrians



Why do linear SVMs trained on HOG features perform so well?

Hilton Bristow[1] and Simon Lucey[2]
[1]Queensland University of Technology, Australia
[2]Carnegie Mellon University, USA

:1406.2419v1 [cs.CV] 10 Jun 2014

Abstract

Linear Support Vector Machines trained on HOG features are now a de facto standard across many visual perception tasks. Their popularisation can largely be attributed to the step-change in performance they brought to pedestrian detection, and their subsequent successes in deformable parts models. This paper explores the interactions that make the HOG-SVM symbiosis perform so well. By connecting the feature extraction and learning processes rather than treating them as disparate plugins, we show that HOG features can be viewed as doing two things: (i) inducing capacity in, and (ii) adding prior to a linear SVM trained on pixels. From this perspective, preserving second-order statistics and locality of interactions are key to good performance. We demonstrate surprising accuracy on expression recognition and pedestrian detection tasks, by assuming only the importance of preserving such local second-order interactions.

1. Introduction

Despite visual object detectors improving by leaps

- HOG features can be viewed as an affine weighting on the margin of a quadratic kernel SVM,
- underlying this prior and added capacity is the preservation of local pixel interactions and second-order statistics,
- using these foundational components alone, we show it is possible to learn a high performing classifier, with no further assumptions on images, edges or filters.

2. Representing HOG features as a linear transform of pixels

HOG features can be described as taking a nonlinear function of the edge orientations in an image and pooling them into small spatial regions to remove sensitivity to exact localisation of the edges. A pictorial representation of this pipeline is shown in Figure 1. This type of representation has proven particularly successful at being tolerant to non-rigid changes in object geometry whilst maintaining high selectivity [8].

The exact choice of non-linear function is discretionary, however Dalal and Triggs try $f(x) = |x|$ and $f(x) = |x|^2$. Both functions remove the edge direction, leaving only a function of the magnitude. In this pa-

Link: https://arxiv.org/pdf/1406.2419.pdf

3. HOG + SVM

Scientific article

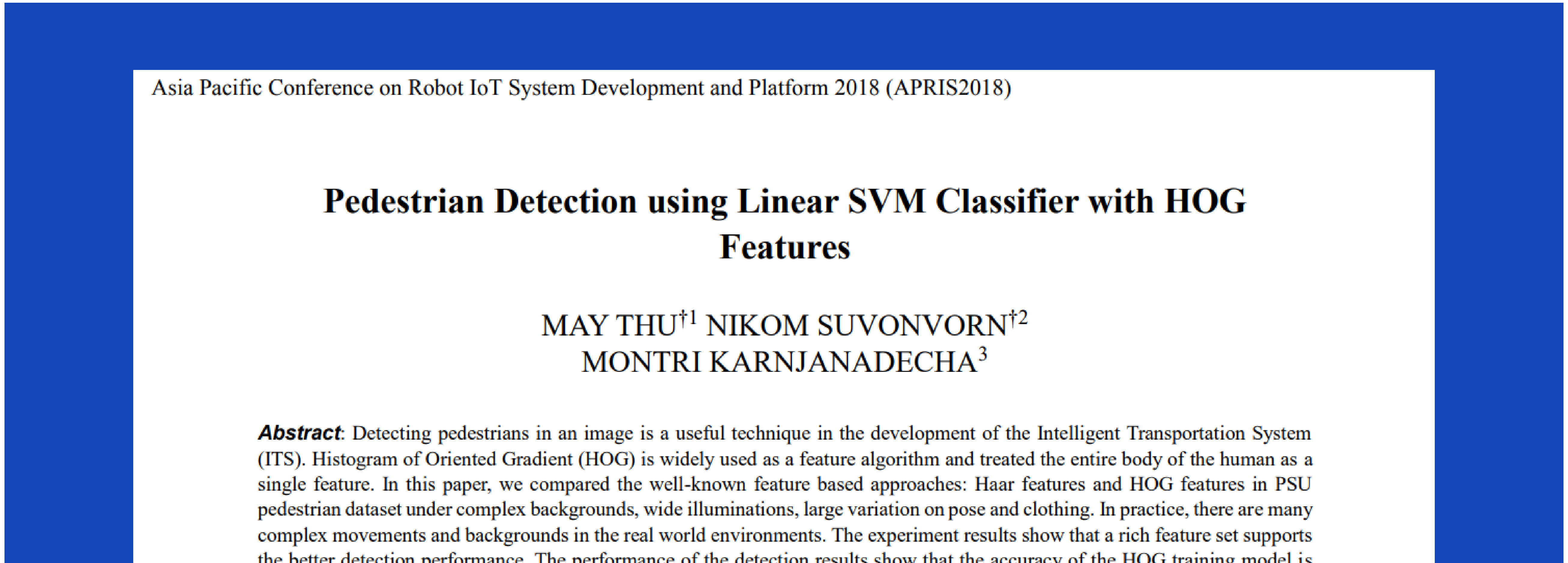
Pedestrian Detection using Linear SVM Classifier with HOG Features

Author: May Thu et al.

Date: 2/12/2018

At: Asia Pacific Conference on Robot IoT System Development and Platform 2018 (APRIS2018)

Link: https://ipsj.ixsq.nii.ac.jp/ej/?action=repository_uri&item_id=192976&file_id=1&file_no=1



Asia Pacific Conference on Robot IoT System Development and Platform 2018 (APRIS2018)

Pedestrian Detection using Linear SVM Classifier with HOG Features

MAY THU[†1] NIKOM SUVONVORN[†2]
MONTRI KARNJANADECHA[3]

Abstract: Detecting pedestrians in an image is a useful technique in the development of the Intelligent Transportation System (ITS). Histogram of Oriented Gradient (HOG) is widely used as a feature algorithm and treated the entire body of the human as a single feature. In this paper, we compared the well-known feature based approaches: Haar features and HOG features in PSU pedestrian dataset under complex backgrounds, wide illuminations, large variation on pose and clothing. In practice, there are many complex movements and backgrounds in the real world environments. The experiment results show that a rich feature set supports the better detection performance. The performance of the detection results show that the accuracy of the HOG training model is

3. HOG + SVM

Implementation

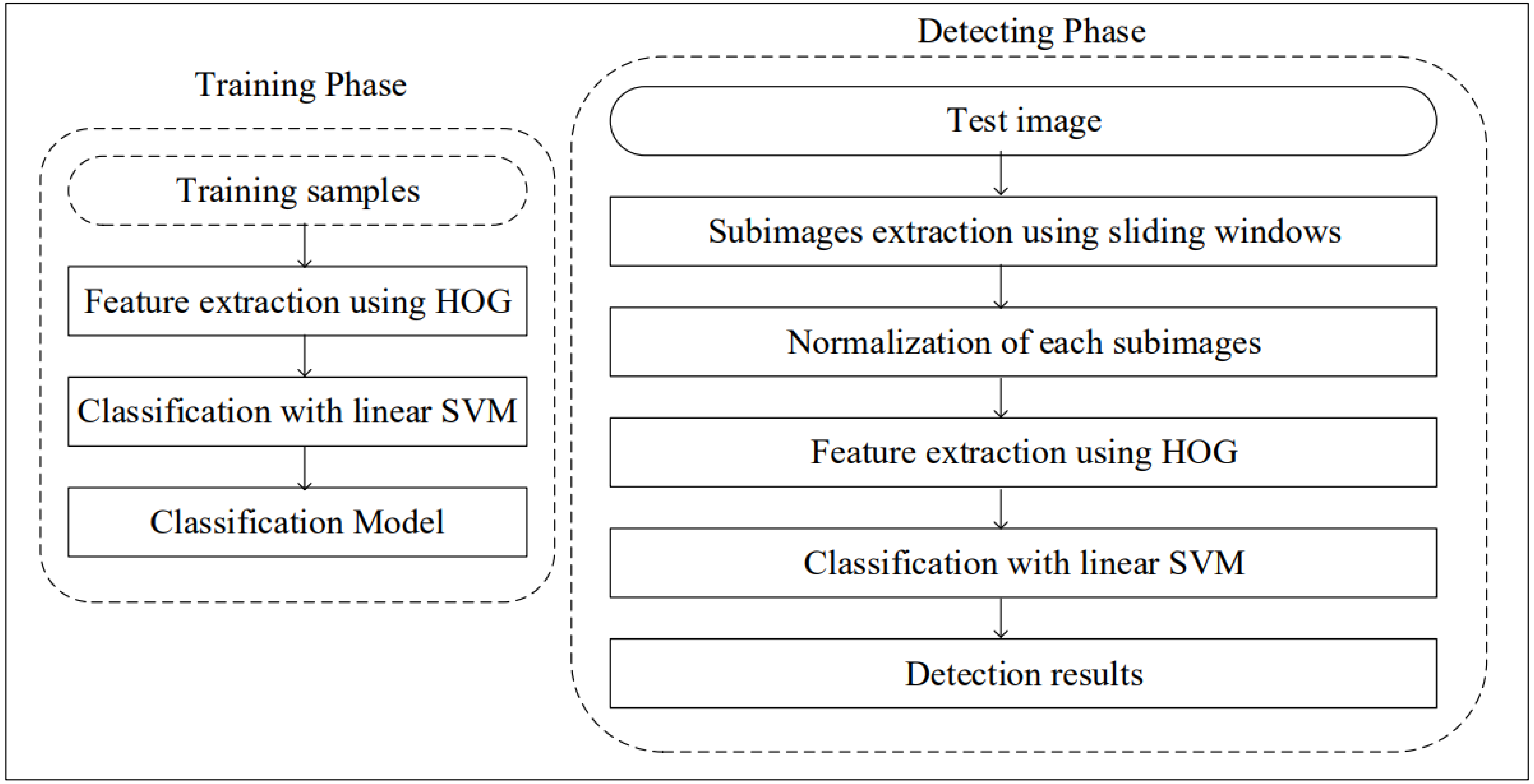


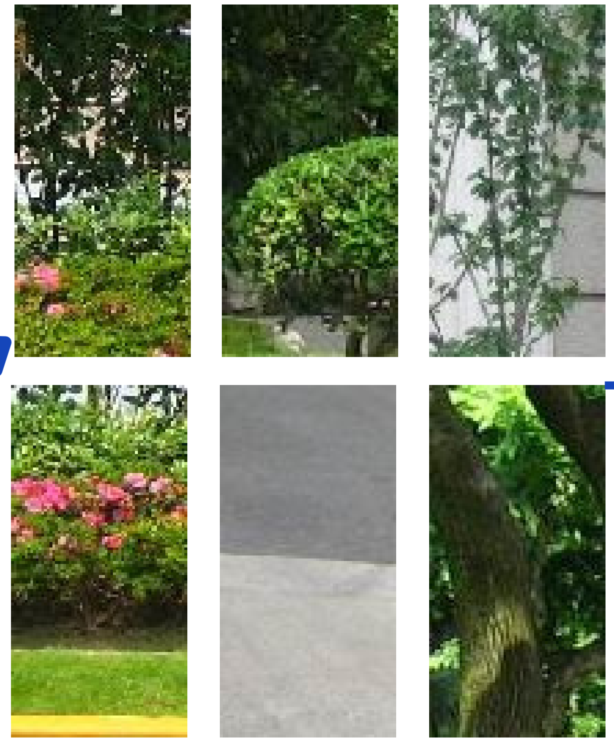
Fig 1. Detection model of the pedestrian detection system

3. HOG + SVM

Implementation

Training phase

800 negative samples



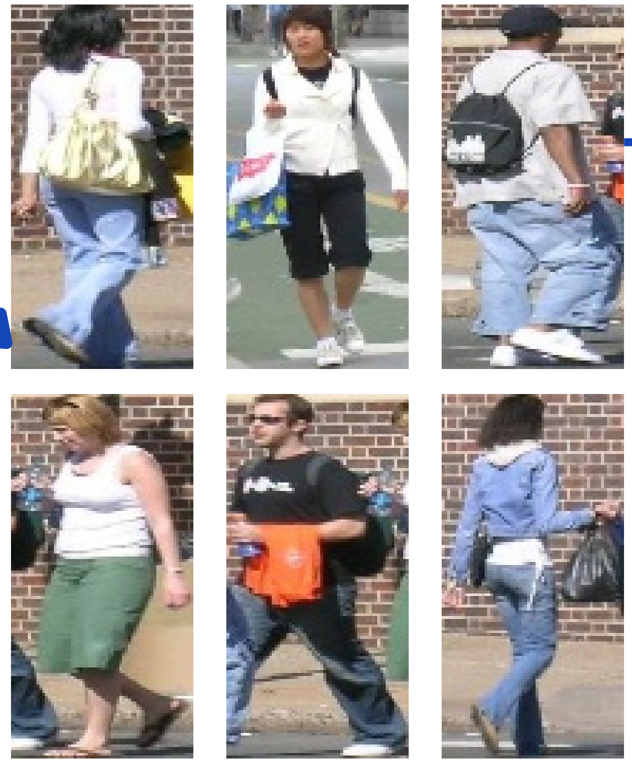
- use the sliding window to crop the image and get the background

0

Dataset Labeling

1

345 positive samples



- used annotation to crop bounding box

3. HOG + SVM

Implementation

Training phase

- Compute the HOG features.
- Combine and split the positive and negative labeled HOG features with the train set : 915 images, and the test set : 230 images
- Train the SVM classifier => model trained
- Test model through the test data.

3. HOG + SVM

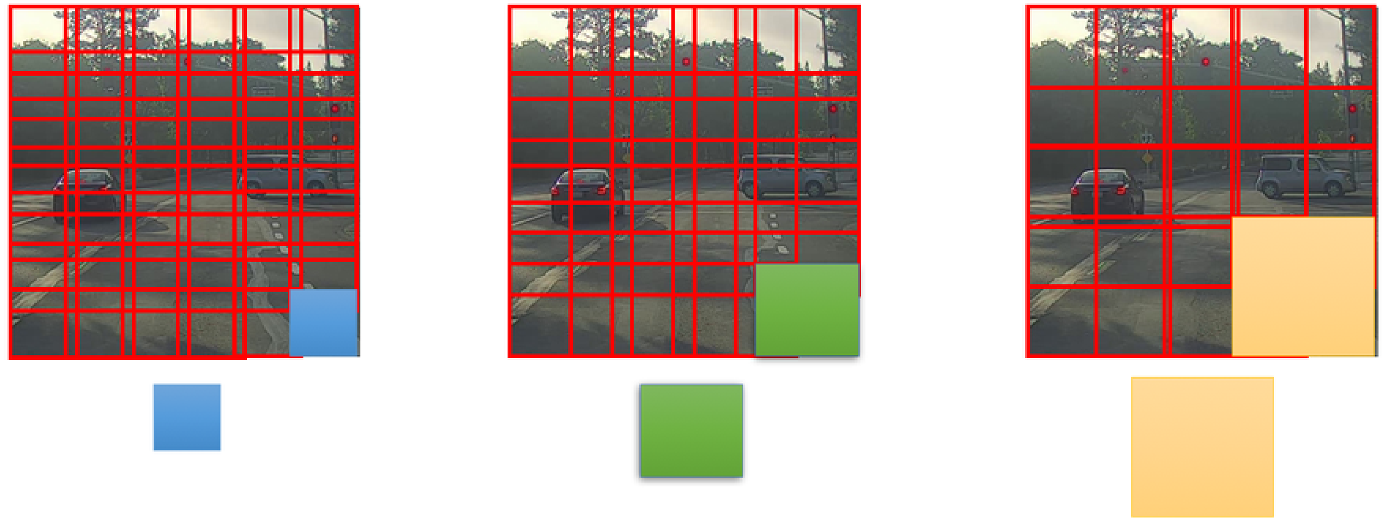
Implementation

Detecting phase

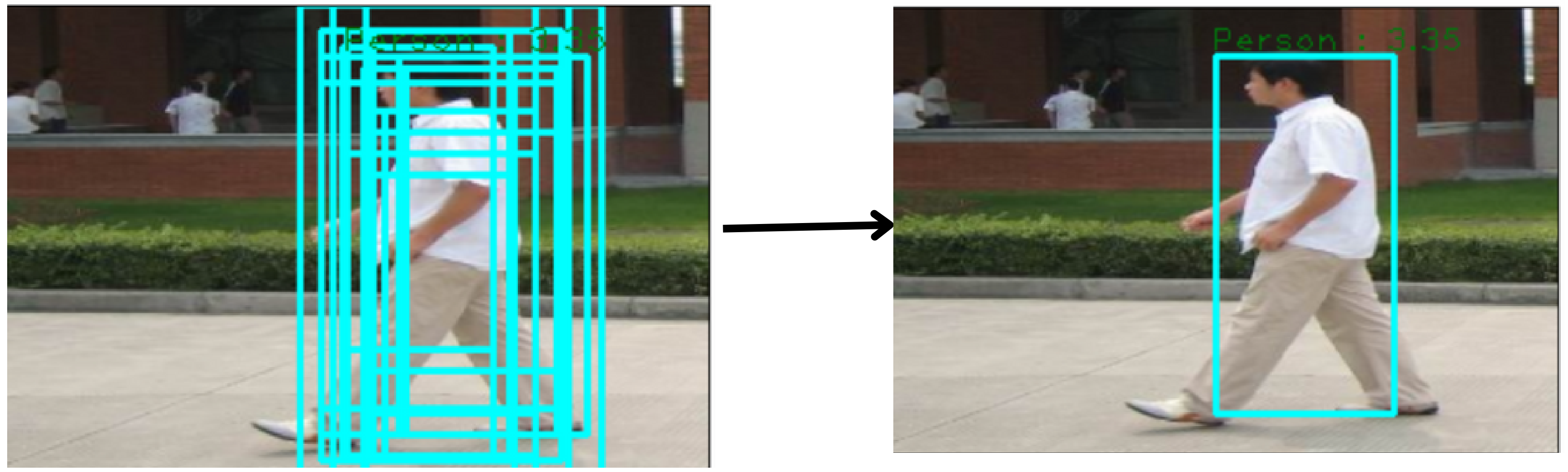
Testing and Detection:

- Extract HOG features from a test image using a sliding window technique.
- Apply non-maximum suppression to remove duplicate detections.
- Draw bounding boxes around the detected pedestrian regions.

Sliding window

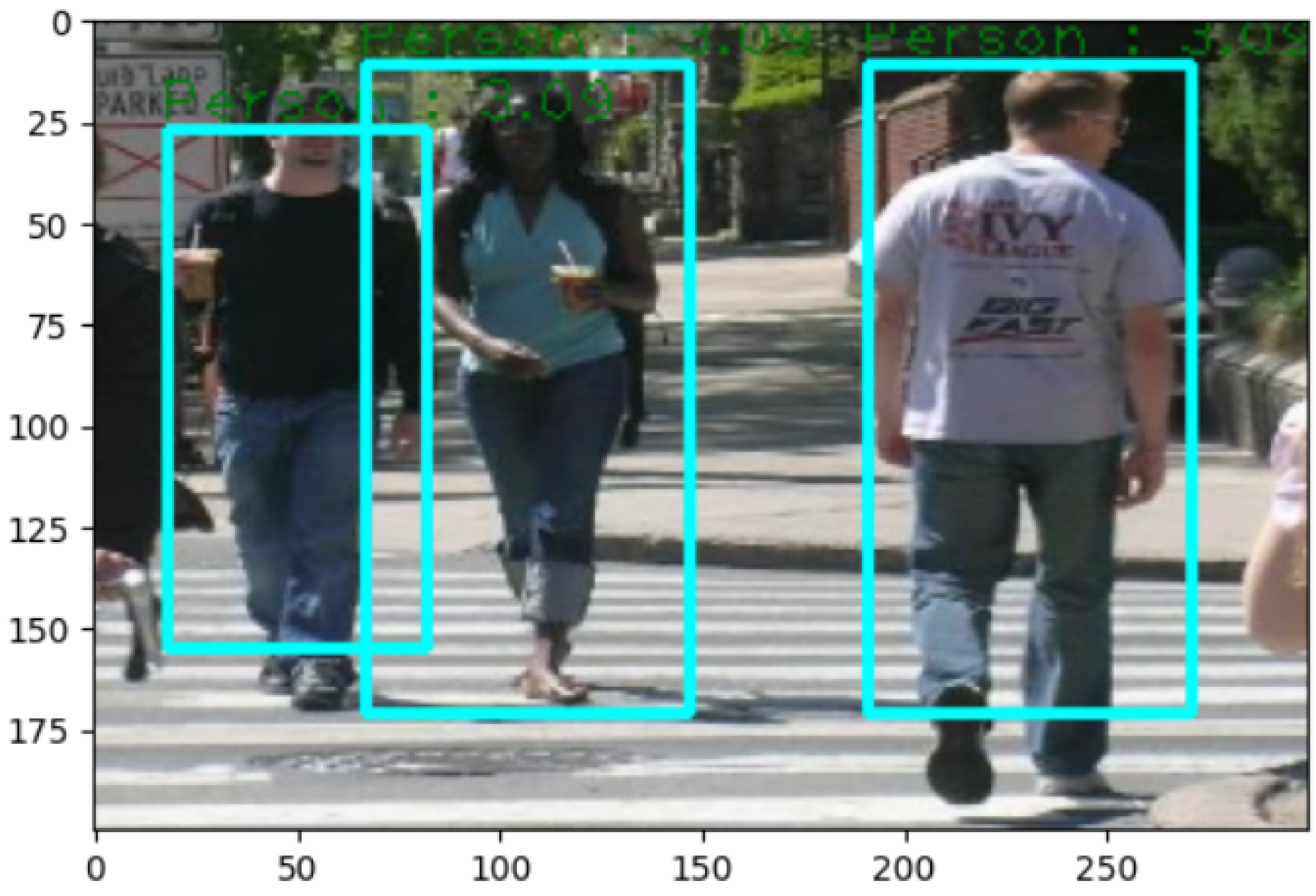


Non-maximum suppression



3. HOG + SVM

Demo



3. HOG + SVM

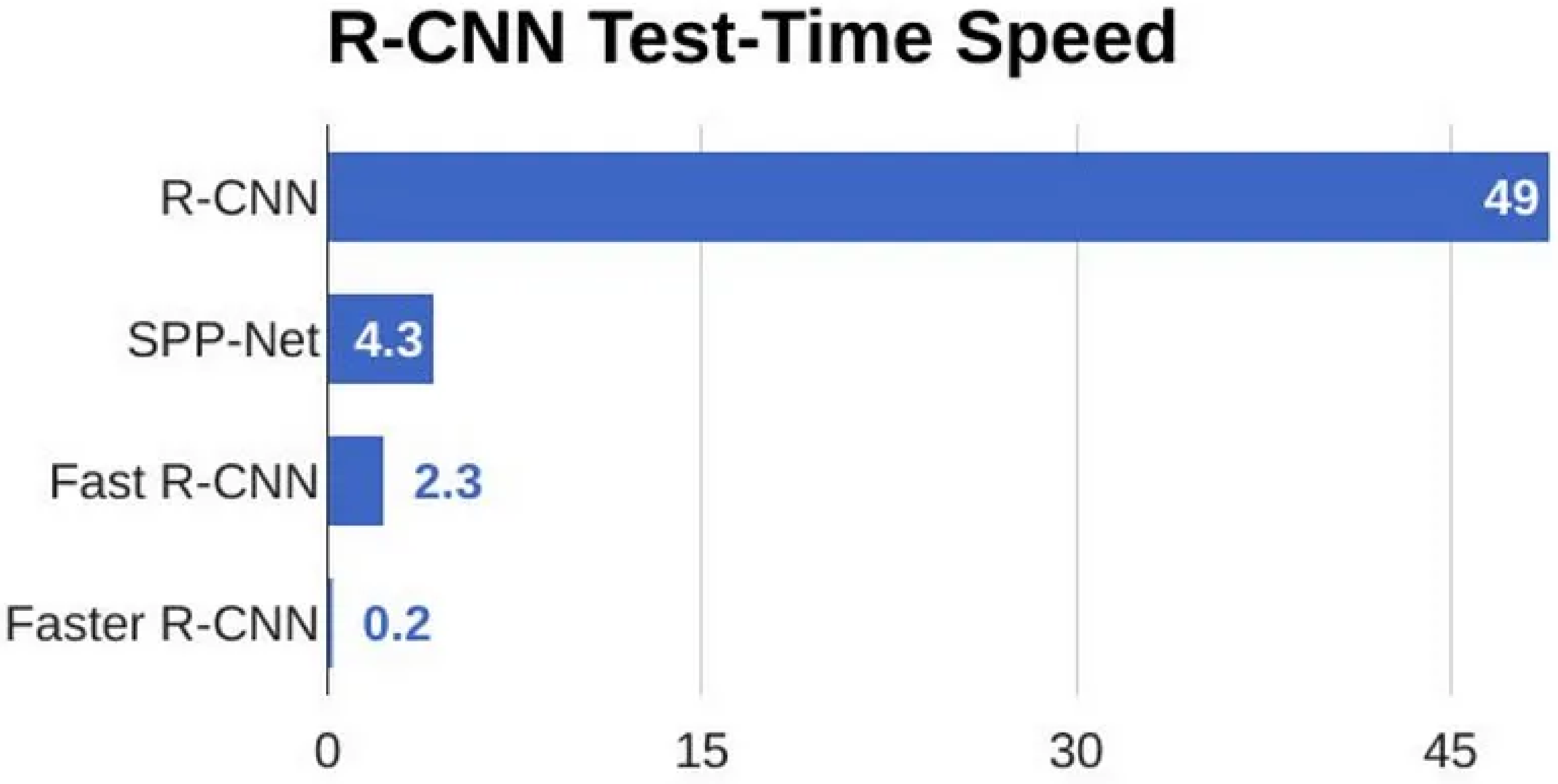
Demo

Faster R-CNN

Introduction, Why using Faster RCNN, Implementation, Demo

4. Faster R-CNN

Why choosing Faster R-CNN



http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

4. Faster R-CNN

Faster Region Convolutional Network

Paper: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Author: Shaoqing Ren et al.

Link: https://arxiv.org/pdf/1506.01497.pdf

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun

Abstract—State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet [1] and Fast R-CNN [2] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a *Region Proposal Network* (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully convolutional network that simultaneously predicts object bounds and objectness scores at each position. The RPN is trained end-to-end to generate high-quality region proposals, which are used by Fast R-CNN for detection. We further merge RPN and Fast R-CNN

4. Faster R-CNN

Introduction

Faster R-CNN

Region Proposal Network

Fast R-CNN

4. Faster R-CNN

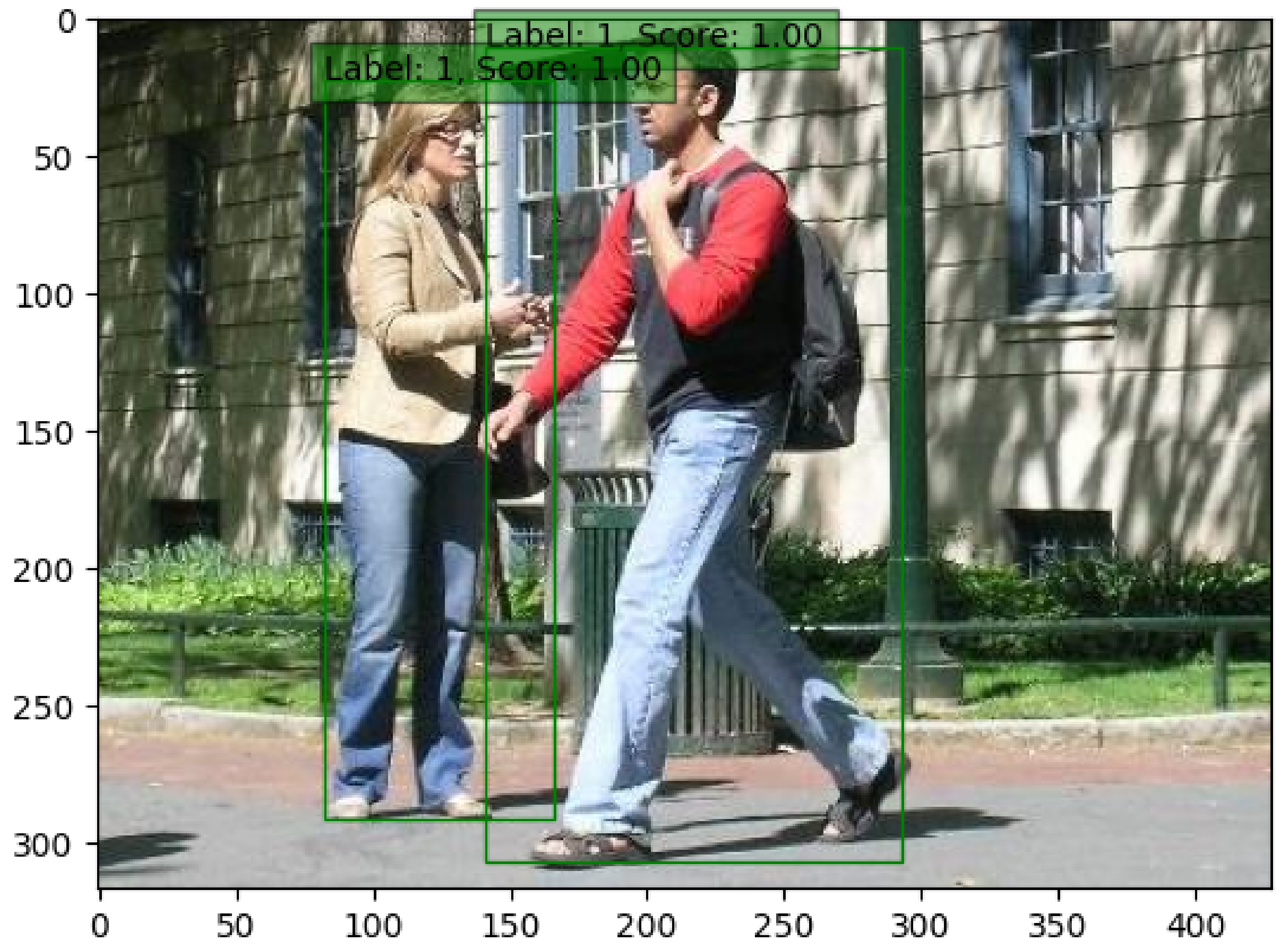
Implementation

PyTorch

Google colaboratory

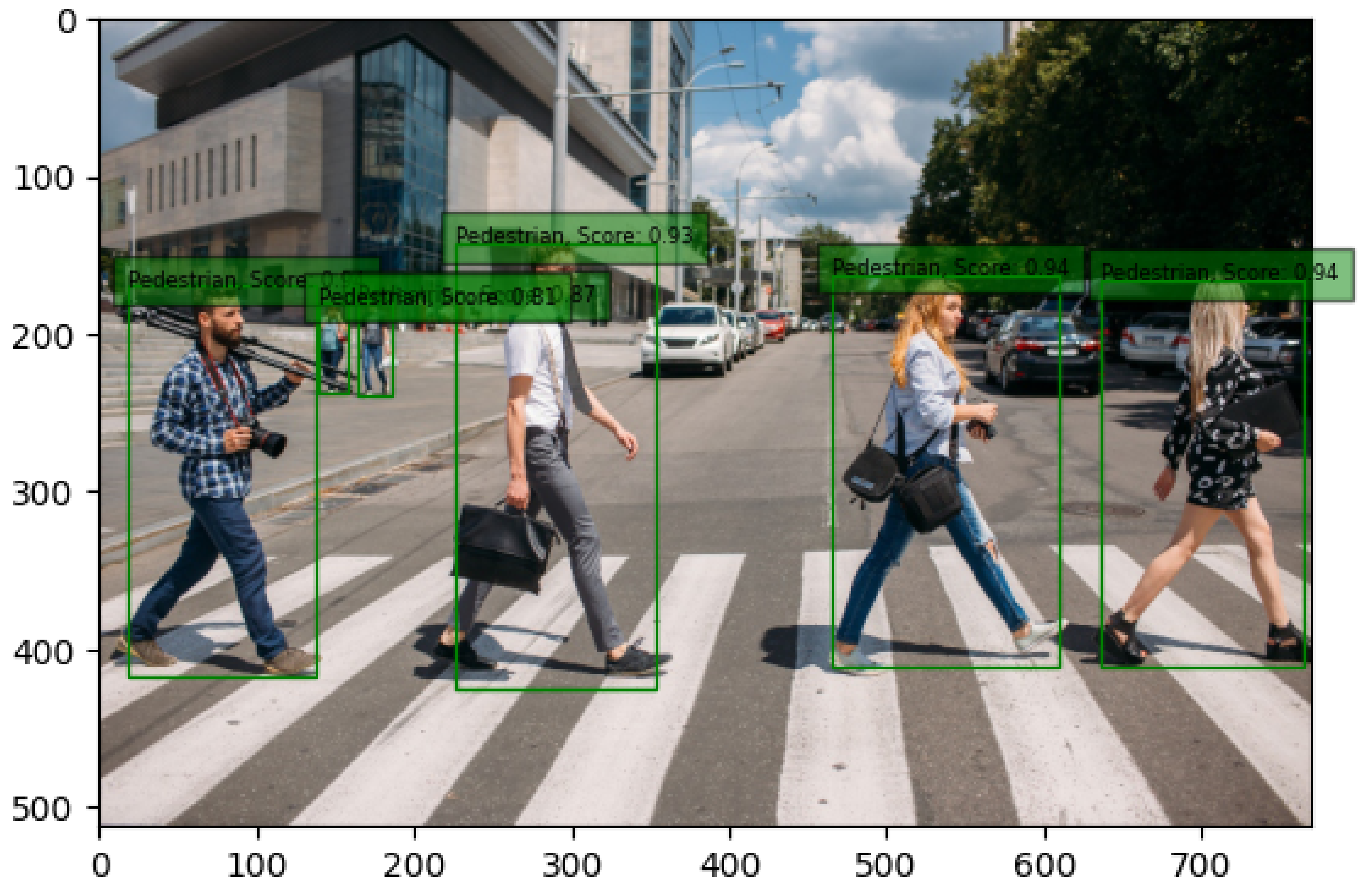
4. Faster R-CNN

Test result



4. Faster R-CNN

Test result



Evaluation

	SVM + HOG	Faster R-CNN lr=0.004	Faster R-CNN lr=0.005	Faster R-CNN lr=0.006
mAP threshold= 0.5	0.271257	0.591090	0.604074	0.618742

References

https://www.cis.upenn.edu/~jshi/ped_html/

https://ipsj.ixsq.nii.ac.jp/ej/action=repository_uri&item_id=192976&file_id=1&file_no=1

https://arxiv.org/pdf/1406.2419.pdf

https://arxiv.org/pdf/1506.01497.pdf

http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf

https://arxiv.org/abs/1607.07032

https://pyimagesearch.com/2022/05/02/mean-average-precision-map-using-the-coco-evaluator/

http://www.researchgate.net/publication/224364841_Pedestrian_Detection_Using_Boosted_HOG_Features

Thank you for listening

4. Faster R-CNN

But...

Is Faster R-CNN Doing Well for Pedestrian Detection?

Liliang Zhang[1] Liang Lin[1*] Xiaodan Liang[1] Kaiming He[2]

[1]School of Data and Computer Science, Sun Yat-sen University
zhangll.level0@gmail.com; linliang@ieee.org; xdliang328@gmail.com
[2]Microsoft Research
kahe@microsoft.com

Abstract. Detecting pedestrian has been arguably addressed as a special topic beyond general object detection. Although recent deep learning object detectors such as Fast/Faster R-CNN [1,2] have shown excellent performance for general object detection, they have limited success for detecting pedestrian, and previous leading pedestrian detectors were in general hybrid methods combining hand-crafted and deep convolutional