Towards Autonomous Training of Scene-Specific Pedestrian Detectors in Visual Surveillance Environments

Mohammed Shahnewaz Chowdhury

A thesis submitted in total fulfilment of the requirements for the degree of
DOCTOR OF PHILOSOPHY

Department of Electrical and Computer Systems Engineering
FACULTY OF ENGINEERING
MONASH UNIVERSITY

OCTOBER 2018

Copyright Notices

Notice 1

Notice 2

General Declaration

This thesis contains no material which has been accepted for the award of any other degree or diploma at any university or equivalent institution and that, to the best of my knowledge and belief, this thesis contains no material previously published or written by another person, except where due reference is made in the text of the thesis.

Mohammed Shahnewaz Chowdhury
October 2018

Table of Contents

Abstract

Intelligent visual surveillance (IVS) is one of the foremost practical applications of the actively researched field of pedestrian detection. In conventional IVS environments, pedestrian detection is of paramount importance and has been extensively studied for over two decades. Unfortunately, despite seemingly tremendous advancements, the performance of generic pedestrian detectors trained on publicly available datasets is still plagued by the dataset shift complication, whereby, the distribution of the test data in unseen, target surveillance scenes often differs significantly from that of the source training data. Recently, this problem has been considerably alleviated by the development of scene-specific training algorithms, which handle each unseen scene independently and generate a pedestrian detector specific to that scene by collecting and utilizing target samples. However, there are serious practical limitations - some of these algorithms require manual labelling of the target samples, consequently compromising scalability, while others are dependent on a pre-trained generic detector for the acquisition of target samples, rendering them inapplicable in complex environments.

The aim of this research is to develop a training framework that addresses the two aforementioned limitations, but through a paradigm shift from existing works. Rather than developing new detectors or adaptation approaches, the focus is transferred to the exclusive exploitation of target samples only, in an autonomous and practical way. Concretely, a Virtually Autonomous Training (VAT) framework is developed that trains scene-specific pedestrian detectors for unseen target surveillance environments with zero manual labelling of target samples, but simultaneously, does not utilize any source dataset or pre-trained generic detectors. To achieve automatic labelling of target samples, oracles are designed that evaluate miscellaneous generic attributes prevalent in pedestrians and exploit the resultant distributions to segregate pedestrians from non-pedestrians. By integrating oracles within the framework, VAT executes a sequence of carefully designed training stages that maximizes the exploitation of target samples and progressively improves the classifier to ultimately generate the scene-specific pedestrian detector.

The proposed VAT was extensively tested on a large number of video surveillance datasets with varying levels of difficulty. To demonstrate the flexibility of VAT, it was implemented with both popular real-time classifiers – support vector machines and adaptive boosting, on each dataset. Our designed oracles labelled pedestrians correctly with an average precision of above 90 % across all datasets, indicating their high reliability in automatic labelling of target samples in different surveillance environments. On fairly difficult datasets, such as PETS, VAT achieves similar or better performance than scene-specific training approaches that depend on generic detectors. However, for more complex, unconstrained datasets, such as QMUL-J, QMUL-R and KWSI, VAT outperforms them by substantial margins of up

to 40%. Compared to the state-of-the-art, VAT achieves amongst the best performances on the two most commonly used datasets, CUHK and MIT, for evaluating scene-specific training approaches. Specifically, on CUHK, VAT achieves a detection rate of 74.6 %, which is only 0.9 % behind the top-performing scene-specific training approach, but on MIT, VAT achieves a detection rate of 86.7% which is not only 9.7% higher than the second best scene-specific training approach, but is remarkably the highest ever reached on the MIT dataset.

Acknowledgments

I would like to express my deepest gratitude to ELID SDN BHD for funding my postgraduate studies and allowing me to pursue this PhD. Without their continued financial support, this research would not have been possible.

I would like to offer my sincerest thanks to my supervisors: Associate Professor Poh Phaik Eong and Associate Professor Kuang Ye Chow. I would particularly like to show my utmost appreciation to Associate Professor Kuang Ye Chow for the truly unconditional patience he has shown me, despite the numerous setbacks I have faced over the course of this study. Without his support and guidance, the completion of this thesis would not have been attainable.

I would like to thank all my family members – my beloved mother, my two beautiful little sisters, my younger brothers and my elder brother and his family. Their unwavering love, support and confidence in me has given me the mental fortitude to continue moving forward through all the trials and tribulations.

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Description
Acc	Accuracy
ACF	Aggregate Channel Features
AdaBoost	Adaptive Boosting
ADCNN	Adaptive Deep Convolutional Neural Networks
AGPD	Adaptation of Generic Pedestrian Detector
CBA	Class Biased Attribute
CESVM	Confidence Encoded Support Vector Machine
CNN	Convolutional Neural Networks
CNNDAC	Convolutional Neural Network with Dynamically Adjusted Classifier
CovBoost	Covariance Boosting
CPD	Cognitive Pedestrian Detector
DAE	Direct Attribute Evaluation
DET	Detection Error Trade-off
DPM	Deformable Part-based Models
ENIDA	Efficient Non Iterative Domain Adaptation
FAT	Fully Autonomous Training
FN	False Negatives
ForeGd	Foreground Training Sample Filter
FP	False Positives
FPPI	False Positives Per Image
GraySc	Grayscale Training Sample Filter
HN	Hard Negatives
HOG	Histogram of Oriented Gradients
ICF	Integral Channel Features
IN	Initial Negatives
IP	Internet Protocol
ITS	Intelligent Transportation Systems
IVS	Intelligent Visual Surveillance
LBP	Local Binary Patterns
LBP	Local Binary Patterns
MIL	Multiple Instance Learning
MLCNN	Multi-stage Learning of Convolutional Neural Network
MskGrad	Masked Gradient Training Sample Filter
NS	No Source
PCA	Principal Component Analysis
PLM	Progressive Latent Model
Pre	Precision

R-CNN	Region-Convolutional Neural Network
Rec	Recall
ROC	Receiver Operating Characteristic
RP	Rejected Positives
SA	Scale Aware
SIFT	Scale Invariant Feature Transform
SS	Semi-Supervised
SVM	Support Vector Machine
Temp	Template matching Training Sample Filter
TN	True Negatives
TP	True Positives
TSF	Training Sample Filter
TSF-Con	TSF Contribution
TSF-Pre	TSF Precision
UOLF	Unsupervised Object Learning Framework
VA	Video Analytics
VAT	Virtually Autonomous Training
VCA	Video Content Analysis
VerStruct	Vertical Structures Training Sample Filter
VMS	Video Management Software
VSaaS	Video Surveillance as a Service

Primary Notations and Nomenclature

As the core of this thesis involves exploitation of training samples, the symbols, *x* and $\mathbb{X}$, will appear frequently, with superscripts and/or subscripts that have various meanings depending on the context, as explained next. Throughout this thesis, a training sample is denoted by *x*, in italics. It is important to remember its difference from 'x' to avoid confusion. The normal letter, x, refers to the x-coordinate of a point, for example, (x1,y1) and (x2,y2) denote the co-ordinates of points 1 and 2 respectively. The symbol, $\mathbb{X}$, represents the set of all training samples obtained from a particular domain. In this thesis, the domain is a visual surveillance environment/scene, denoted by ϵ. Hence, $\mathbb{X}^{\epsilon}$ represents the set of all training samples from ϵ. Given an arbitrary object class of interest, denoted by $\mathcal{C}$ (in this research, $\mathcal{C}$ = pedestrian) $\mathbb{X}^{+}$ denotes the set of training samples that belongs to $\mathcal{C}$ and $\mathbb{X}^{-}$ denotes the set of training samples that does not belong to $\mathcal{C}$. Therefore, $\mathbb{X}^{\epsilon} = \mathbb{X}^{+} \cup \mathbb{X}^{-}$. A single training sample is denoted by x_i, where, *i* is the index of the training sample. If the training sample is obtained from ϵ, it is denoted by x_i^{ϵ}. So if *N* training samples are obtained from ϵ, they are represented by $\{x_i^{\epsilon}\}_{i=1}^{N}$ and $\mathbb{X}^{\epsilon} = \{x_i^{\epsilon}\}_{i=1}^{N}$.

An oracle is an automatic labeller denoted by $\mathbb{X}$, which is a hierarchical combination of training sample filters and pruners. Concretely, an oracle is composed of sequentially arranged training sample filters, denoted by T and each training sample filter can be composed of one or more pruners, denoted by ρ. T_j is the j-th training sample filter in the oracle and ρ_{jk} is the k-th pruner in the j-th training sample filter. The subscript of $\mathbb{X}$ indicates the level of the oracle under consideration. So, $\mathbb{X}_{\mathbb{X}}^{+}/\mathbb{X}_{\mathbb{X}}^{-}$, $\mathbb{X}_{\mathrm{T}}^{+}/\mathbb{X}_{\mathrm{T}}^{-}$ and $\mathbb{X}_{\rho}^{+}/\mathbb{X}_{\rho}^{-}$ represent the set of training samples labelled at the oracle, training sample filter and pruner levels, respectively, as belonging/not belonging to $\mathcal{C}$. Correspondingly, $\mathbb{X}_{\mathrm{T}_j}^{+}/\mathbb{X}_{\mathrm{T}_j}^{-}$ denote the training samples labelled by the j-th training sample filter and $\mathbb{X}_{\rho_{jk}}^{+}/\mathbb{X}_{\rho_{jk}}^{-}$ denote the training samples labelled by the k-th pruner in the j-th training sample filter.

For clarity and brevity, the primary nomenclature utilized in this thesis is listed in the following table.

Symbol	Meaning
x_i	i-th training sample
y_i	class label of the i-th training sample
$\mathbb{C}$	Object class of interest ($\mathbb{C}$= pedestrian in this thesis)
$\mathbb{X}$	Set of training samples from a particular domain
ϵ	A surveillance environment or scene
f	An extracted feature representation of a training sample
φ	A DAE function that computes the CBA of a training sample based on f
z_i	An output score from the executed DAE function on the i-th sample
N	Number of training samples passed to a particular pruner
Λ	A rejection function that segregates pedestrian from non-pedestrian instances based on the distribution of the output scores
$\mathbb{K}^t$	t-th oracle
T_j	j-the training sample filter in an oracle
ρ_{jk}	k-th pruner of the j-th training sample filter in an oracle
$\mathbb{E}_i$	i-th frame of video sequence from environment ϵ
Ω	Sample extractor based on background subtraction
ar	Aspect ratio of acquired potential pedestrian sample
$\mathcal{M}$	Potential pedestrian samples extracted by Ω
μ	Mean
σ	Standard deviation
$\mathbb{E}_i^{-\mathrm{ve}}$	i-th frame for collecting non-pedestrian instances
(r, s, fr)	(Bootstrapping rounds, bootstrapping segments per round, bootstrapping frames per segment)
$\Re$	Set of detection responses obtained by applying a detector to a video sequence from environment ϵ
$Pos_{\mathbb{K}^t}$	Samples labelled by $\mathbb{K}^t$ as pedestrian instances
$Neg_{\mathbb{K}^t}$	Samples labelled by $\mathbb{K}^t$ as non-pedestrian instances
Đ	Training Dataset
Ξ	Supervised classifier learning – AdaBoost/SVM
Ж	Pedestrian detector trained using classifier learning, Ξ, on training dataset, Đ
§	Scene-specific

1 Introduction

1.1 Intelligent visual surveillance (IVS) – State of market

The world's leading market research organizations estimate that the global video surveillance market will account for over USD 60 billion by the year 2023 [1, 2]. Compared to fewer than 10 million in 2006, approximately 130 million surveillance cameras will be shipped worldwide in 2018 [3] . In china alone, where surveillance networks are expanding more rapidly than anywhere else in the world, there were 170 million surveillance cameras in operation at the end of 2017; it has been reported that this number is expected to skyrocket to 626 million by 2020 [4] .

The aforementioned figures are an indisputable testament to the magnitude of demand for video surveillance, but what is causing such unprecedented market escalation? The video surveillance industry is a complex ecosystem of cameras, storage, software and miscellaneous hardware, and it is the technological innovations across every sector of this ecosystem that is propelling market growth. Major factors include the shifting preference from analogue to Internet Protocol (IP) cameras, enhanced video compression standards like H.265 enabling more efficient data storage, cloud-based video surveillance called Video Surveillance as a Service (VSaaS) and increasingly sophisticated Video Management Software (VMS) [5]. However, one of the most significant drivers is the advent of Video Analytics (VA) - according to *The Video Surveillance Report 2018* [6], 68% of the surveyed industry professionals are either already using VA or intend to do so in the coming years and 38% responded that VA is likely to motivate the next upgrade to their customers' surveillance systems.

In the video surveillance domain, VA or Video Content Analysis (VCA) are commercial terms for Intelligent Visual Surveillance (IVS), and video surveillance itself is used interchangeably with visual surveillance. To comprehend the need for IVS, it is important to first comprehend the limitations in the absence of IVS. Traditional visual surveillance systems require human operators in a control room monitoring live video feeds from surveillance cameras to spot any suspicious/dangerous activity. As the rising concerns for public safety coupled with the decreasing prices of surveillance equipment cause the

Figure 1.1: A typical surveillance control room

number of installed cameras to rapidly proliferate, the overwhelmingly large ratio of cameras to operators makes the task of handling the video information overload increasingly inefficient and impractical [7] (see Figure 1.1). This shortcoming is compounded by the fact that anomalous events occur infrequently, necessitating continuous observation but human focus degrades to an inadequate level after only 20 minutes of watching video monitors [8] . Inevitably, a significant portion of the video channels are usually not monitored, meaning potentially critical events go undetected. In the end, the surveillance system becomes undesirably passive, and large amounts of video footage are merely archived to be utilized retrospectively as a forensic tool.

The ideal goal of visual surveillance should not be to just place cameras in the place of human eyes but to also embed "intelligence" for automating tasks that usually require considerable human efforts. The current massive interest in the video surveillance industry to achieve this goal is motivated by the substantial potential benefits such as time and cost savings, efficient security solutions that are capable of taking preventative action rather than simply reacting after an event and creation of business value through intelligent use of metadata. This has led IVS to become an intensively researched [9-13] multi-disciplinary field involving the topics of computer vision, pattern analysis, signal processing, image sensors and artificial intelligence. As defined by Elliott [14], IVS is generally "any video surveillance solution that utilizes technology to automatically, without human intervention, process, manipulate and/or perform actions to or because of either live or stored video images."

The plethora of developed applications of IVS include, but are not limited to:

- motion detection to trigger a recording or sound alarms due to movement in restricted areas, like intrusion detection or line crossing [15]
- abandoned or removed object detection to prevent terrorist attacks [16]
- license plate recognition and vehicle detection, tracking and recognition for various purposes such as electronic toll collection, security monitoring, traffic flow analysis and illegal activity detection in Intelligent Transportation Systems (ITS) [17]
- various customer analytics for business intelligence in retail, such as people counting, dwell time monitoring and queue management [18]
- biometrics such as face or iris recognition and gait analysis to identify and track criminals/suspicious individuals or grant authorized access [19]
- detection of anomalous/suspicious behaviour such as person falling [20] , loitering [21], or following/chasing [22]
- detection of aggressive/violent behaviour such as kicking, punching or fighting [23]
- crowd analysis such as flux statistics, congestion analysis and anomaly detection [24]

IVS is applicable to an exhaustive array of visual surveillance environments ranging from private properties like apartments, condominiums, offices and factories to high security areas like banks, airports, prisons, casinos and ATMs to public spaces like shopping malls, museums, parking lots, stores, hospitals, educational institutions, government buildings and public transport to transit scenes like bus stations, railway stations, petrol stations, subways, highways, elevators and traffic intersections. In other words, IVS has penetrated every major industry vertical, including commercial, industrial, infrastructure, residential, institutional, financial, healthcare, government/public and transportation [25]. As IVS becomes increasingly pervasive, it is very important to realize that the motivation of developing such technology should not be the replacement of human operators with IVS systems, but to supplement a layer of intelligence in visual surveillance to assist the human operators so that, instead of performing routine and laborious tasks, they can engage their cognitive powers to make timely, higher-level decisions on how to deal with the incidents reported by IVS.

1.2 Pedestrian detection in IVS

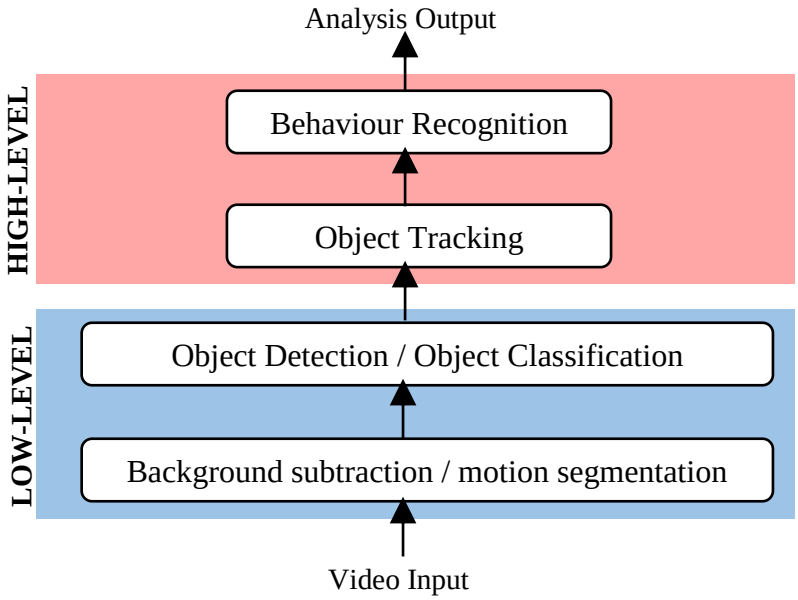


Figure 1.2: A standard IVS system

An IVS system usually consists of low-level and high-level automated processes that execute in a sequential manner [26] (see Figure 1.2). First, background subtraction (or motion segmentation) [27] is applied to the video input to extract moving regions as objects of interest. As these moving objects could belong to various categories/classes such as humans or vehicles, the next step is to correctly classify the moving objects using object detection (or object classification) [28]. Then, by finding the corresponding locations of detected objects across a finite sequence of video frames, object tracking [29] generates a spatiotemporal trajectory for every detected object. Lastly, the motion patterns in the trajectories are analysed using behaviour recognition [30] to produce high-level description of the actions and interactions that occur in the scene. This kind of sequential processing, where the output of each process is fed as input to the next, enables each subsequent process to generate increasingly informative metadata, but at the same, it makes the higher-level processes critically dependent on the performance of the lower-level processes.

Visual surveillance is synonymous with people; in most IVS environments, people or pedestrians are the primary targets under surveillance [10, 15, 21-23, 31, 32]. Pedestrian detection can be described as the task of identifying and labelling the location of pedestrians with rectangular

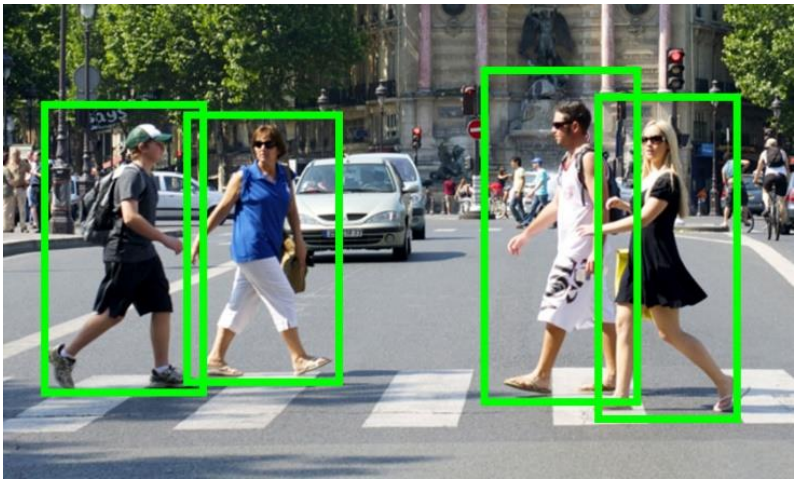


Figure 1.3: An illustration of pedestrian detection

bounding boxes (see Figure 1.3). For IVS, reliable pedestrian detection is of paramount importance because *A*) Background subtraction simply segments moving regions without providing any knowledge of whether they are actually pedestrians or not – since pedestrian detection is able to obtain class-specific metadata by localizing pedestrian instances, it acts as the initial level of intelligence in IVS and *B*) As per the preceding elaboration of Figure 1.2, pedestrian detection directly impacts the performance of subsequent high level technologies [9] that require the detection responses as inputs, such as tracking, anomaly detection, suspicious/aggressive behaviour recognition and person identification.

1.3 Focus of this thesis

In this section, the limitations of existing pedestrian detection approaches in IVS are discussed before formulating the research objectives.

1.3.1 The need for scene-specific pedestrian detectors

Pedestrian detection has been extensively studied for over two decades. It has long been well established that the best approaches are based on binary classifiers trained using supervised learning algorithms [33-35]. Under the supervised learning framework, a large set of N training samples, $\{(x_1, y_1),\ldots(x_N, y_N)\}$, is formed, where (x_i, y_i) is the i-th sample input-output pair, such that, x_i is the feature vector of the i-th sample denoting the input and y_i is its label (or class) denoting the desired output. Based on these labelled

training samples, a learning algorithm induces a function $g : X \rightarrow Y$, where X is the input space and Y is the output space. The ultimate objective is that the learned function, usually referred to as the model or classifier, should subsequently be able to correctly predict the labels of unseen instances. The most popular learning approaches are neural networks [36], adaptive boosting (AdaBoost) [37] and support vector machines (SVM) [38]. In order to apply a supervised learning paradigm to specifically perform the pedestrian detection task in IVS, the necessary steps are:

1) Manually collect a large number of pedestrian (object class = pedestrian) and non-pedestrian (object class = non-pedestrian) images.
2) Manually label every image to identify its class. For example, pedestrians can be assigned a label of 1 and non-pedestrians a label of -1.
3) Extract appropriate feature vectors for all images and form them into training samples (input-output pairs) by pairing with their corresponding class labels.
4) Select a learning algorithm to train a model or classifier based on the labelled training samples. In the context of pedestrian detection, the model/classifier is referred to as a *pedestrian detector*.
5) Apply the trained pedestrian detector to the target IVS environment to detect pedestrians.

The main challenge is the extent to which such supervised learning approaches can generalize from the labelled training data to data in unseen target IVS environments/scenes.

Compared to early ground-breaking works based on Haar-like features [39] or Histogram of Oriented Gradients (HOG) [40], contemporary state-of-the-art generic pedestrian detectors utilizing Integral Channel Features (ICF) [41] or Deformable Part-based Models (DPM) [42] can achieve two-fold increase in accuracy, at incredible speeds of over an order of magnitude faster [43-46]. With the most recent breakthroughs in deep neural networks such as Convolutional Neural Networks (CNN) [47], today's state-of-the-art generic pedestrian detectors have pushed the increase in accuracy to five-fold [48-51] ! Unfortunately, despite such tremendous advancements, the performance of generic pedestrian detectors remains particularly prone to the *dataset shift* complication [52, 53], whereby, the distribution of the test data in target visual surveillance scenes often differs significantly from that of the source training data due to various scene-specific factors such as scale, viewpoint, illumination, resolution, image quality and background complexity. As a consequence, the performance of state-of-the-art

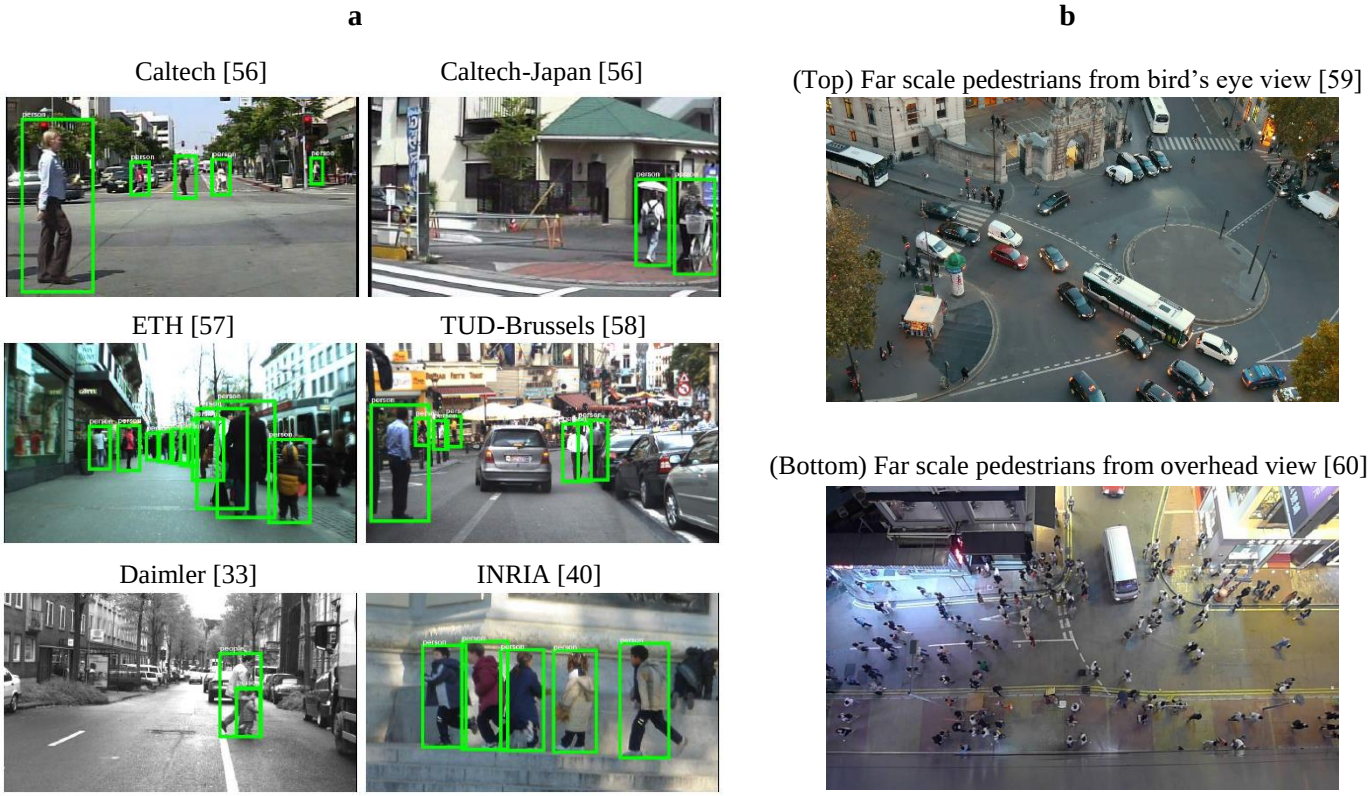


Figure 1.4: a) Annotations of different datasets done by [34] for evaluating state-of-the-art generic pedestrian detectors and b) Examples of difficult surveillance environments, gathered from the internet.

generic pedestrian detectors trained on publicly available datasets like INRIA [40] drops unacceptably when applied to unseen target surveillance environments [32-35, 54]. Even detectors based on deep neural networks are no exception to this performance drop [55].

A definitive indication of the performance limitations inflicted by dataset shift can be found by taking a closer look at the most comprehensive evaluation of state-of-the-art generic pedestrian detectors in literature [61], which is an online version of the original works by Dollar et al. [34], and is kept up-to-date with the 15 top-performing generic pedestrian detectors. All evaluated detectors are trained on INRIA, Caltech or both before evaluating on the datasets shown in Figure 1.4a. Of the six datasets, only Caltech provides a detailed breakdown of results with separate performance plots under various factors such as occlusion level, aspect ratio and scale – hence this is the only dataset ideal for in-depth analysis. It can be observed in [61] that though the overall detection rate of the best detectors is around 50%, they achieve near perfect detection rate (~100%) at near scale (at least 80 pixels), but this drops to 70% for medium scale (30-80 pixels) and plummets to 30% for far scale (under 30 pixels). This suggests that the

poor performance at medium/far scale heavily influences the overall detection rate. Therefore, how much would the overall detection rate drop if the detectors are to be applied to the surveillance scene shown in Figure 1.4b (Top), where all pedestrians are solely at far scale? Additionally, note how if the detectors are trained on INRIA/Caltech and then evaluated on any of the six datasets in Figure 1.4a, the performance is reasonable because there is minimal variation in viewpoint between the training set and evaluation set. As INRIA/Caltech contain no samples of pedestrians from overhead view, the trained detectors would perform poorly if applied to an overhead surveillance scene. Thus, how much would the overall detection rate drop if the detectors are to be applied to the surveillance scene shown in Figure 1.4b (Bottom), which is not only from an overhead view, but also at far scale that has already been proven problematic for the detectors?

Intuitively, training scene-specific detectors is the effective and necessary solution to the dataset shift problem. Instead of making futile attempts to train one generic detector that works well in all scenes, each unseen target scene and its associated complications can be handled independently, and by exploiting training samples and other complementary cues acquired directly from the target scene itself, a pedestrian detector specifically optimized for that scene can be generated. Various methods for training scene-specific pedestrian detectors have been developed, such as co-training [62-64], active learning[65-67], online incremental learning [68-70] and weakly supervised learning [71, 72], but domain adaptation [73-89] has emerged as the most popular and potent approach in recent years.

1.3.2 Research motivation

Scene-specific training approaches have made substantial progression [54] in tackling dataset shift, but they also have serious practical limitations. It can be observed that, for almost every existing method, training the scene-specific pedestrian detector entails the exploitation of either unlabelled or few manually labelled target samples. From an application standpoint, approaches [66, 71-73, 76, 80, 85, 86] that require any form of manual labelling are highly undesirable due to the need for repetitive human effort and suffer from poor scalability when deployed on today's large distributed camera networks [90, 91] - if a small network of just 50 cameras is considered, having to manually label as few as 100 samples per camera would demand the painstakingly laborious task of 5000 total annotations!

Figure 1.5: Samples from MIT (top), INRIA (middle) and QMUL Junction [92] (bottom). To gauge intra-dataset range of sample quality, view from left (better) to right (worse). Notice the dataset shift between INRIA-QMUL is considerably larger than that between INRIA-MIT

On the other hand, approaches that can exploit unlabelled samples are naturally less susceptible to scalability issues. To automate the labelling of target samples, these approaches invariably begin by applying a pre-trained generic detector on the target scene to acquire detection responses as potential target samples, which are then filtered by novel labelling algorithms to segregate the true positives (pedestrians) from the false positives. Promising results have been reported [70, 75, 78, 81-84]; however, it is crucial to perceive that the test surveillance datasets used to assess the performance of these approaches, such as MIT[93], CUHK[94] or PETS [95] do not adequately manifest the extent of complications in real-world surveillance environments. Under common extreme conditions such as very poor image quality [96, 97], resolution [98, 99] or illumination [100], the dataset shift between the source dataset and pedestrians in the target scene may be so large (see Figure 1.5) that the pre-trained generic detector, which is trained on the source dataset, may fail to acquire enough true positives when applied to the target scene. During scene-specific training, this shortage of true target positives would severely deteriorate the "specialization" of the pedestrian detector to the target scene. Thus, methods that depend on a pre-trained generic detector may often be rendered inapplicable in difficult environments.

Concretely, Figure 1.6 illustrates a typical visual surveillance network of multiple cameras monitoring different scenes, with a number of selected scenes zoomed in to highlight extreme scene-specific complications usually present in such large networks, such as poor resolution, low/excessive

illumination, abnormal viewpoint and far scale. To perform optimal pedestrian detection in these difficult scenes, it is necessary to train scene-specific pedestrian detectors using target samples acquired directly from these scenes. However, as previously elaborated, a) manually labelling target samples for each problematic scene is too laborious and time-consuming, hence unscalable, while b) applying pre-trained generic pedestrian detectors to acquire target samples is likely to fail in such extreme scenes.

We now converge all the aforementioned limitations to a pivotal question: Given unseen target surveillance environments of arbitrary difficulty as demonstrated in Figure 1.6, is it possible to train optimal scene-specific pedestrian detectors for each scene separately, subject to the constraints: A) No manual labelling of target samples is allowed, and B) No source dataset (real or virtual) or pre-trained generic detector can be utilized? In this thesis, we provide the surprising but favourable answer: yes.

Contrary to predominant research efforts that aim to alleviate dataset shift by adapting source dataset and/or source model to the target scene, we explore the possibility of eliminating dataset shift entirely by utilizing absolutely no source domain information. Essentially, we transcend the realm of domain adaptation, and propose to move towards an autonomous training framework that trains scene-specific pedestrian detectors for unseen target surveillance environments with zero manual labelling of target samples, but simultaneously, does not utilize any source dataset or pre-trained generic detector.

Our work stems from the following hypotheses: *Pedestrians have rudimentary but measurable attributes that make them similar to each other, and different from other objects. The triviality of these attributes allow them to be exploitable regardless of the complexity of the surveillance environments. If a mechanism can be devised to exploit multiple attributes and aggregate enough discriminative information, pedestrians can be reliably labelled in any surveillance environment. By integrating such an automated labelling mechanism into a practical and robust training framework that alternates between sample acquisition and training, scene specific information can be maximally exploited and the trained classifier can be iteratively improved to ultimately generate optimal scene-specific pedestrian detectors, all in an autonomous fashion.*

In order to validate the above hypotheses, the following research problems must be addressed:

- In the absence of both source training data and pre-trained generic detector, how to exploit pedestrian “attributes" to reliably label pedestrians/non-pedestrians in arbitrary scenes?

Figure 1.6: Scene-specific complications in different scenes of a typical visual surveillance network [101]

- How to design a training framework that can be applied to arbitrary scenes to generate optimal scene-specific pedestrian detectors, with zero manual labelling and zero source domain information? How would potential target samples be acquired? What are the necessary stages of such a framework? How would the automated labelling method be optimally exploited in such a framework?
- Given the limited experimental validation performed by existing works, how to carry out a comprehensive evaluation of the developed framework to ascertain its performance?
- There exists a significant gap between most theoretical works and their application to real-world problems. Therefore, how to ensure that developed framework does not only achieve convincing results on experimental datasets, but can be readily deployed to real-world surveillance environments?

1.3.3 Research aim and objectives

Based on a summarization of the research motivation detailed in the previous subsection, the aim of this research is to develop a training framework that can train scene-specific pedestrian detectors for unseen target surveillance environments without requiring any manual labelling of target samples, nor utilizing any source training dataset and/or pre-trained generic pedestrian detectors. Accordingly, the formulated objectives are:

- Develop mechanisms to automatically label target samples as pedestrians/non-pedestrians in surveillance environments, regardless of scene specifics or complexity.
- Design a sequence of training stages, with appropriate integration of the developed automated labelling mechanisms, to build an end-to-end training framework. When applied to an unseen surveillance environment, each stage of the framework should progressively improve the trained classifier to ultimately generate an optimal scene-specific pedestrian detector.
- Perform a thorough evaluation of the developed training framework by testing on a large number of experimental datasets, with different types of pedestrian detectors.
- Validate the applicability of the developed framework on real-world visual surveillance scenes.

In order to fulfil the third objective of performing a comprehensive evaluation of the developed framework, it is necessary to test the framework on different datasets with varying levels of difficulty. Hence, the 10 datasets selected for performance evaluation can be categorized into 3 difficulty levels as follows:

Hard

1) MIT Traffic (MIT) [93]: It is a far-field video of a traffic intersection.

2) *CUHK Square* (CUHK) [94]: It captures a pedestrian square at a university and is the most commonly used dataset for testing scene-specific pedestrian detectors.

3) *MONASH Frontgate* (MONASH): This dataset was entirely constructed by us. We captured the scene at the front entrance of MONASH University, Malaysia Campus.

Very Hard

4) *QMUL Roundabout* (QMUL-R) [102]: It captures a traffic roundabout.

5) QMUL Junction (QMUL-J) [92] : It captures a very busy traffic junction.

6) Karl-Wilhelm-Straße Intersection (KWSI) [103] : It captures a traffic intersection from a bird's eye view similar to MIT, but with a much larger camera tilt angle.

Medium

PETS 2009 [95] contains several clips of different scenarios enacted by a group of actors at a university campus. The selected clip is S1.L1, at timestamp 13-59, and the developed training framework is tested on four different views of this clip:

7) PETS 2009 View 1 (PETS-01): The scene is captured from a side-view.

8) PETS 2009 View 2 (PETS-02): The scene is captured from a frontal view.

9) PETS 2009 View 3 (PETS-03): It captures the scene from a similar view-point as PETS-01, but from slightly further.

10) PETS 2009 View 4 (PETS-04): It captures the scene from a rear view.

The difficulty level is determined by combinations of various scene-specific factors such as image quality, video resolution, pedestrian scale, scene illumination, viewpoint, occlusion levels and background complexity. Details pertaining to these factors and further elaboration of the datasets is provided at the beginning of the experimental results in Chapter 4.

The research aim and objectives indicate that the direction of this work is not the development of novel pedestrian detectors or innovative extensions to existing learning algorithms - an enormous amount of effort has already been dedicated to develop newer pedestrian detectors and domain adaptation techniques. However, as explained in subsections 1.3.1 and 1.3.2, a substantial gap remains between the achieved and desired performances, due to dataset shift. Therefore, the focus of this research is shifted towards devising strategies for exclusive exploitation of the target samples to train optimal scene-specific pedestrian detectors, with autonomy and practicality as primary design requirements.

1.4 Commercial output of this research

The training framework developed in this research has been implemented in the security industry to build a commercial anti-tailgate product. The product, and the application of the training framework in the product, are described in Chapter 5. The brochure and some images of the product are attached in Appendix C and Appendix D, respectively. More details on the product can be obtained from: http://www.elid.com/index.php/products/vision-based-system2/elideye-ev100

The commercial output of this research fulfils the final research objective listed in subsection 1.3.3.

1.5 Organization of this thesis

The remainder of the thesis is organized as follows:

Chapter 2: Literature Review. In this chapter, an overview of the performance of generic pedestrian detectors is presented. Then the existing scene-specific training paradigms are discussed– active learning, co-training, incremental learning, weakly supervised learning and domain adaptation. Representative works from each sub-category are discussed, before the placement of this research is shown, relative to existing works in literature.

Chapter 3: Virtually Autonomous Training (VAT). The definition of Class-Biased Attributes (CBA) is presented, which are derived from existing works on attributes [104]. The notion of how Direct

Attribute Evaluation (DAE) computes CBAs to differentiate pedestrians from non-pedestrians is then detailed. Pruners are the modules that implement DAE on a specific CBA – their specifics are explained. The oracle structure is subsequently presented, which is a hierarchical combination of Training Sample Filters (TSF) and Pruners; hence the labelling procedure is detailed on the pruner level (low), the TSF level (mid) and the oracle level (high). Full implementation details of multiple pruners are discussed, including their role in the labelling procedures and the design considerations that influence their contribution and precision in different surveillance environments.

Virtually Autonomous Training (VAT) is composed of three stages: Inception, Bootstrapping and Finalization. Inception generates the initial detector – procedures for automatic acquisition of potential target samples, application of the first oracle for labelling the target samples and training the initial detector with minimal errors are detailed. Bootstrapping reduces the false alarm rate and miss rate of the initial detector – optimal strategies for achieving these objectives are explored. Finalization aims to acquire target samples that may have been missed by the previous two stages – accordingly, procedures for deploying the retrained detector from the bootstrapping stage for obtaining the remaining potential samples, application of the second oracle for labelling these target samples and generating the final scene-specific pedestrian detector are presented. Various parameters that influence the performance of each stage, including the design of the applied oracles, are discussed in detail.

Chapter 4: Experimental Results. The 10 datasets used for validation are described in detail, 8 of which are annotated as a contribution of this research. Next, the three different detectors that are to be tested with VAT are presented. All implementation details and evaluation criteria are listed out. The oracle results are shown in full detail – overall performance of both oracles are evaluated in terms of their precision and recall, and the progression of samples as they pass through the oracles are demonstrated. For the individual TSFs, exhaustive numerical statistics are reported, graphical plots are presented for comparison, and pictorial results are shown to illustrate the kind of samples rejected by each TSF as well as those passed by the oracle. For every detection algorithm implemented with VAT and every dataset combination, DET curves were generated to assess the progression of the VAT stages and to compare the performance of VAT with generic detectors and manually trained detectors. A

thorough comparison of VAT was performed against the state-of-the-art scene-specific training approaches on the two most commonly used datasets. Important aspects such as influential factors in comparisons with state-of-the-art, VAT performance, oracle performance and influence of the selected detection algorithm are thoroughly discussed.

Chapter 5: Applications of VAT. An overview of the commercial product that has been developed as a full-blown output of the VAT framework is provided. Other industry problems that VAT can be applied to are also briefly described.

Chapter 6: Conclusions and Future Work. This chapter provides a brief summary of the developed VAT framework, the results achieved, and critiques the extent to which the objectives set out in the beginning of the thesis are met. The thesis concludes with various extensions, improvements and additional experiments of VAT suggested as future works.

2 Literature Review

Despite extensive studies for over 20 years, research and progress on pedestrian detection display no signs of slowing down and a large number of research papers are published each year. The vast literature on pedestrian detection can be divided into two broad categories:

- Generic pedestrian detectors
- Scene-specific pedestrian detectors

Accordingly, section 2.1 provides a short overview of the current performance of state-of-the-art pedestrian detectors. In section 2.2, the various scene-specific pedestrian detectors are reviewed, and representative state-of-the-art methods are compared to the approach developed in this research.

2.1 Overall performance of generic pedestrian detectors

The focus of this thesis is scene-specific pedestrian detection, therefore different generic pedestrian detectors are not reviewed in detail. Detailed reviews on state of the art prior to deep learning can be found in [34, 35]. For the current state-of-the-art pedestrian detectors, the reader is referred to [55, 105].

The most comprehensive pedestrian detection benchmarks are found at [34, 61]. The performance plots are presented under different conditions such as different levels of scale and occlusion. The overall results demonstrate performance inclusive of all conditions. Figure 2.1 compares the current state-of-the-art to the state-of-the-art prior to 2012, in terms of the overall performance (Top two plots in Figure 2.1) and when detecting far scale pedestrians (Bottom two plots in Figure 2.1). The most important takeaways are:

1) Previous state-of-the-art approaches have been knocked out completely by deep-learning approaches (the top two are standard baselines), primarily CNN.
2) Overall, compared to earlier approaches, the detection rates have improved considerably, but are still limited to 50% at 0.1 False Positives Per Image (FPPI).

3) On far scale, the performance compared to earlier approaches is once again better, but with detection rates of 25% at 0.1 FPPI (calculated by subtracting the miss rate from 100%), the performance on far scale pedestrians is still abysmal.

The latest reviews on deep-learning approaches have reinforced that small scale accounts for the largest number of false negatives by CNNs [105] and most CNNs have a difficult time dealing with small scale due to generation of "plain" features [55]. Furthermore, it has been emphasized [55] that CNNs are often sub-optimal for specific applications, and need to go through a fine-tuning phase before achieving the required performance.

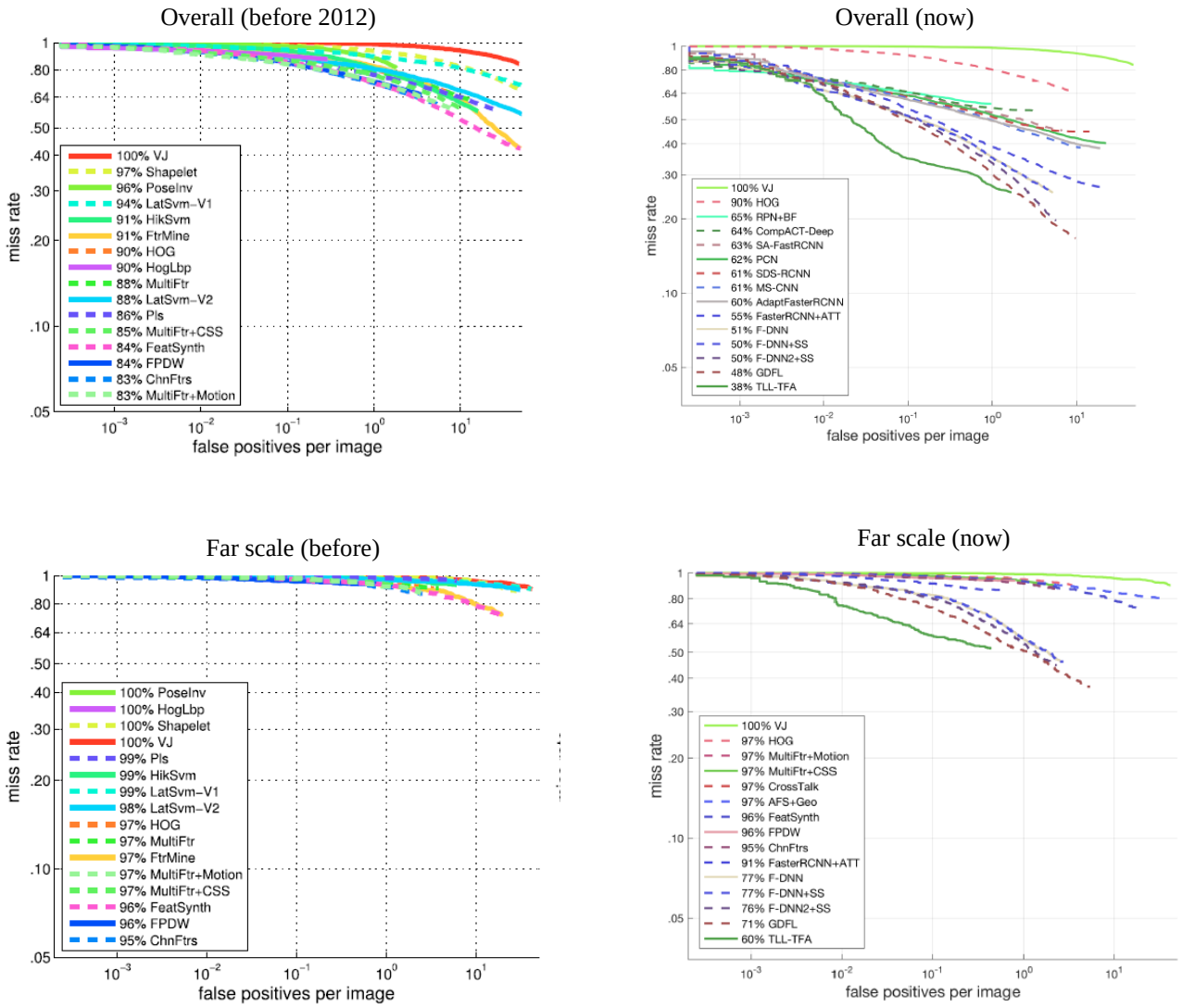


Figure 2.1: Comparison of current state-of-the-art against the state-of-the-art prior to 2012 [34, 61]

2.2 Scene-specific pedestrian detectors

2.2.1 Overview of various approaches

Co-training

Amongst the earliest works on scene-specific training is co-training. From a small training set of manually labelled data, Levin et al. [106] trained two detectors based on separate feature sets. Subsequently, using the co-training framework, these detectors trained each other using unlabelled data from the scene. Examples confidently classified by one detector were used to enlarge the training set of another. Javed et al.[64] employed co-training to boost ensemble classifiers in a similar manner based on PCA features and used it for learning vehicle and pedestrian detectors. Roth et al. [63] use a PCA-based reconstructive model and a discriminative boosted classifier to iteratively train each other. Sternig et al. [62] combined classifier grids [107] and co-training to form classifier co-grids, where individual grid classifiers operated with an overall compact classifier in co-training fashion. For co-training to be effective, the correlation between the detectors must be low. However this is difficult to achieve in practice and results in low detector performance.

Active learning

Active learning approaches interactively query a human user for "informative" target training examples and iteratively train the classifier until the desired performance level is achieved. Munaro and Cendese [65] queried a human agent to validate and select training examples from motion regions. Comparatively, Joshi and Porikli [66] only selected those examples that lie closer to the SVM hyperplane. However this can impose significant limitations on the autonomy of the training approach.

Online Incremental Learning

Incremental learning approaches perform online adaptation of a pre-trained detector based on a continuous stream of incoming detection responses acquired by applying the detector to the target scene. The main challenge is to correctly label these detection responses as true/false target positives. .

Rosenberg et al. [108] only used online samples that were confidently classified by the generic detector. In [109], Grabner and Bischof introduced a novel online boosting framework, where they trained a classifier in an incremental manner by performing on-line feature selection based on each new online training sample. This framework has been incorporated in multiple subsequent works on scene-specific detectors. Wu and Nevatia [68] made their labelling decisions by combining the responses of multiple part-based detectors and adapted their model by updating the base classifiers, cascade decision strategy and classifier complexity.

Drift is an inherent problem in incremental learning that causes degradation of the detector by updating it with sub-optimal positive online samples. Grabner et al. [110] alleviated this problem by formulating an update process in a semi-supervised fashion as a combined decision of a given prior and an on-line classifier. Babenko et al [111] proposed multiple instance learning (MIL), where instead of just collecting a single positive sample, a collection of instances around the positive sample were gathered into a bag, and a label was assigned to the bag rather than the single positive sample, with the assumption the bag had at least one correct positive sample. MIL successfully resolves the drift issue and various works [112, 113] have proposed extensions for further performance improvement. Sharma et al. [70] incrementally learnt a boosted classifier by optimizing a hybrid loss function comprising an offline loss function and a MIL loss function based on bags of online samples generated from successfully tracked detections.

Weakly supervised learning

Weakly supervised learning approaches considerably reduce human efforts by employing less labour-intensive labelling paradigms, such as annotating estimated centres of pedestrian instances rather than bounding boxes [71] or assigning labels to video frames to indicate the presence/absence of pedestrians [72].

Domain adaptation

The vast majority of scene-specific training approaches are based on domain adaptation. Strategies to achieve the "adaptation" are diverse due to the possible combinations of various system options such as supervision level, detector adaptation or detector retraining, iterative or non-iterative, target samples only or target plus source samples and normal or reweighed training samples. Since identifying a domain adaptation approach may be puzzling, we offer a generalized guideline: *Given a target scene, a domain adaptation approach requires and utilizes some form of information from both the target domain and the source domain to train a scene-specific pedestrian detector*. Note that incremental learning is technically domain adaptation too, but were presented separately to conform to the category those works were originally reported as. We discuss domain adaptation approaches that adapt the pre-trained detector, followed by those that retrain a scene-specific detector.

Cao et al. [73] adapted their pre-trained cascade classifier to the target scene by dynamically optimizing the threshold vector using cross entropy. In another approach [76], they generated a codebook which is a dictionary of visual key words extracted from HOG features via manifold learning. During detection, the codebook was dynamically updated by adding frequently occurring visual words as new key words and removing rarely used key words. Pang et al. [74] assessed the relationship between the source and target samples and shifted the features to the most discriminative locations and scales, and updated weak classifier coefficients using Covariate Boost. Xu et al. [81] used confident detection responses to predict the labels of uncertain detection responses by Gaussian Process Regression and adapted their pre-trained DPM-based SVM by perturbing the hyperplane. They extended this in [85] , where multiple target domains were arranged in a hierarchical structure and the pre-trained SVM was adapted to them jointly. Recently, proprietary classifiers have been generated for each target sample by dynamically adjusting the final layers of a pre-trained CNN [86] and using learnt regression networks that map samples to Exemplar SVMs [88].

For re-training approaches, a primary concern is to correctly label the target samples prior to training. Li-pin et al. [114] devised a converse approach to [85], and adapted multiple source domains to the target scene. Samples that were consistently predicted by multiple source detectors were assigned higher weight during retraining. Wang et al. [75] iteratively improved their pre-trained detector. In each

round, multiple context cues were exploited to reliably label the detection responses. The source samples were reweighed based on their similarity to target samples, whose detection confidences were incorporated to train a confidence-encoded SVM. In [82], they replaced the SVM with a deep model that automatically learnt discriminative features, the distribution of these features and scene-specific patterns. Zhang et al. [84] found a shared attribute space based on conditional distribution transfer sparse coding, where the target samples and the source samples have similar distributions. An attribute classifier was trained and subsequently used to label the target samples. Some approaches [78, 83] tracked the detection responses to verify labelling correctness and confident labels in a track were propagated across detection responses within that track. To address cases where a target-specific pedestrian detector is required prior to any on-site observation of pedestrians, training using virtual samples have also been studied [80, 87].

2.2.2 Comparisons with this research

Scene-specific training approaches can be classified as supervised (labelled target samples), unsupervised (unlabelled target samples) or semi-supervised (both). ***An autonomous system is highly automated and independent. Automation would necessitate minimum human assistance, while independence mandates minimal reliance on any external factors not from within the system.***

Table 2.1: Comparison of state-of-the-art scene specific training approaches for pedestrian detection

(Year) Author	Classifier compatibility	Pre-trained model used	Source samples used	Target samples labelling	No. of test datasets	Difficulty
(2011) Cao et al. [73]	AdaBoost	Yes	No	Supervised	2	**
(2011) Pang et al. [74]	AdaBoost	Yes	Yes	Supervised	2	***
(2012) Sharma et al. [70]	AdaBoost	Yes	No	Unsupervised	2	**
(2014) Wang et al. [75]	SVM	Yes	Yes	Unsupervised	2	***
(2014) Wang et al. [82]	Deep Model	Yes	Yes	Unsupervised	2	***
(2014) Xu et al. [81]	SVM	Yes	No	Unsupervised	6	***
(2014) Vasquez et al. [80]	SVM	No	Virtual	Supervised	3	***
(2014) Htike & Hogg [78]	SVM	Yes	No	Unsupervised	2	***
(2015) Zhang et al. [84]	SVM	Yes	Yes	Unsupervised	3	***
(2016) Xu et al. [85]	SVM	Yes	Yes	Supervised	5	***
(2017) Tang et al. [86]	Deep Model	Yes	No	N/A	2	***
(2017) Li et al. [89]	Deep Model	Yes	No	Supervised	3	***
(2017) Ye et al. [72]	SVM	No	No	Weakly Supervised	6	***
(2018) Hattori et al. [87]	Deep Model	No	Virtual	N/A	3	**
This research	**SVM & AdaBoost**	**No**	**No**	**Unsupervised**	**10**	*****

***Accordingly, the autonomy of a scene-specific approach can be evaluated by the levels of a) supervision and b) dependence on prior information*.** Table 2.1 presents a comparison of the current state-of-the-art scene-specific training approaches for pedestrian detection with respect to autonomy, classifier compatibility and experimental validation. The "maximum difficulty" is a subjective indication of the difficulty level of the most difficult dataset tested. The difficulty level is determined by combinations of various scene-specific factors such as image quality, video resolution, pedestrian scale, scene illumination, viewpoint, occlusion levels and background complexity. The reader is directed to Section 4.1 for detailed explanation of these factors.

Evidently, state-of-the-art approaches tend to adopt unsupervised labelling to improve autonomy but depend on the pre-trained model and/or source samples. In contrast, our approach has maximum autonomy because it performs unsupervised labelling, yet requires no prior information. More importantly, we advocate that surveillance environments can be so varied and complex that a scene-specific training approach must be tested on a large number of datasets to ascertain its performance. Except [81, 85], all approaches test at most 3 datasets and no approach was tested on extremely difficult datasets. Our extensive experimental validation exceeds the state-of-the-art in terms of number of classifiers tested, number of datasets tested and the upper bound of the dataset difficulty. Lastly, unsupervised labelling is bound to err, yet no approach provides any qualitative or quantitative error reports whatsoever. We provide detailed error reports of our unsupervised labelling in Chapter 4.

3 Virtually Autonomous Training (VAT)

This chapter commences with the conceptualization of the proposed training framework in Section 3.1. The conceptualized framework is then fully elaborated in Section 3.2 by presenting the complete algorithm and detailed pictorial overview of the end-to-end training framework. Finally, each individual stage of the end-to-end training framework is fully described in Sections 3.3-3.6.

3.1 Conceptualization

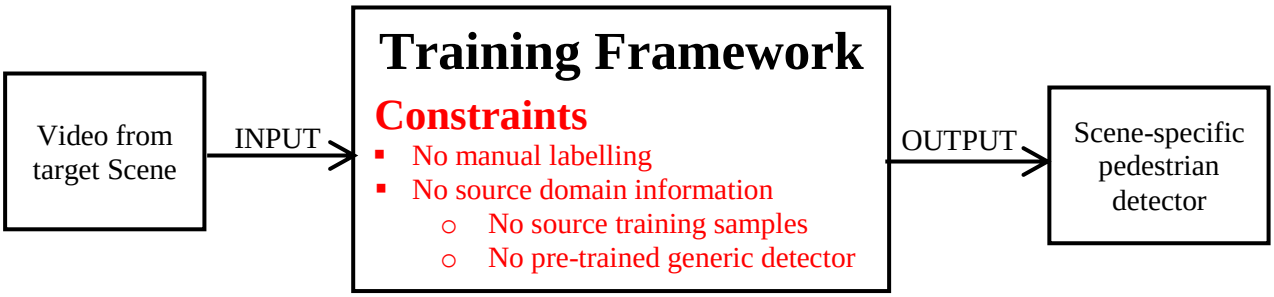


Figure 3.1: A block diagram of the research aim

The block diagram in Figure 3.1 presents a high-level overview of the research aim of this thesis introduced in Chapter 1. Given the highlighted constraints, a series of analytical questions arise (each **A**nalytical **Q**uestion is denoted by "**AQ**" subsequently):

- **AQ-1**: If no manual labelling is allowed and no pre-trained generic detector can be used, how to acquire potential target samples for training the scene-specific pedestrian detector?
- **AQ-2**: Once acquired, how to automatically and reliably label them as pedestrians and non-pedestrians?
- **AQ-3**: To train an optimal scene-specific pedestrian detector, various issues have to be addressed, such as:
    - **AQ-3A**: Maximizing exploitation of scene-specific information.
    - **AQ-3B**: Minimizing false positives (low false alarm rate).
    - **AQ-3C**: Minimizing missed positives (low miss rate).

Can all the above be accomplished just by acquiring and correctly labelling the target samples, or is there a need to design a more intricate sequence of training stages?

- **AQ-4**: What design strategies must be implemented in developing the framework to ensure that it is applicable, in the intended autonomous fashion, to different surveillance environments having considerable variations in and combinations of scene-specific complexities?

To successfully operate under the imposed constraints and simultaneously address the abovementioned questions, the developed training framework must be able to induce a pedestrian detector, and progressively improve it to generate the optimal scene-specific detector. This mandates the careful design of a concatenation of training stages. Accordingly, the proposed framework (Figure 3.2) is a sequence of three interlinked stages, where each stage serves a specific purpose as follows:

Stage 1: Inception. The objective of this stage is to generate an initial detector. Scene-specific training usually initiates with the acquisition of potential target samples. Under the constraints outlined in Figure 3.1, the robust option is to exploit motion to acquire moving regions as potential target samples [solves AQ-1]. To automatically label acquired target samples as pedestrians or non-pedestrians, automated *oracles* are engineered. The initial detector is trained using the initial training samples labelled by Oracle-1 [solves AQ-2]. It is critical to minimize labelling errors while training this initial detector to avoid error propagation as training progresses. By employing a high-precision oracle, a high percentage of correctly labelled initial target samples can be acquired, but at the same time, difficult and informative samples are likely to be rejected. Correspondingly, the initial detector is expected to have high recall, but high false alarm rate as well.

Stage 2: Bootstrapping. This stage reduces the false alarm rate of the initial detector by retraining it with hard negatives (false positives) [solves AQ-3A]. Bootstrapping with hard negatives usually has an undesirable side-effect of suppressing detection rate. To compensate for the decline in detection rate, a second retraining step must be performed with hard positives (false negatives) [solves AQ-3B].

Stage 3: Finalization. The necessity of this stage arises from the realization that the motion-based Inception stage has a high likelihood of lacking the localization ability inherent in pedestrian detectors – this shortcoming results in failure to acquire possibly impactful target samples during inception wherever accurate localization is required. To maximize the total acquisition of target samples [solves

AQ-3C], the target scene is revisited, but this time the retrained detector from Stage 2 is applied, with the objective that the acquired detection responses will now include the missed potential target samples in Stage 1. The detection responses (final samples) are labelled by Oracle-2 [solves AQ-2], and the final detector is subsequently trained. The final detector is ultimately output as the scene-specific pedestrian detector.

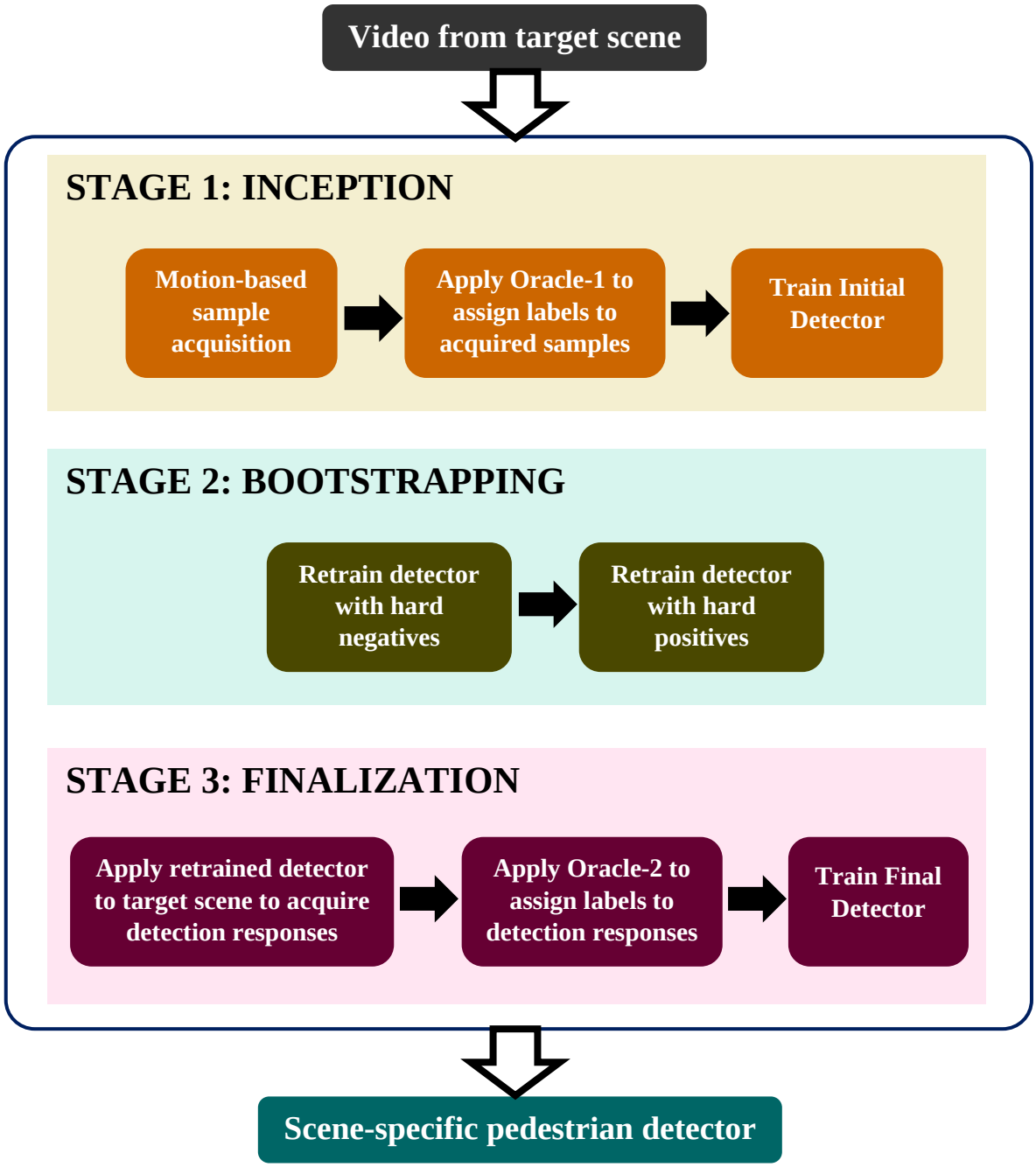


Figure 3.2: Conceptualization of the proposed VAT framework

The conceptualized training framework is depicted in Figure 3.2, and is termed as *Virtually Autonomous Training* (VAT). **Note that VAT is "virtually" autonomous due to the one-off human effort required to design the oracles**.

The capability of VAT to generate optimal scene-specific pedestrian detectors in a wide range of visual surveillance environments depends on the oracle design and the implementation of each VAT stage [solves AQ-4]. The rest of this chapter provides comprehensive implementation details of the conceptualized VAT framework. First, the conceptualized framework is elaborated in Section 3.2; specifically, the full algorithm of the end-to-end VAT framework is presented, along with a pictorial overview that is a detailed version of Figure 3.2, illustrating all the components and process flow within each stage. Next, Section 3.3 introduces oracles, their hierarchical composition and the underlying concepts that enable their utilization for labelling target samples reliably. Finally, Sections 3.4, 3.5 and 3.6 fully describe each of the three stages of VAT - the Inception stage, the Bootstrapping stage and the Finalization stage, respectively, including ideal execution strategies and design of relevant oracles and their integration, where applicable.

3.2 The end-to-end VAT framework

Figure 3.3 shows a detailed pictorial illustration of how all three training stages are concatenated to form the resultant end-to-end VAT framework. The illustration is meant to be utilized as a complete overview of the VAT framework, to observe the passage of the training data and the progression of detectors, and get comprehensive visualization of the exploited training samples at each step of every stage. The corresponding full algorithm of the VAT framework is detailed in Algorithm 1, with all the necessary notations listed in Table 3.1.

Every step in Algorithm 1 has a corresponding visualization in Figure 3.3; therefore, using Figure 3.3 in conjunction with Algorithm 1 can provide a clearer understanding of the VAT framework. Particularly, in cases where ambiguities may arise in Algorithm 1 when trying to perceive certain steps such as the augmentation of samples to the training dataset or application of different detectors to video sequences/set of samples, referring to the relevant visualization in Figure 3.3 can provide clarity.

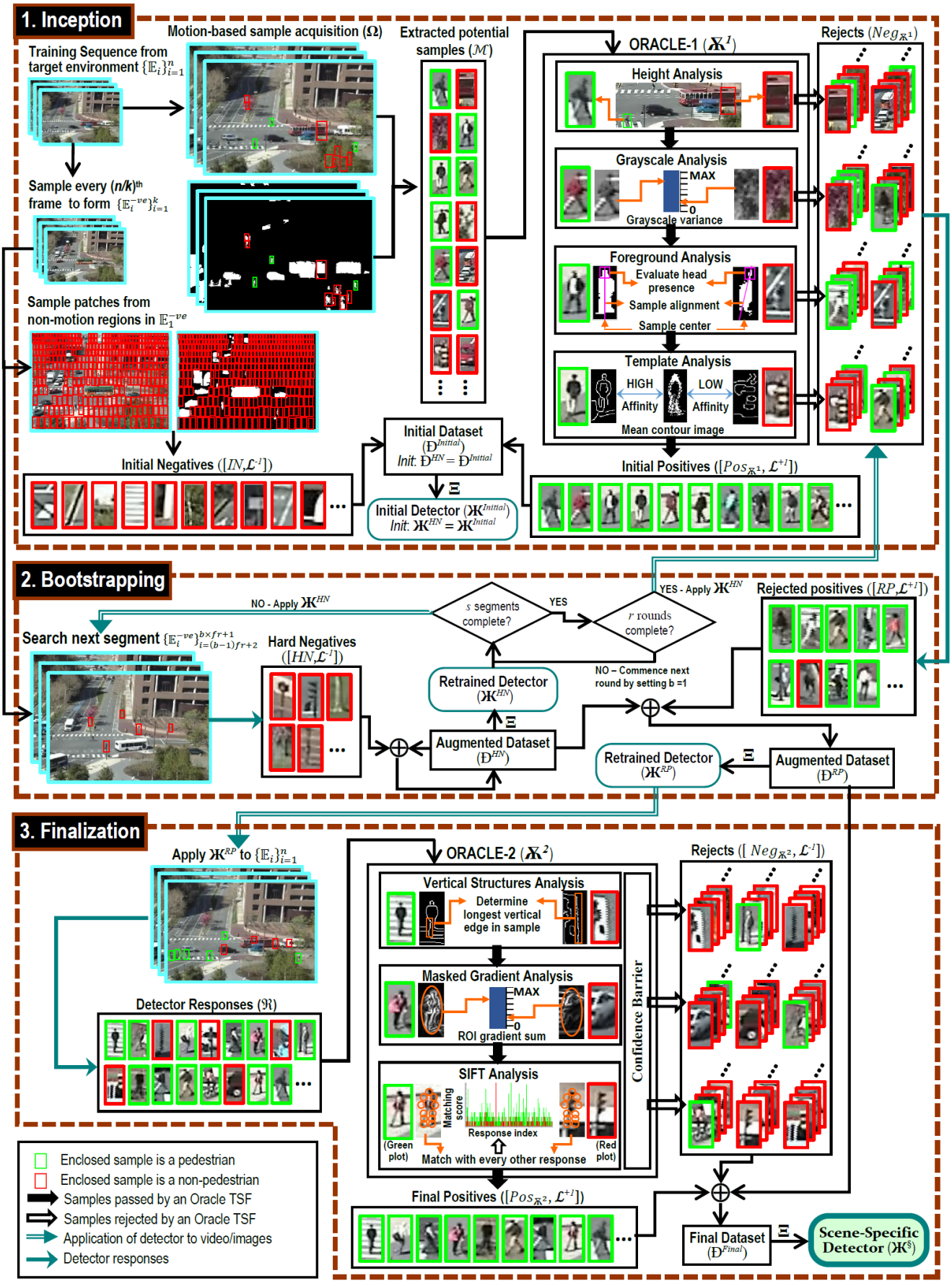


Figure 3.3: Detailed pictorial illustration of the end-to-end VAT framework implemented on MIT

Table 3.1: Notations used in Algorithm 1

$\boldsymbol{\Omega}$	Motion detection based sample extractor
$\{\mathbb{E}_i\}_{i=1}^{n}$	Training sequence from the target surveillance environment, ϵ
$\mathcal{M}$	Extracted potential pedestrian samples
$\boldsymbol{\aleph}^t$	Level t Oracle or t^{th} Oracle
$Pos_{\aleph^t}$	Samples labelled by $\boldsymbol{\aleph}^t$ as pedestrian instances
$Neg_{\aleph^t}$	Samples labelled by $\boldsymbol{\aleph}^t$ as non-pedestrian instances
Ð	Dataset
$\boldsymbol{\Xi}$	Supervised classifier learning – AdaBoost/SVM in this thesis
$\boldsymbol{Ж}$	Pedestrian detector
$\Re$	Detection responses acquired by applying a detector on $\{\mathbb{E}_i\}_{i=1}^{n}$
(r, s, fr)	(Bootstrapping rounds, bootstrapping segments per round, bootstrapping frames per segment)
$\{\mathbb{E}_i^{-ve}\}_{i=1}^{k}$	Sequence of k frames for collecting initial negatives and hard negatives, $k = s \times fr + 1$
IN	Initial negative samples
HN	Hard negatives or false positives
RP	Rejected positives of false negatives
§	Scene-specific
$\boldsymbol{\mathcal{L}}^z$	For training data [S, $\boldsymbol{\mathcal{L}}^z$], where S = $\{x_i\}$and $\boldsymbol{\mathcal{L}}^z = \{y_i\}$, $\forall x: \ y = z$

Algorithm 1: *The Virtually Autonomous Training* (**VAT**) *Framework*

Input :

Training video sequence of n frames from target surveillance environment - $\{\mathbb{E}_i\}_{i=1}^{n}$

Output :

Scene-specific detector - $\boldsymbol{Ж}^{\S}$

Stage 1 – Inception

1.1. Execute motion detection, perform post-processing and apply constraints to extract potential positive samples:

$$\mathcal{M} = \boldsymbol{\Omega}\,(\{\mathbb{E}_i\}_{i=1}^{n})$$

1.2. Feed potential samples to Oracle-1, to be labelled as pedestrians / non-pedestrians:

$$[Pos_{\aleph^1}, Neg_{\aleph^1}] = \boldsymbol{\aleph}^{\boldsymbol{1}}\,(\mathcal{M})$$

1.3. Form $\{\mathbb{E}_i^{-ve}\}_{i=1}^{k}$ for collecting initial negatives and bootstrapping by sampling $\{\mathbb{E}_i\}_{i=1}^{n}$ at n/k interval

1.4. Assemble the set *IN* by sampling all possible patches from all non-motion regions in $\mathbb{E}_1^{-ve}$

1.5. Label pedestrian instances obtained from Oracle-1 as positives, *IN* as initial negatives and learn $\boldsymbol{Ж}^{Initial}$:

$$Ð^{Initial} = \{[Pos_{\aleph^1}, \boldsymbol{\mathcal{L}}^{+1}], [IN, \boldsymbol{\mathcal{L}}^{-1}]\}$$
$$\boldsymbol{Ж}^{Initial} = \boldsymbol{\Xi}\,(Ð^{Initial})$$

Stage 2 – Bootstrapping

2.1. Retrain with hard negatives (false positives) in bootstrapping manner to learn $\boldsymbol{Ж}^{HN}$

Initialize: $\boldsymbol{Ж}^{HN} = \boldsymbol{Ж}^{Initial}$, and $Ð^{HN} = Ð^{Initial}$

for bootstrapping round = 1: r, **do**

for bootstrapping segment, b = 1 : s, **do**

Search exhaustively with $\boldsymbol{Ж}^{HN}$ for hard negatives : $HN = \boldsymbol{Ж}^{HN}(\{\mathbb{E}_i^{-ve}\}_{i=(b-1)fr+2}^{b\times fr+1})$

Augment dataset with acquired hard negatives and retrain to learn $\boldsymbol{Ж}^{HN}$:

$$Ð^{HN} = \{Ð^{HN}, [HN, \boldsymbol{\mathcal{L}}^{-1}]\}$$
$$\boldsymbol{Ж}^{HN} = \boldsymbol{\Xi}\,(Ð^{HN})$$

end for

end for

2.2. Retrain with rejected positives (false negatives) to learn $\boldsymbol{Ж}^{RP}$

Reacquire non-pedestrians from Oracle-1 if classified as positive by $\boldsymbol{Ж}^{HN}$: $RP = \boldsymbol{Ж}^{\boldsymbol{HN}}\,(Neg_{\aleph^1})$

Augment dataset with reacquired rejected positives and retrain to learn $\boldsymbol{Ж}^{RP}$:

$$Ð^{RP} = \{Ð^{HN}, [RP, \boldsymbol{\mathcal{L}}^{+1}]\}$$
$$\boldsymbol{Ж}^{RP} = \boldsymbol{\Xi}\,(Ð^{RP})$$

Stage 3 – Finalization

3.1. Apply $\boldsymbol{Ж}^{RP}$ on the training video sequence and obtain detection responses : $\Re = \boldsymbol{Ж}^{RP}\,(\{\mathbb{E}_i\}_{i=1}^{n})$

3.2. Feed detection responses to Oracle-2, to be labelled as pedestrians / non-pedestrians:

$$[Pos_{\aleph^2}, Neg_{\aleph^2}] = \boldsymbol{\aleph}^{\boldsymbol{2}}\,(\Re)$$

3.3. Augment pedestrian/non-pedestrian instances labelled Oracle-2 as final positives/negatives respectively and retrain to learn $\boldsymbol{Ж}^{Final}$:

$$Ð^{Final} = \{Ð^{RP}, [Pos_{\aleph^2}, \boldsymbol{\mathcal{L}}^{+1}], [Neg_{\aleph^2}, \boldsymbol{\mathcal{L}}^{-1}]\}$$
$$\boldsymbol{Ж}^{Final} = \boldsymbol{\Xi}\,(Ð^{Final})$$

3.4. Output $\boldsymbol{Ж}^{Final}$ as the trained scene-specific pedestrian detector : $\boldsymbol{Ж}^{\S} = \boldsymbol{Ж}^{Final}$

3.3 Oracles

Attributes are mid-level semantic features used in object description and classification [104, 115]. While low-level features such as HOG or Haar can only be processed by a computer, attributes such as “red”, “arm thickness” and “striped” are comprehensible to a human. We provide an expanded version of the definition of an attribute presented in [116] : *A property of an object is an attribute, if a human can visually determine its presence or measure it in a given object.* So an attribute like “red” is assigned a binary value based on its presence or absence, but an attribute like “arm thickness” would require determination of its presence followed by an evaluation of its thickness. Attributes have been employed in soft biometrics [117, 118] for person identification and attribute classifiers have been successfully applied to zero-shot classification [116, 119] and human pose estimation [120].

3.3.1 Direct Attribute Evaluation

Given an arbitrary object class of interest $\mathbb{C}$ (in this research , $\mathbb{C}$ = pedestrian) , if $\mathbb{X}$ represents the sample set of all objects from a particular target domain, let sample set $\mathbb{X}^+$ comprise of objects that belong to $\mathbb{C}$ (positive instances) and the sample set $\mathbb{X}^-$ comprise of objects that do not belong to $\mathbb{C}$ (negative instances). Depending on the target domain, $\mathbb{X}^-$ may contain objects from various classes. A class-biased attribute or CBA of $\mathbb{C}$ is a measurable appearance attribute that is highly prevalent amongst objects of $\mathbb{C}$ and discriminative enough to distinguish objects of $\mathbb{C}$ from objects of other classes. The critical difference between a CBA and a standard attribute is that a CBA should be highly exploitable regardless of the target domain, demanding it to be very generic in design. For example, a standard attribute like arm thickness can be evaluated in clear, close-range, frontal view surveillance scenes, but the limbs are often unclear in unconstrained surveillance environments due to poor resolution, far range or unusual camera angles, making this attribute unusable. Contrastingly, a CBA like sample alignment can always be evaluated, even under extreme conditions, as long as the person is visible.

Direct attribute evaluation or DAE is the computation of a real-valued score for a given sample by executing a relevant function that is formulated to measure a CBA. If DAE is applied to $\mathbb{X}$ to measure

a CBA of $\mathbb{C}$, the output values are expected to have two properties: *a)* instances of $\mathbb{X}^+$ would have similar values and *b)* there would be adequate differences between the values of instances of $\mathbb{X}^+$ and $\mathbb{X}^-$. It is suggestive that if these properties are utilized jointly, $\mathbb{X}^+$ can be reliably distinguished from $\mathbb{X}^-$. However, the first property may be rendered invalid if instances in $\mathbb{X}^+$ have large appearance variations and the second property may become infeasible if there are so many different classes in $\mathbb{X}^-$ that some instances are inevitably too similar to instances of $\mathbb{X}^+$. For $\mathbb{C}$ = pedestrian, both these complications are significantly subdued if the target domains are stationary surveillance environments. Firstly, because of fixed resolution and viewpoint, there would be limited appearance variations amongst instances of $\mathbb{X}^+$, producing similar values. Secondly, due to fixed scene, the total number of different object classes is limited compared to non-stationary scene, reducing the complexity of segregation.

In light of the above-mentioned aptness of DAE in stationary scenes, we make some intuitive conjecture. Since a DAE function is formulated based on a CBA of $\mathbb{C}$, the range of expected values from objects of $\mathbb{C}$ can be estimated a priori. When DAE is applied to $\mathbb{X}$ acquired from a stationary scene, and the distribution of the resultant values is observed, the values corresponding to instances of $\mathbb{X}^+$ will be clustered in the proximity of the expected range, whereas values corresponding to instances of $\mathbb{X}^-$ will lie further away and can be treated as outliers. Applying simple rejection criteria, these outliers can be removed. Obviously, a considerable number of instances from $\mathbb{X}^-$ will not be removed by using a solitary CBA, because they belong to classes that cannot be adequately differentiated from $\mathbb{C}$. To effectively eliminate a very high percentage, if not all, of the instances of $\mathbb{X}^-$, a combination of different CBAs must be used. The functionality becomes analogous to cascade classifiers [39]; only those instances not rejected as outliers are evaluated by subsequent DAE functions, and only an instance that is within the accepted range of the distribution for every CBA in the combination is ultimately labelled as an instance of $\mathbb{X}^+$ and hence, an object of class $\mathbb{C}$.

3.3.2 Pruners and Training Sample Filters (TSF)

A pruner is a module that executes DAE, and applies appropriate rejection criteria to segregate objects of interest. In order to construct a DAE function based on a CBA, a relevant first-order feature must be

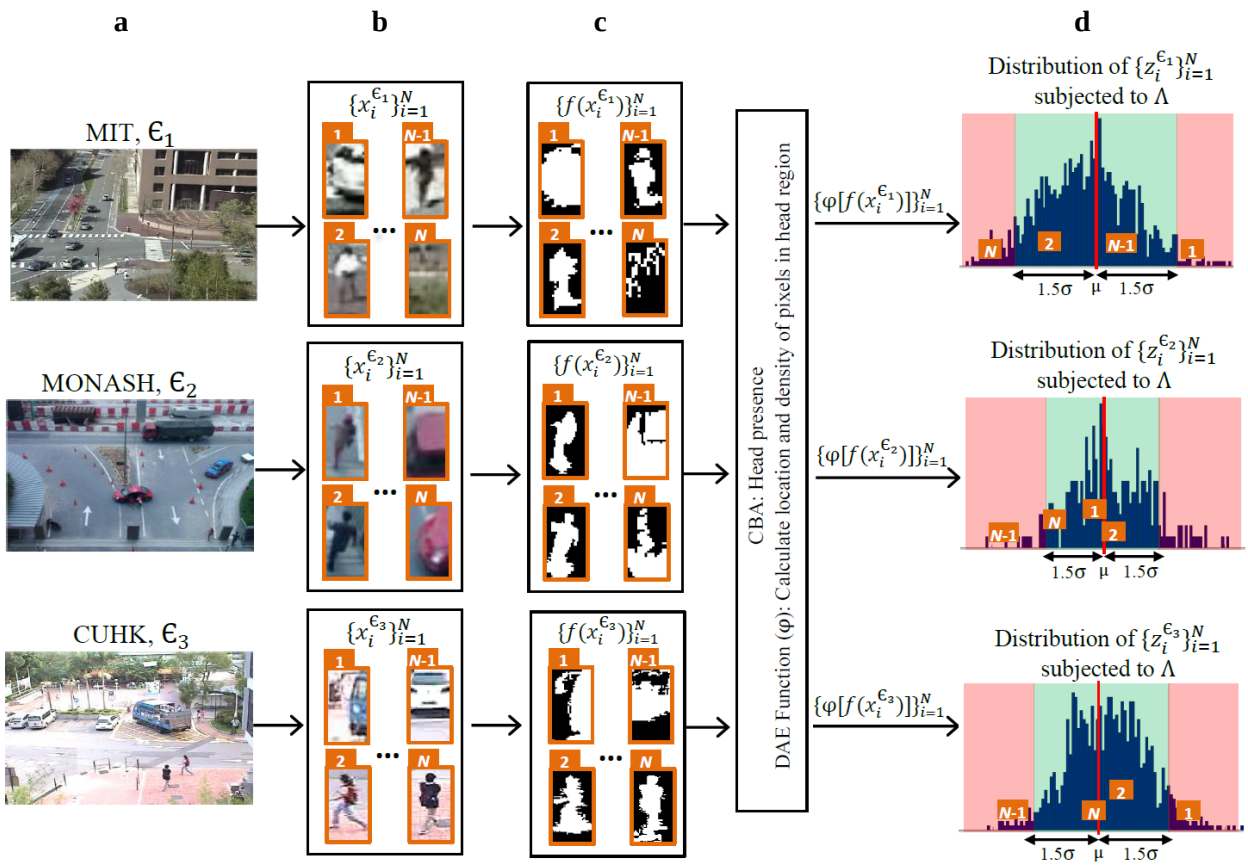


Figure 3.4: Functionality of a pruner for CBA "Head presence" of object class "Pedestrian". (a) Different surveillance environments/scenes. (b) N samples extracted from each surveillance environment, labelled by their index. (c) Binary foreground obtained as the feature to be passed as argument to the DAE function φ that evaluates the CBA. (d) Distribution of output values subjected to simple rejection criteria formulated from the μ and σ of the output values. The red regions indicate the rejection range and the green region indicates the expected range. Passed and rejected samples can be identified by the location of their index labels in the distribution

selected to be utilized as an argument, for example, grayscale, foreground or gradient. Given a set of N samples from environment/scene ϵ, denoted by $\{x_i^{\epsilon}\}_{i=1}^N$, we select a feature f. The corresponding feature representations are denoted by $\{f(x_i^{\epsilon})\}_{i=1}^N$. A discriminative DAE function φ can then be formulated based on feature f to compute the CBA for all samples:

$$\{z_i^{\epsilon}\}_{i=1}^N = \{\varphi[\, f(x_i^{\epsilon})\,]\}_{i=1}^N \tag{3.1}$$

and the resultant distribution of output values $\{z_i^{\epsilon}\}_{i=1}^N$ is subjected to an appropriate rejection function Λ to segregate the samples as follows:

$$\left[\{\text{pass}(j)\}_{j=1}^{N^{pass}}, \{\text{reject}(k)\}_{k=1}^{N^{reject}}\right] = \Lambda(\{z_i^{\epsilon}\}_{i=1}^{N}) \mid N^{pass} + N^{reject} = N$$
$$\text{s.t. } x_i = \begin{cases} \text{passed , if } i \in \{\text{pass}(j)\} \\ \text{rejected , if } i \in \{\text{reject}(k)\} \end{cases} \tag{3.2}$$

An elaborate illustration of the implementation of a pruner (for object class "Pedestrian") in three different environments is presented in Figure 3.4. In ϵ_1, the pedestrians appear very small due to far-field video, in ϵ_2, they have unusual orientation due to large camera tilt angle and ϵ_3 has low-contrast and smeared pedestrians (zoom in to view). Despite the considerable differences between the environments, the pruner can fairly differentiate pedestrians by focusing on a CBA of pedestrians. The accuracy of the segregation can be checked through the location of the index labels on the distribution. The pruner operates correctly for ϵ_1 and ϵ_3, but errs for ϵ_2. In ϵ_2, the target sample with label '*N*' is an image of car part, i.e. non-pedestrian instance, but upon executing DAE, its output value narrowly falls within the expected range because the head region is similar to that of a pedestrian instance. Consequently, it is incorrectly passed as a pedestrian instance. As explained in subsection 3.3.1, such errors should be anticipated. When the set of target samples are subjected to a combination of pruners, each evaluating a different CBA, non-pedestrian instances which may be incorrectly passed by some pruners are likely to be correctly rejected by others.

Two different pruners can be implemented based on the same feature but evaluating different CBA or they may evaluate the same CBA based on different features. For improved coherence and structure, a training sample filter (TSF) comprises of one pruner or a collection of pruners grouped together either because they utilize the same feature or evaluate the same CBA. Pruners in a TSF are arranged in series if they use the same feature but evaluate different CBAs, and in parallel if they use different features but evaluate the same CBA.

3.3.3 Oracle configuration

We now describe the hierarchical composition and functionality of an oracle. An oracle is constructed by sequentially combining multiple training sample filters (TSF), each of which may consist of one or more pruners, arranged in series or parallel. Figure 3.5 depicts the possible configuration of components

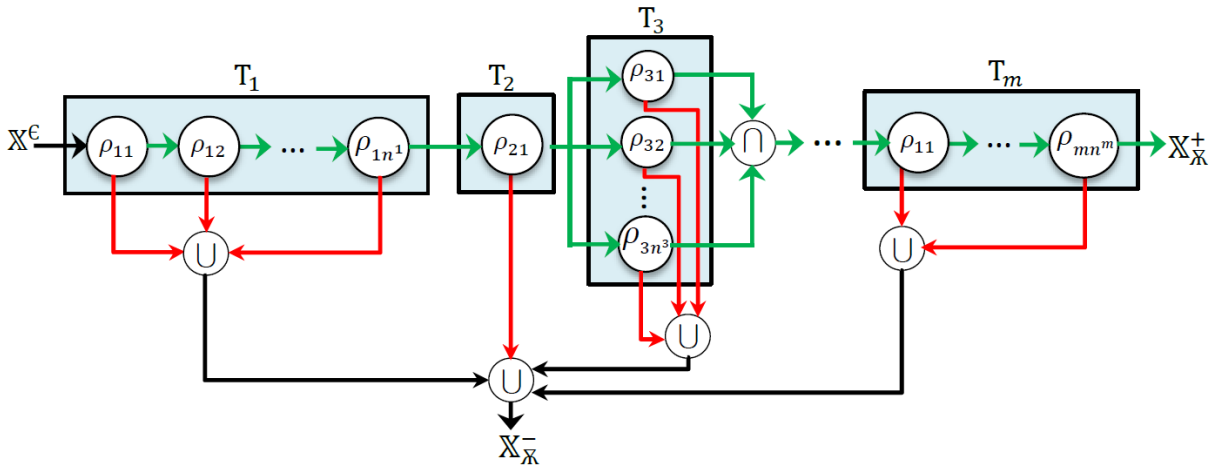


Figure 3.5: Oracle configuration. Green arrows indicate flow of passed samples and red arrows indicate flow of rejected samples

within an oracle and the passage of samples. For the rest of this subsection, it is advised to refer to Figure 3.5 for better comprehension and visualization of the discussed concepts.

Denote an oracle by $\mathbb{A}$, a TSF by T and a pruner by ρ. Given an object class of interest $\mathbb{C}$, let $\mathbb{X}^{\epsilon}$ represent the set of samples extracted from an environment ϵ, such that $\mathbb{X}^{\epsilon} = \mathbb{X}^{+} \cup \mathbb{X}^{-}$, where $\mathbb{X}^{+}$comprises samples that belong to $\mathbb{C}$ and $\mathbb{X}^{-}$comprises samples that do not belong to $\mathbb{C}$. The goal of an oracle, $\mathbb{A}$, is to segregate $\mathbb{X}^{\epsilon}$ into $\mathbb{X}_{\mathbb{A}}^{+}$ and $\mathbb{X}_{\mathbb{A}}^{-}$, aiming to achieve $\mathbb{X}_{\mathbb{A}}^{+} \subseteq \mathbb{X}^{+}$and $\mathbb{X}_{\mathbb{A}}^{-} \subseteq \mathbb{X}^{-}$ To that end, the segregation is performed hierarchically via:

$$\begin{aligned} [\mathbb{X}_{\mathbb{A}}^{+}, \mathbb{X}_{\mathbb{A}}^{-}] &= \mathbb{A}(\mathbb{X}^{\epsilon}) \\ [\mathbb{X}_{\mathrm{T}}^{+}, \mathbb{X}_{\mathrm{T}}^{-}] &= \mathrm{T}(\mathbb{X}_{\mathrm{T}}^{\epsilon}), \text{s.t. } \mathbb{X}_{\mathrm{T}}^{\epsilon} \subseteq \mathbb{X}^{\epsilon}, \mathbb{X}_{\mathrm{T}}^{+} \subseteq \mathbb{X}_{\mathbb{A}}^{+} \text{ and } \mathbb{X}_{\mathrm{T}}^{-} \subseteq \mathbb{X}_{\mathbb{A}}^{-} \\ [\mathbb{X}_{\rho}^{+}, \mathbb{X}_{\rho}^{-}] &= \rho(\mathbb{X}_{\rho}^{\epsilon}), \text{s.t. } \mathbb{X}_{\rho}^{\epsilon} \subseteq \mathbb{X}_{\mathrm{T}}^{\epsilon}, \mathbb{X}_{\rho}^{+} \subseteq \mathbb{X}_{\mathrm{T}}^{+} \text{ and } \mathbb{X}_{\rho}^{-} \subseteq \mathbb{X}_{\mathrm{T}}^{-} \end{aligned} \tag{3.3}$$

where, $\mathbb{X}^{\epsilon}$, $\mathbb{X}_{\mathrm{T}}^{\epsilon}$ and $\mathbb{X}_{\rho}^{\epsilon}$ are input samples at the oracle, TSF and pruner levels, respectively and $\mathbb{X}_{\mathbb{A}}^{+}/\mathbb{X}_{\mathbb{A}}^{-}$, $\mathbb{X}_{\mathrm{T}}^{+}/\mathbb{X}_{\mathrm{T}}^{-}$ and $\mathbb{X}_{\rho}^{+}/\mathbb{X}_{\rho}^{-}$ represent the samples labelled at the oracle, TSF and pruner levels, respectively, as belonging/not belonging to $\mathbb{C}$. Pruners are the building blocks of the oracle as CBAs are evaluated and samples are segregated at this level. As such, the pruner level equation in (3.3) is a compact, combined representation of equations (3.1) and (3.2) described in subsection 3.2.2, where $\mathbb{X}_{\rho}^{\epsilon} = \{x_i^{\epsilon}\}_{i=1}^{N}$, $\mathbb{X}_{\rho}^{+} = \{x_{\mathrm{pass}(j)}^{\epsilon}\}_{j=1}^{N^{pass}}$ and $\mathbb{X}_{\rho}^{-} = \{x_{\mathrm{reject}(k)}^{\epsilon}\}_{k=1}^{N^{reject}}$.

Denote the set of TSFs in an oracle by $\{\mathrm{T}_j\}_{j=1}^{m}$, where m is the number of TSFs in the oracle and denote the set of pruners in the j^{th} TSF by $\{\rho_{jk}\}_{k=1}^{n^j}$, where n^j is the number of pruners in the j^{th} TSF. The inputs and outputs of the j^{th} TSF is given by:

$$[\mathbb{X}_{\mathrm{T}_j}^{+}, \mathbb{X}_{\mathrm{T}_j}^{-}] = \mathrm{T}_j\left(\mathbb{X}_{\mathrm{T}_{j-1}}^{+}\right) \tag{3.4}$$

Equation (3.4) indicates that samples passed by a TSF are forwarded to the next TSF as input samples. Note that for j=1, $\mathbb{X}_{\mathrm{T}_{j-1}}^{+} = \mathbb{X}^{\epsilon}$. Within the j^{th} TSF, the inputs and outputs of the k^{th} pruner is given by:

$$[\mathbb{X}_{\rho_{jk}}^{+}, \mathbb{X}_{\rho_{jk}}^{-}] = \begin{cases} \rho_{jk}\left(\mathbb{X}_{\mathrm{T}_{j-1}}^{+}\right), \text{ if } n^j = 1 \text{ or } n^j > 1 \text{ in parallel} \\ \rho_{jk}\left(\mathbb{X}_{\rho_{j(k-1)}}^{+}\right), \text{if } n^j > 1 \text{ in series. For k} = 1, \mathbb{X}_{\rho_{j(k-1)}}^{+} = \mathbb{X}_{\mathrm{T}_{j-1}}^{+} \end{cases} \tag{3.5}$$

Referring to (3.5), if pruners using different features but same CBA are arranged in series, it would be redundant to segregate passed samples from the previous pruner by evaluating the same CBA as the previous pruner. Hence, such pruners are executed concurrently on the passed samples from the previous TSF in a parallel configuration. The outputs of the j^{th} TSF is a combination of the outputs of its constituent pruners as follows:

$$\begin{aligned} &\mathbb{X}_{\mathrm{T}_j}^{+} = \mathbb{X}_{\rho_{j1}}^{+}, \ \mathbb{X}_{\mathrm{T}_j}^{-} = \mathbb{X}_{\rho_{j1}}^{-}, \text{if } n^j = 1 \\ &\mathbb{X}_{\mathrm{T}_j}^{+} = \mathbb{X}_{\rho_{jn^j}}^{+}, \mathbb{X}_{\mathrm{T}_j}^{-} = \mathbb{X}_{\rho_{j1}}^{-} \cup \mathbb{X}_{\rho_{j2}}^{-} \ldots \cup \mathbb{X}_{\rho_{jn^j}}^{-}, \text{if } n^j > 1 \text{ in series} \\ &\mathbb{X}_{\mathrm{T}_j}^{+} = \mathbb{X}_{\rho_{j1}}^{+} \cap \mathbb{X}_{\rho_{j2}}^{+} \ldots \cap \mathbb{X}_{\rho_{jn^j}}^{+}, \mathbb{X}_{\mathrm{T}_j}^{-} = \mathbb{X}_{\rho_{j1}}^{-} \cup \mathbb{X}_{\rho_{j2}}^{-} \ldots \cup \mathbb{X}_{\rho_{jn^j}}^{-}, \text{if } n^j > 1 \text{ in parallel} \end{aligned} \tag{3.6}$$

The outputs of the oracle $\mathbb{X}$ can therefore be expressed in terms of the TSF outputs as:

$$\begin{aligned} \mathbb{X}_{\mathbb{X}}^{+} &= \mathbb{X}_{\mathrm{T}_m}^{+} \\ \mathbb{X}_{\mathbb{X}}^{-} &= \mathbb{X}_{\mathrm{T}_1}^{-} \cup \mathbb{X}_{\mathrm{T}_2}^{-} \cup \mathbb{X}_{\mathrm{T}_m}^{-} \end{aligned} \tag{3.7}$$

Referring to (3.4), the solution $\mathbb{X}_{\mathrm{T}_m}^{+}$is obtained from $[\mathbb{X}_{\mathrm{T}_m}^{+}, \mathbb{X}_{\mathrm{T}_m}^{-}] = \mathrm{T}_m\left(\mathbb{X}_{\mathrm{T}_{m-1}}^{+}\right)$. This is a recursive procedure that requires the passed samples from the previous filter, and commences operation with initial input samples from the environment, denoted by $\mathbb{X}_{\mathrm{T}_0}^{+} = \mathbb{X}^{\epsilon}$.

3.3.4 Design guidelines

There are fundamental differences between oracles and existing approaches [104, 115, 116, 120] that use attribute classifiers for labelling. Attribute classifiers usually output binary scores indicating the presence or absence of an attribute and the combination of these outputs are then processed to determine the object label. As a simple example, given an object and six trained attribute classifiers – black, white, brown, stripes, water and eats fish, if the outputs of the classifiers are

a) black: 0, white: 1, brown: 0, stripes: 0, water: 1 and eats fish: 1, then object = polar bear

b) black: 1, white: 1, brown: 0, stripes: 1, water: 0 and eats fish: 0, then object = zebra

Attribute classifiers are similar to object classifiers in the sense that they also need to be trained on large datasets, such as the *Animals with Attributes* dataset [116] for labelling animals; except that their end goal is to determine the presence of attributes rather than objects. The need for training exposes them to the same dataset shift complications faced by generic object classifiers like pedestrian detectors.

The principal advantage of DAE is that by directly evaluating real-valued scores for CBAs, they eliminate the need for any training, consequently making the dataset shift problem non-existent. However, unlike attribute classifiers that can be used to output the corresponding object label when given a single object, pruners that execute DAE are not designed to operate on lone objects, but rather on a set of objects. As explained in subsections 3.2.1 and 3.2.2, a pruner applies DAE on a set of objects, and exploits the resultant distribution to collectively reject those instances that do not belong to the object class of interest. Thus, given a set of objects, an oracle constructed as a hierarchical composition of pruners performs multiple stages of rejections to ultimately segregate the set of objects into two groups: 1) instances that belong to the object class of interest, and 2) all the rejects, which are the instances that do not belong to the object class of interest.

The objective of designing oracles is to enable reliable labelling of object classes of interest in different environments of varying levels/combinations of complexities. However, the avoidance of training means that oracles have to be engineered for each object class of interest based on heuristics. It can be challenging to design oracles with widespread applicability as factors such as choice of CBAs or

formulated DAE functions, can affect the overall performance of the oracle. The following are important guidelines for designing optimal oracles:

Collect enough datasets to represent object class variations. For a given object class of interest, collect multiple application scenarios or environments where the oracle may have to be applied. This facilitates taking into account as many scene-specific complications as possible that the oracle may need to tackle.

Analyse the datasets collectively. Observe the instances that do not belong to the object class of interest and have to be rejected, across all datasets. Find CBAs that can be exploited to reject particular types; however, ensure the rejection ability of the selected CBA spans as many datasets as possible.

String pruners. Combine pruners accordingly if their extracted features or their evaluated CBAs are similar. This provides better modularity and coherence as the oracle increases in size.

Over-emphasize generalization. This is the single most important (and difficult) rule that governs optimal oracle design. Generalization should be the top priority when selecting a CBA and formulating the corresponding DAE function. If specific attention is not paid to this rule, the engineered pruners will only have limited applicability – this will make it necessary to design a large number of pruners in order to cover all the different types of objects that have to be rejected across all datasets. This not only adds redundancy to the resultant oracle but will also cause incorrect rejections due to poorly generalized pruners.

Design based on input. Oracles are not object detectors – they are not meant to localize the object class of interest in images. Rather, when a set of objects are made available to them, oracles assign labels to these objects. For the same application domain, the input set of objects can be substantially different based on the acquisition mechanism. For example, in IVS environments, the types of objects in motion regions acquired by background subtraction will be different from those in detection responses acquired by applying a pedestrian detector. Attempting to devise a single oracle to deal with all kinds of input mechanisms is doable but is likely to increase oracle complexity and redundancy. Instead, designing an oracle with widespread applicability, but at the same time tuned to a specific input mechanism, can be a much more efficient approach. This, and the previous rule, reinforces that when designing an oracle, the focus should be to find the minimum number of CBAs that are able to reject all types of objects across all datasets.

Determine correct sequence of TSFs. This requires a good understanding of any existing relationships between the different types of objects that needs to be rejected. For example, it is possible that the presence of objects of type A, can reduce the efficiency of rejection of objects of type B – in that case it becomes mandatory to apply the TSF for rejecting objects of type A first. Determining the correct sequence usually requires a combination of intuition and experimentation.

Increase discriminative ability to maintain rejection capability. TSFs are required to be generic in order to be highly exploitable. However, as the objects that need to be rejected are gradually removed by the oracle, it becomes more difficult to segregate the remaining instances that do belong to the object class of interest. Therefore, each subsequent TSF in the sequence must perform DAE of increasingly discriminative CBAs, without compromising generality too much.

3.4 VAT Stage 1: Inception

This section provides full implementation details for the first stage of VAT: Inception. The reader is advised to frequently refer back to Figure 3.3 and Algorithm 1 as they progress through this section to enable a clearer understanding of the connections between various components/steps as well as their relevance to the overall framework.

3.4.1 Motion-based sample acquisition

The first step in training a scene-specific pedestrian detector is the automatic acquisition of potential pedestrian samples from the target environment. Unlike many existing scene-specific approaches that apply a pre-trained detector and acquire detection responses as potential pedestrian samples, VAT begins operation by relying solely on motion information and acquires moving objects as potential pedestrian samples. Motion-based sample acquisition is rational because pedestrians are likely to be in motion in most surveillance environments. Even stationary pedestrians would eventually need to move to get from one point to another, making them attainable as potential samples. Sample acquisition using motion detection has two important advantages over using pre-trained detectors: 1) Sample acquisition relying on pre-trained generic detectors may be substantially hindered in difficult environments (see Figure

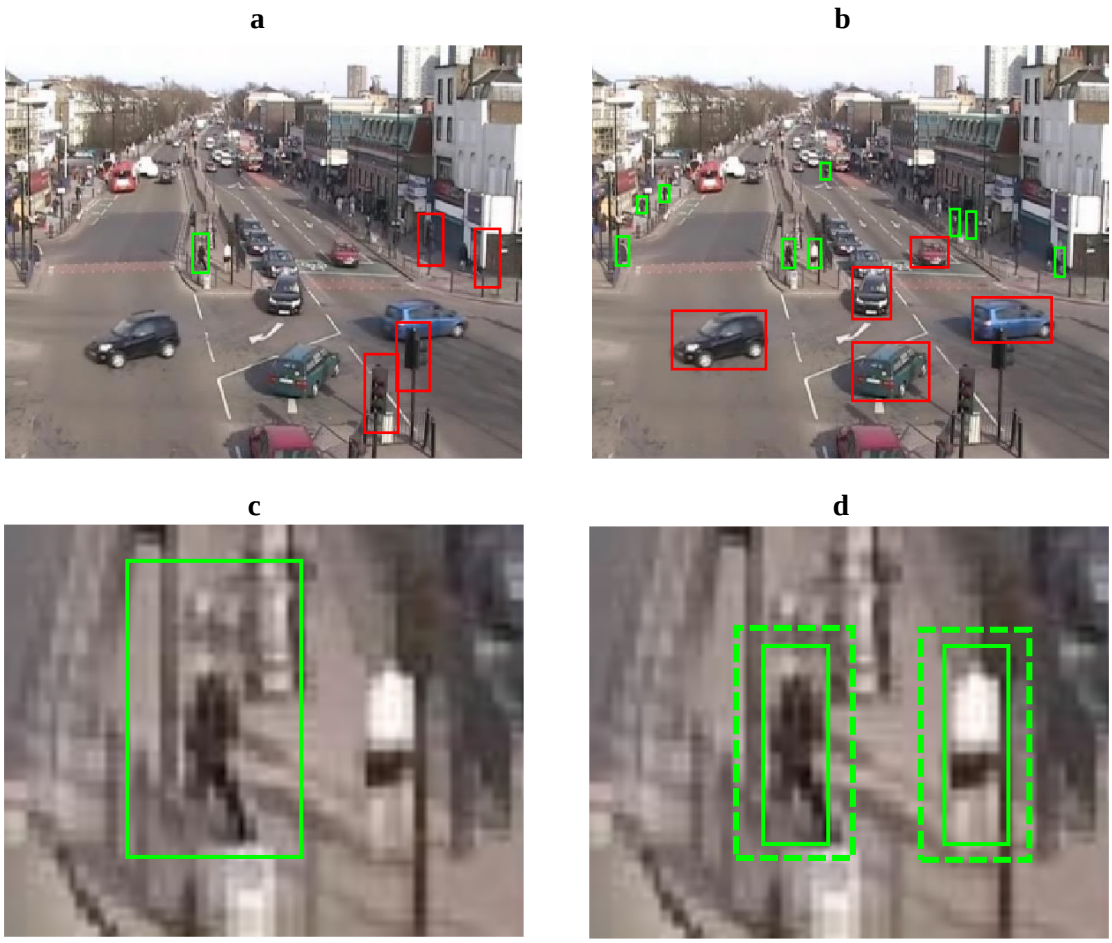


Figure 3.6: Green and red bounding boxes indicate pedestrian and non-pedestrian instances, respectively. a) Sample acquisition using pre-trained generic pedestrian detection. b) Sample acquisition using motion detection. c) Zoomed in version of a), demonstrating alignment of pedestrian in the acquired sample. d) Zoomed in version of b), solid boxes indicate the original bounding boxes proposed by motion detection and dashed boxes indicate the region to be acquired after 15% expansion.

3.6a) due to dataset shift, but motion is a highly exploitable cue that is unaffected by dataset shift, making sample acquisition feasible even under extreme conditions (see Figure 3.6b), and 2) As detection responses are non-maximal suppressions of multiple overlapping output bounding boxes, they often have sub-optimal sample alignment (see Figure 3.6c), whereas, pedestrian instances obtained using tight bounding boxes around motion regions usually have near-perfect central alignment, both vertically and horizontally (see solid boxes in Figure 3.6d).

The acquisition procedure is as follows. Given a video sequence $\{\mathbb{E}_i\}_{i=1}^{n}$, from a target surveillance environment ϵ, background subtraction is applied and the foreground pixels are subjected to morphological operations to remove noise and fill holes. Using connected component analysis on the

resultant foreground regions, tight bounding boxes are proposed. A margin of few pixels around a sample that is about 15% of the dimensions of the bounding box provides valuable context that improves detection rates [40]; similar expansion ratios are applied to the acquired samples (see dashed boxes Figure 3.6d). Lastly, using the facts that the aspect ratio, *ar*, of pedestrians is much smaller than that of vehicles and at the same time not too small; the foreground regions whose expanded bounding boxes satisfy the conservative constraint $0.2 < ar < 0.6$, are extracted as the potential pedestrian samples ($\mathcal{M}$). The effectiveness of exploiting the aspect ratio can be inferred by observing Figure 3.6b - most of the samples with red bounding boxes will not be acquired by applying the aspect ratio constraint.

3.4.2 Oracle-1

$\mathcal{M}$ is highly likely to contain several false positives from non-pedestrian moving objects. To segregate $\mathcal{M}$ into pedestrian and non-pedestrian instances, Oracle-1 ($\mathbb{X}^1$) was designed. In order to engineer the pruners and TSFs, and to configure their optimal arrangement in Oracle-1, $\mathcal{M}$ from different datasets (see Section 4.1) were jointly examined. Through experimentation based on the guidelines from subsection 3.2.4, it transpired that Oracle-1 has the highest precision when it is configured as a sequence of 4 TSFs. The full implementation details of these TSFs are presented in the next four subsections. Note that the value of 1.5σ used in the rejection criteria of the pruners has been determined empirically.

3.4.2.1 TSF 1 – Height analysis (Height)

The heights of pedestrians increase as their foot locations are further from the top of a video frame. The rate of increase is inversely proportional to the camera tilt angle; the smaller the tilt angle, the greater the height of pedestrians at the bottom of the frame compared to those towards the top. It was found that a significantly large proportion of non-pedestrian instances in $\mathcal{M}$ had different heights compared to pedestrians at the same location (see Figure 3.7). This TSF is placed at the beginning of the sequence because of its consistent ability to reject the largest number of non-pedestrians across most datasets. Implementation details are as follows:

Number of pruners: 1

Pruner implementation

[CBA: location based height, feature: sample bounding box]

Given that the Inception stage acquires *N* samples from environment ϵ, $\{x_i^{\epsilon}\}_{i=1}^{N}$, via motion detection, for any particular sample, denote its bounding box by the top-left corner, (x1,y1) and bottom-right corner, (x2,y2), denote its height by *z* = y2-y1 and denote its foot location by *l* = y2. Generate the set of pairs, $\{l_i^{\epsilon}, z_i^{\epsilon}\}_{i=1}^{N}$, where, l_i^{ϵ} is the foot location and $z_i^{\epsilon}$is the height of the *i*th sample. Divide the frame height, *h*, into *k* intervals, where, *k* = *h*/*r*. *r* is empirically set to 20 pixels. Calculate the set of points, $\{l\_mid_j^{\epsilon}, z\_mod_j^{\epsilon}\}_{j=1}^{k}$, where, $z\_mod_j^{\epsilon}$ is the modal height in the interval (*j*-1)*r* – *jr*, and $l\_mid_j^{\epsilon}$ is the

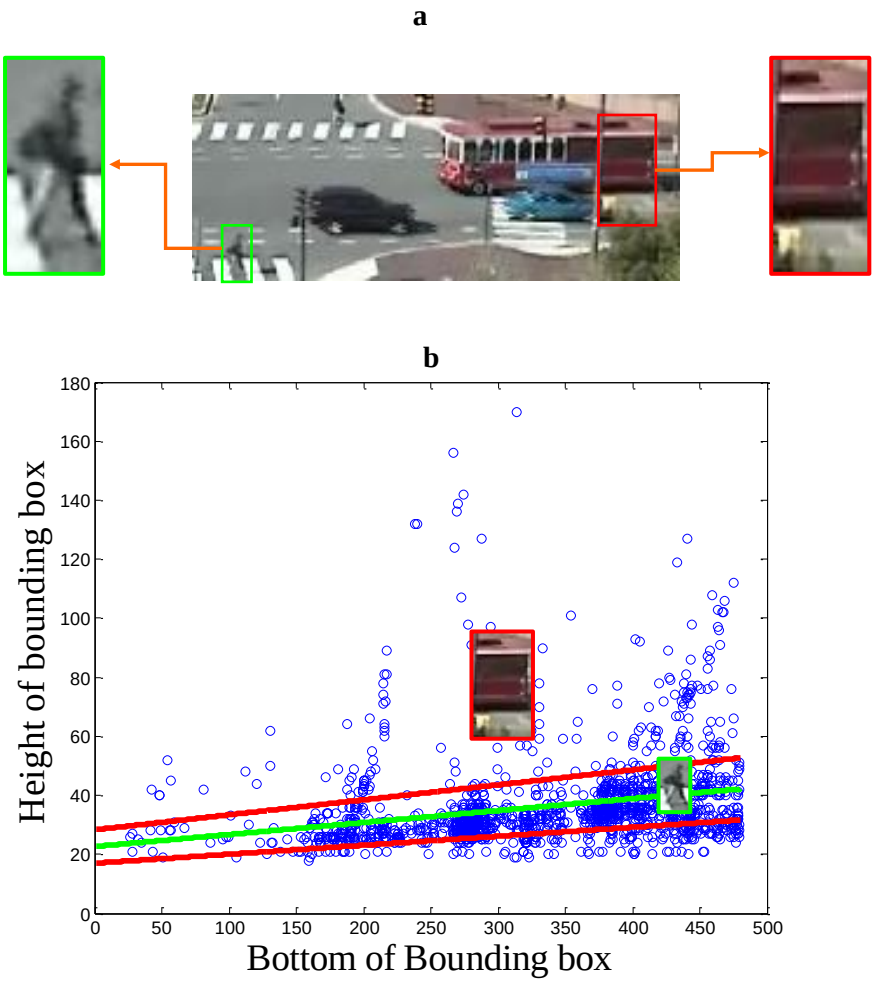


Figure 3.7: Visualization of the Height TSF. a) Examples of sample acquisitions b). Plot of height of bounding box against location of bounding box for acquired samples. Green line indicates the linear model of height vs location and red lines indicate the upper and lower bounds of accepted variations. The projections of the samples from a) onto b) indicate their respective positions on the plot.

midpoint of that interval. Use linear regression to construct a linear model from these points and apply the model to calculate the expected heights, $\{z\_exp_v^{\mathfrak{C}}\}_{v=1}^{h}$, for all vertical locations, $\{v\}_{v=1}^{h}$ (see Figure 3.7).

Rejection Criteria: $\left|z_i^{\mathfrak{C}} - z\_exp_{l_i}^{\mathfrak{C}}\right| > 0.25 \times z\_exp_{l_i}^{\mathfrak{C}}$, where, $z_i^{\mathfrak{C}}$ and $z\_exp_{l_i}^{\mathfrak{C}}$ are the observed height and expected height, respectively, at the foot location, $l_i^{\mathfrak{C}}$ of the *i*th sample. 0.25 is a conservative ratio, determined empirically.

3.4.2.2 TSF 2 – Grayscale analysis (GraySc)

It was observed that a considerable number of non-pedestrian instances in $\mathcal{M}$ were obtained as a result of sporadic scene changes, rather than objects actually traversing. Some examples are movement of trees or changing regions on roads/buildings due to lighting fluctuations. These instances have limited intra-object intensity variations compared to pedestrians (see Figure 3.8). Implementation details are as follows:

Number of pruners: 1

Pruner implementation

[CBA: grayscale variance, feature: grayscale]

For every sample $x_i^{\mathfrak{C}}$, denote its height by *h*, its width by *w* and its grayscale image by $g_i^{\mathfrak{C}}$. Calculate

$$z_i^{\mathfrak{C}} = \sum_{j=1}^{div\_w} Var(g_i^{\mathfrak{C}}[W_j]) + \sum_{j=1}^{div\_h} Var(g_i^{\mathfrak{C}}[H_k]) \tag{3.8}$$

where, div_w is empirically set at 15 and W_j denotes the *j*th vertical partition with pixel co-ordinates (x,y), such that,

$$\frac{w}{div\_w}(j-1) < \mathrm{x} \leq \frac{w}{div\_w} j \quad \text{and} \quad 0 < \mathrm{y} \leq h$$

and where, div_h is empirically set at 20 and H_k denotes the *k*th horizontal partition with pixel co-ordinates (x,y), such that,

$$0 < \mathrm{x} \leq w \quad \text{and} \quad \frac{h}{div\_h}(k-1) < \mathrm{y} \leq \frac{h}{div\_h} k$$

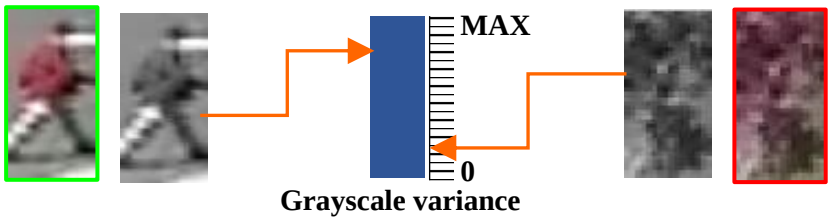


Figure 3.8: Visualization of the GraySc TSF.

Rejection Criteria: $z_i^c < \mu\left(\{z_i^c\}_{i=1}^N\right) - 1.5\sigma\left(\{z_i^c\}_{i=1}^N\right)$

3.4.2.3 TSF 3 – Foreground analysis (ForeGd)

The binary foreground of moving pedestrians contains some very generic structural characteristics that are easily exploitable, but at the same time adequately specific to pedestrians. These include the presence of an identifiable blob corresponding to the pedestrian head, the ratio of head to torso and normally central alignment of the pedestrian foreground in the window (see Figure 3.9). A combination of these characteristics can be used to differentiate and reject the more challenging non-pedestrian instances in $\mathcal{M}$, such as cropped/misaligned pedestrians, portions of vehicles or complex agglomerations of multiple non-pedestrian moving objects. For small scale pedestrians, it may become very difficult to perform any of the aforementioned head analysis if the head region is too small. Therefore, for this particular TSF, all samples are resized to 200 pixels tall. Implementation details are as follows:

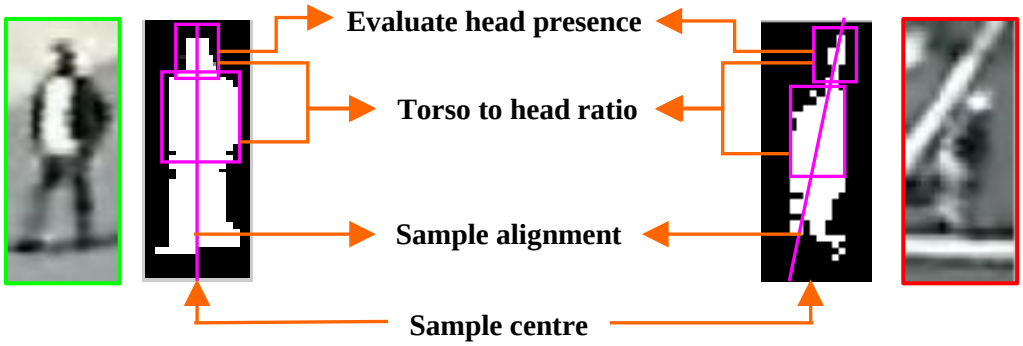


Figure 3.9: Visualization of the ForeGd TSF. Note the different structural characteristics of pedestrians that can be exploited using foreground to differentiate from non-pedestrians

Number of pruners: 4, configuration: series

For every sample x_i^{ϵ}, denote its height by *h* (set to 200), its width by *w* and its binary foreground image by b_i^{ϵ}. It is widely accepted that an average person is generally 7.5 heads tall [121], where the head and torso together make up 4 heads. The head is expected to be in the upper 1/7.5 portion of the sample, which corresponds to the vertical range of 0 - 0.133*h* in b_i^{ϵ}. Using connected component analysis, find and assume the largest blob in this range to be the head. Denote the centroid of this blob by $(\mathrm{x\_head}_i^{\epsilon}, \mathrm{y\_head}_i^{\epsilon})$.

Pruner 1 implementation

[CBA: head-region presence, feature: binary foreground]

Define the head analysis region of the sample by the bounding box co-ordinates, (x1,y1) and (x2,y2), where, y1 = 1, y2 = 0.133*h*, x1 = $\mathrm{x\_head}_i^{\epsilon} - 0.133h/2$ and x2 = $\mathrm{x\_head}_i^{\epsilon} + 0.133h/2$. The analysis region is then a square of length *l* = y2 – 1 = x2 – x1. Calculate

$$z_i^{\epsilon} = \left(\sum_{k=y1}^{\mathrm{y2}} \sum_{j=x1}^{\mathrm{x2}} b_i^{\epsilon}(j,k) \right) \Big/ (\mathrm{y2} - \mathrm{y1})(\mathrm{x2} - \mathrm{x1}) \tag{3.9}$$

Rejection Criteria: $z_i^{\epsilon} < \mu\left(\{z_i^{\epsilon}\}_{i=1}^{N}\right) - 1.5\sigma\left(\{z_i^{\epsilon}\}_{i=1}^{N}\right)$

Pruner 2 implementation

[CBA: head-region consistency, feature: binary foreground]

Denote *n* as the number of analysis points. *n* is set to 0.133*200 = 26 (*h* = 200). Using the bounding box notations defined in Pruner 1 implementation, calculate

$$\{z_i^{\epsilon}(m)\}_{m=1}^{n} = \left\{ \sum_{k=1}^{w} b_i^{\epsilon}\left(k, m\frac{l}{n}\right) \Big/ \sum_{j=1}^{\mathrm{y2}} b_i^{\epsilon}\left(\mathrm{x1} + m\frac{l}{n}, j\right) \right\}_{m=1}^{n} \tag{3.10}$$

Note that for *l* < 26pixels, the same horizontal and/or vertical locations may processed more than once as *l* < *n*.

Calculate the errors as

$$err_i^{\epsilon}(m) = \begin{cases} 1\text{, if } \left| z_i^{\epsilon}(m) - \mu\left(\{z_i^{\epsilon}(m)\}_{i=1}^{N}\right)\right| > 1.5\sigma\left(\{z_i^{\epsilon}(m)\}_{i=1}^{N}\right) \\ 0\text{, otherwise} \end{cases} \quad (3.11)$$

Rejection Criteria: $\sum_{m=1}^{d} err_i^{\epsilon}(m) > 0.33n$

Pruner 3 implementation

[CBA: sample alignment, feature: binary foreground]

$$z_i^{\epsilon} = \text{x\_head}_i^{\epsilon} \quad (3.12)$$

Rejection Criteria: $\left|z_i^{\epsilon} - w/2\right| / (w/2) > 0.5$

Pruner 4 implementation

[CBA: torso-head ratio, feature: binary foreground]

According to the previously mentioned human body proportions, the head and torso constitute 4 heads. The vertical range for head is 0 – t1, where t1 = (1/7.5)h , and the vertical range for torso is t2 – t3, where, t2 = (1.5/7.5)h and t3 = (3.5/7.5)h. Calculate

$$z_i^{\epsilon} = \frac{1}{\text{t3} - \text{t2}} \sum_{b=\text{t2}}^{\text{t3}} \sum_{k=1}^{w} b_i^{\epsilon}(k,b) \Big/ \frac{1}{\text{t1}} \sum_{a=1}^{\text{t1}} \sum_{k=1}^{w} b_i^{\epsilon}(k,a) \quad (3.13)$$

Rejection Criteria: $\left| z_i^{\epsilon} - \mu\left(\{z_i^{\epsilon}\}_{i=1}^{N}\right)\right| > 1.5\sigma\left(\{z_i^{\epsilon}\}_{i=1}^{N}\right)$

3.4.2.4 TSF 4 – Template analysis (Temp)

The previous three TSFs successfully reject a large majority of the non-pedestrian instances; consequently, the remaining samples are mostly pedestrians. This predominance can be exploited to generate an average template that is bound to be highly representative of pedestrians and then identify those instances with insufficient similarity using template matching approaches (see Figure 3.10). Implementation details are as follows:

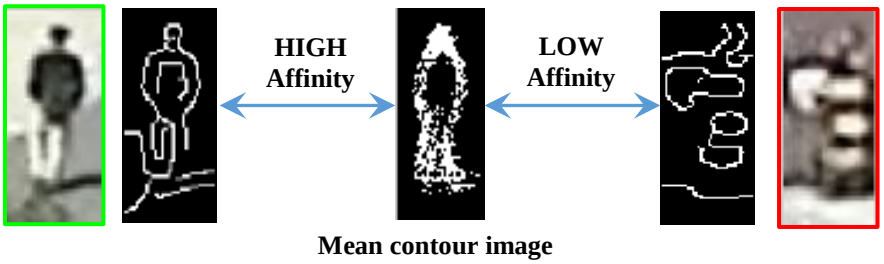


Figure 3.10: Visualization of the Temp TSF.

Number of pruners: 2, configuration: parallel

Pruner 1 implementation

[CBA: affinity to average template, feature: sample edges]

For every sample x_i^{ϵ}, use any edge detection algorithm (Canny is used here) to extract the edge image, c_i^{ϵ}. Compute the average of all edge images as $c\_mean^{\epsilon} = \mu\left(\{c_i^{\epsilon}\}_{i=1}^{N}\right)$. Denote the set of foreground pixels in c_i^{ϵ} by $P = \{p_k\}_{k=1}^{N}$ and the set of foreground pixels in c_mean^{ϵ} as $Q = \{q_k\}_{k=1}^{M}$. Calculate the modified hausdorff [122] distance between each sample and the average edge image

$$z_i^{\epsilon} = \max\{\boldsymbol{d}(P,Q)\,, \boldsymbol{d}(Q,P)\}, \tag{3.14}$$

where, $\boldsymbol{d}(P,Q) = (1/N)\sum_{k=1}^{N}\boldsymbol{d}(p_k,Q)$, such that, $\boldsymbol{d}(p_k,Q) = \min_{q\in Q}\|p_k,q\|$,

and, $\boldsymbol{d}(Q,P) = (1/M)\sum_{k=1}^{M}\boldsymbol{d}(q_k,P)$, such that, $\boldsymbol{d}(q_k,P) = \min_{p\in P}\|q_k,p\|$.

Rejection Criteria: $z_i^{\epsilon} > \mu\left(\{z_i^{\epsilon}\}_{i=1}^{N}\right) + 1.5\sigma\left(\{z_i^{\epsilon}\}_{i=1}^{N}\right)$

Pruner 2 implementation

[CBA: affinity to average template, feature: sample contour/perimeter]

For every sample x_i^{ϵ}, obtain the contour or perimeter of the sample, u_i^{ϵ}. Repeat all steps outlined in Pruner 1 implementation.

3.4.3 Training the initial detector

Recall that the objective of the inception stage is to train an initial detector with minimal labelling errors, to prevent error propagation through the subsequent stages. The following strategies fulfil that objective:

1) $\mathbb{X}^1$ is designed to segregate pedestrian instances with high-precision. Therefore, samples that successfully pass all TSFs of $\mathbb{X}^1$ are confidently labelled as the initial positive samples.
2) The above high-precision labelling consequently causes a number of pedestrian instances to be incorrectly rejected. Therefore, the samples that are rejected by $\mathbb{X}^1$ are not labelled as initial negative samples to avoid mislabelled training samples. Rather, step 3) is executed to acquire the initial negative samples.
3) The training sequence is sampled at regular intervals to form a sequence of k frames $\{\mathbb{E}_i^{-ve}\}_{i=1}^{k}$ to be utilized for bootstrapping hard negatives. The initial negatives are acquired by densely sampling the non-motion regions in the first frame, $\mathbb{E}_1^{-ve}$, only. Instead of performing multi-scale sampling, the height model learnt from the TSF 1 in $\mathbb{X}^1$ is employed to determine the window size to be sampled at a given location based on the expected pedestrian height at that location. The initial positives and the initial negatives are then combined to train the initial detector, $\mathbb{K}^{Initial}$.

The rejected samples may contain valuable pedestrian instances that can be utilized for improving the performance of the detector. Therefore, the rejects are not discarded; they are stored and revisited in Stage 2 to reacquire the incorrectly rejected pedestrian instances.

3.5 VAT Stage 2: Bootstrapping

This section provides full implementation details for the second stage of VAT: Bootstrapping. The reader is advised to frequently refer back to Figure 3.3 and Algorithm 1 as they progress through this section to enable a clearer understanding of the connections between various components/steps as well as their relevance to the overall framework.

3.5.1 Retraining with hard negatives

Bootstrapping is the iterative retraining of a detector with “hard negatives”. It is an important technique employed in pedestrian detector training to effectively reduce the false alarm rate by focusing on negative samples that lie close to the decision boundary. The standard approach for training a pedestrian detector involves two steps [40, 42, 44] . First, the detector is trained using a dataset that comprises of a large number of pedestrian and non-pedestrian instances. Next, this detector is applied, or “bootstrapped” to background images containing no pedestrians, the obtained detection responses are augmented as hard negatives to the original datasets and the detector is retrained.

Despite the significance of bootstrapping in training pedestrian detectors, the literature on ideal bootstrapping practices is very limited [123]. There is no comprehensive research that indicates whether it is better to search all the frames at once or update the detector after searching every frame and how many times the frames must be searched to reach convergence. As the initial detector, $\mathbf{Ж}^{Initial}$, was trained using only a small set of initial negatives sampled from a single frame and consequently has high false alarm rate, the importance of optimal bootstrapping is more critical in VAT.

Therefore, detailed studies were performed to determine the best bootstrapping methodology. It was found that bootstrapping exhibits similar convergence behaviour as gradient descent – the most common optimization algorithm used in machine learning to minimize the cost function of the trained model by iteratively updating the parameter values. There are three variants of gradient descent that differ based on the amount of data processed for updating the parameters:

- **Batch gradient descent** [124]: The training error over all samples in the dataset is computed before updating the model parameters. There is higher guarantee of convergence to the global minimum, but this stability itself can cause premature convergence to a sub-optimal set of parameters. This approach is usually slow.
- **Stochastic gradient descent** [125]: For every sample in the dataset, the training error is computed and the model parameters are updated. The frequency of updates can result in a noisy learning process, which can cause the parameters to jump around and consequently prevent convergence to the global minimum. This approach is usually fast.

- **Mini-batch gradient descent** [126]: The training dataset is split into small batches and for each batch of samples, the training error is computed and the model parameters are updated. This approach creates a balance between the convergence abilities of batch gradient descent and speeds of stochastic gradient descent and is therefore the most widely used variant of the three. The batch size is a deciding factor on the speed and accuracy of the learning process.

Analogously, $\mathbf{Ж}^{Initial}$ can be bootstrapped on all the frames at once, one at a time or small batches of frames. It turned out that if all the frames are searched at once, convergence is certain, but the detection rate is undesirably suppressed. On the other hand, if each frame is searched individually, the detection rate fluctuates and does not converge to a stable optimum. The ideal approach proved to be the intermediate one based on small batches. Accordingly, the bootstrapping method was devised as follows. Given k frames sampled from the target surveillance environment, (k-1) frames are used for bootstrapping (the first of k frames is sampled for initial negatives during Inception) and divided into s segments, each consisting of (k-1)/s frames, fr. Bootstrapping commences by initializing the retrained detector $\mathbf{Ж}^{HN}$ with $\mathbf{Ж}^{Initial}$. In each retraining iteration, the non-motion regions in the next fr frames are searched with $\mathbf{Ж}^{HN}$, the acquired hard negatives are augmented to the dataset and $\mathbf{Ж}^{HN}$ is retrained. A single bootstrapping round is completed when all s segments are sequentially processed. The bootstrapping procedure either concludes when r bootstrapping rounds are repeated, or terminates if no hard negatives are acquired for 3 consecutive retraining iterations. Detailed experimental results for all three approaches are presented in subsection 4.3.2.

3.5.2 Retraining with hard positives

Unlike standard bootstrapping that involves searching for hard negatives only, the bootstrapping stage of VAT additionally retrains the detector with hard positives. In the context of VAT, the hard positives are the pedestrian instances that were incorrectly rejected (rejected positives) by $\mathbf{Ж}^{1}$, but not discarded. Retraining with these rejected positives serves dual purposes. Firstly, by revisiting the rejected samples to search for the rejected positives, the valuable information in the acquired rejected positives can be exploited to add improvements to the detector rather than being wasted. Secondly, despite selection of

the optimal bootstrapping strategy detailed in subsection 3.4.1, multiple iterations of retraining with hard negatives will inflict some suppression on the detection rate of $Ж^{HN}$ that is most likely inevitable; retraining with the acquired rejected positives can improve the detection rate while maintaining the achieved reduction in false alarm rate from previous step (retraining with hard negatives). This step is implemented by applying $Ж^{HN}$ to the samples rejected by X^{1}, and those with positive classification scores are reacquired and augmented to the training set to train $Ж^{RP}$.

3.6 VAT Stage 3: Finalization

This section provides full implementation details for the third stage of VAT: Finalization. The reader is advised to frequently refer back to Figure 3.3 and Algorithm 1 as they progress through this section to enable a clearer understanding of the connections between various components/steps as well as their relevance to the overall framework.

3.6.1 Detection-based sample acquisition

Though motion-based sample acquisition is more robust in difficult surveillance environments and can achieve better sample alignment, it does not have localization ability like a pedestrian detector. When people move in a group or close to other moving objects like vehicles, the motion regions merge. Under such circumstances, the Inception stage has no means of localizing the pedestrian instances in these merged regions. These regions are either ignored based on aspect ratio, or rejected by Oracle-1.The Inception stage is designed to only acquire those pedestrian instances that are the sole moving objects in their corresponding motion regions. The ignored or rejected patches is highly likely to contain multiple true positives and false positives that could be exploited to further enhance the detector. Furthermore, as the Bootstrapping stage only searches the non-motion regions for hard negatives, all potential false positives in the motion regions that could have been acquired as hard negatives are ignored.

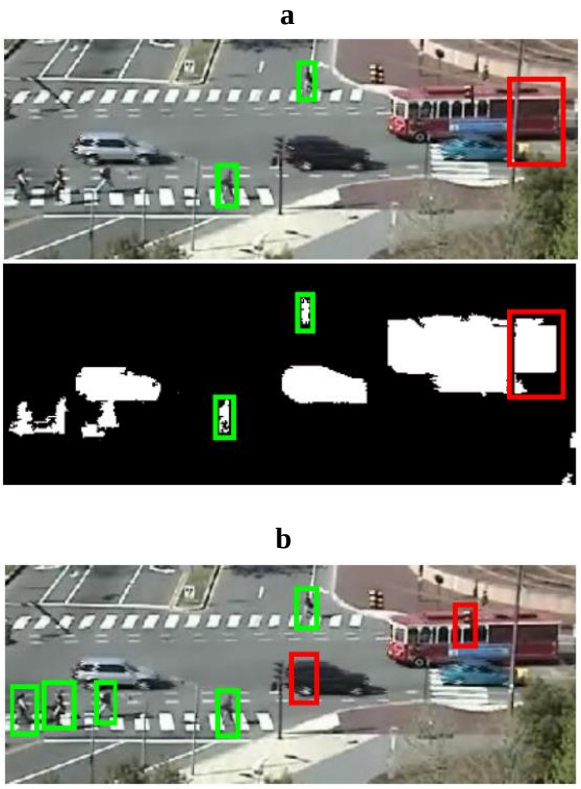


Figure 3.11: Samples acquisition during a) Inception, by applying motion detection and b) Finalization, by applying $\mathbf{Ж}^{RP}$. Green bounding boxes denote true positives and red boxes denote false positives

To acquire the samples missed by the Inception and Bootstrapping stages, the Finalization stage applies the detector, $\mathbf{Ж}^{RP}$ to the training sequence and extracts samples with positive responses, $\mathfrak{R}$. Figure 3.11 compares the sample acquisitions, $\mathcal{M}$, during the Inception stage and $\mathfrak{R}$, during the Finalization stage. During the Inception stage, people in groups and people near to vehicles are ignored by the motion-based acquisition because the merged motion regions have large aspect ratio (See the image and its foreground in Figure 3.11a). However, by densely scanning the whole frame, $\mathbf{Ж}^{RP}$ is able to localize most of the previously missed pedestrians (see Figure 3.11b). Additionally, the new false positives (red bounding boxes in Figure 3.11b) amongst $\mathfrak{R}$ are valuable novel negatives, as they were not acquired in the previous stages, either because the motion region was rejected during Inception (as mentioned in the previous paragraph) or it was not searched during Bootstrapping.

3.6.2 Oracle-2

The samples in $\mathcal{M}$, acquired in the Inception stage were based on motion and therefore, have minimal correlation amongst themselves. Contrastingly, $\mathfrak{R}$ comprises of detection responses acquired by applying the detector $\mathbf{Ж}^{RP}$ to the target surveillance environment. This means that the non-pedestrian instances have substantial correlation with the pedestrian instances; else they would not have been mistakenly acquired as detections responses to begin with. Therefore, this does not only mandate (as per the guidelines in subsection 3.2.4) the design of a new oracle, but one that is composed of more discriminative pruners relative to those in $\mathbb{X}^1$, in order to reliably segregate the pedestrians from non-pedestrians amongst the detection responses.

Furthermore, as Finalization is the last stage of VAT, the rejected samples cannot be discarded; rather, they must be augmented to the training dataset as final negatives in order to maximize exploitation of scene-specific information. Therefore, this second oracle cannot be designed with merely high-precision as the objective, as this would result in a large number of incorrectly rejected pedestrian instances. A good balance of precision and recall must be targeted, which may require the oracle to be constructed using fewer TSFs.

Taking the afore-mentioned factors into consideration, similar to Oracle-1, $\mathfrak{R}$ from different datasets (see Section 4.1) were jointly examined and the guidelines from subsection 3.2.4 were followed to design Oracle-2 ($\mathbb{X}^2$) as a combination of three TSFs. The full implementation details of these TSFs are presented in the next three subsections. Note that the value of 1.5σ used in the rejection criteria of the pruners has been determined empirically.

3.6.2.1 TSF 1 – Vertical structures analysis (VerStrct)

Amongst the most salient features of pedestrians are the long edges extracted from the limbs (arms and legs). Therefore, there are various false positives in $\mathfrak{R}$ acquired by the detector $\mathbf{Ж}^{RP}$ because of the presence of such long edges, such as portions of vehicles or buildings. However, unlike the limbs of pedestrians whose vertical length may extend to only half the total height of the pedestrian at best, those from vehicles/buildings are much longer (see Figure 3.12). Implementation details are as follows:

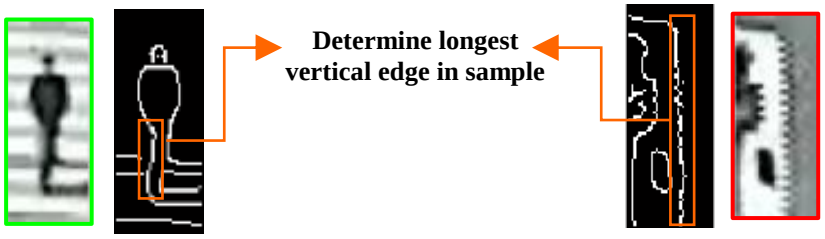


Figure 3.12: Visualization of the VerStrct TSF

Number of pruners: 2, configuration: parallel

Pruner 1 implementation

[CBA: longest vertical edge, feature: edges]

For every sample x_i^{ϵ}, denote its height by *h*, its width by *w* and use Canny edge detection algorithm to extract the edge image, c_i^{ϵ}. Divide c_i^{ϵ} into *m* partitions, where *m* = *w/b*. *b* is the partition width, set to 2 pixels. Calculate

$$\{z\_full_i^{\epsilon}(l)\}_{l=1}^{m} = \left\{ \sum_{j=1+b(l-1)}^{bl} \sum_{k=1}^{h} c_i^{\epsilon}(j,k) \right\}_{l=1}^{m} \tag{3.15}$$

$$\text{and } z\_full_i^{\epsilon} = \max\left(\{z\_full_i^{\epsilon}(l)\}_{l=1}^{m} \right)$$

Every sample has vertical and horizontal context margin around it. Assume the vertical top and bottom vertical margins to be 7.5%, and split $c_i^{\epsilon}(j,k)$ into three vertical ranges of 0-0.075*h*, 0.075*h* – 0.925*h* and 0.925*h* – *h*. Calculate

$$\{z\_mid_i^{\epsilon}(l)\}_{l=1}^{m} = \left\{ \sum_{j=1+b(l-1)}^{bl} \sum_{k=0.075h}^{0.925h} c_i^{\epsilon}(j,k) \right\}_{l=1}^{m} \tag{3.16}$$

$$z\_mid_i^{\epsilon} = \max\left(\{z\_mid_i^{\epsilon}(l)\}_{l=1}^{m} \right)$$

Construct $\{i_t, l_t\}_{t=1}^{n}$, where, i_t is the index of the *t*th sample that satisfies the condition $z\_mid_i^{\epsilon} > \mu\left(\{z\_mid_i^{\epsilon}\}_{i=1}^{N}\right) + \sigma\left(\{z\_mid_i^{\epsilon}\}_{i=1}^{N}\right)$ and l_t is the partition where the longest edge was found. Calculate

$$\{z\_top_{i_t}^{\epsilon}\}_{t=1}^{n} = \left\{ \sum_{j=1+b(l_t-1)}^{bl_t} \sum_{k=1}^{0.075h} c_{i_t}^{\epsilon}(j,k) \right\}_{t=1}^{n} \tag{3.17}$$

$$\left\{z\_bottom_{i_t}^{\epsilon}\right\}_{t=1}^{n} = \left\{ \sum_{j=1+b(l_t-1)}^{bl_t} \sum_{k=0.925h}^{h} c_{i_t}^{\epsilon}(j,k) \right\}_{t=1}^{n} \quad (3.18)$$

Rejection Criteria: $z\_full_i^{\epsilon} > \mu\left(\left\{z\_full_i^{\epsilon}\right\}_{i=1}^{N}\right) + 1.5\sigma\left(\left\{z\_full_i^{\epsilon}\right\}_{i=1}^{N}\right)$

OR $z\_top_{i_t}^{\epsilon} > \mu\left(\left\{z\_top_{i_t}^{\epsilon}\right\}_{t=1}^{n}\right) + 1.5\sigma\left(\left\{z\_top_{i_t}^{\epsilon}\right\}_{t=1}^{n}\right)$

OR $z\_bottom_{i_t}^{\epsilon} > \mu\left(\left\{z\_bottom_{i_t}^{\epsilon}\right\}_{t=1}^{n}\right) + 1.5\sigma\left(\left\{z\_bottom_{i_t}^{\epsilon}\right\}_{t=1}^{n}\right)$

Pruner 2 implementation

[CBA: longest vertical edge, feature: gradient]

For every sample x_i^{ϵ}, denote its height by *h*, its width by *w* and extract the gradient image, g_i^{ϵ}. Set *b* to 3 pixels and repeat all steps outlined in Pruner 1 implementation.

3.6.2.2 TSF 2 – Masked gradient Analysis (MskGrad)

Gradients are, in general, the most discriminative features used in training pedestrian detectors and are usually concentrated along the contour of pedestrians. For this reason, it is likely that, a considerable number of non-pedestrian instances will be acquired because the combination of their extracted gradients are holistically quite similar to the gradients along the contour of pedestrians. Such false positives will not be rejected by the previous TSF if long edges are absent. To differentiate such instances from pedestrians, a region enclosing pedestrians can be approximated and the gradient concentration inside this region can be exploited (see Figure 3.13). Implementation details are as follows:

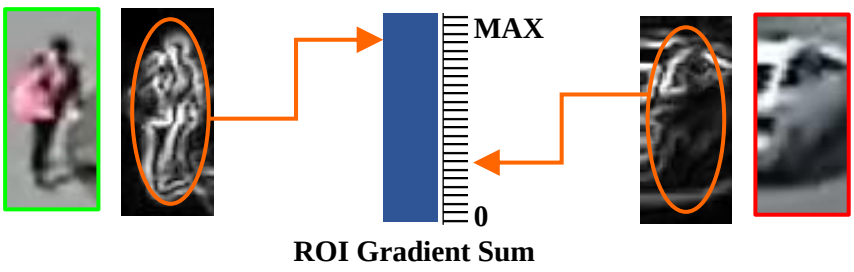


Figure 3.13: Visualization of the MskGrad TSF

Number of pruners: 1

Pruner implementation

[CBA: gradient concentration, feature: gradient]

For every sample $x_i^{\mathfrak{C}}$, denote its height by *h*, its width by *w* and extract the gradient image, $g_i^{\mathfrak{C}}$. Create a binary elliptical mask, *E*, to isolate the area of analysis. *E* has similar dimensions as $x_i^{\mathfrak{C}}$; all regions within the ellipse are assigned a Boolean "true" and a Boolean "false" is assigned everywhere else. The major axis of *E* is along the y-axis from $0.075h$ – $0.925h$ and the minor axis is along the x-axis from $0.075w$ – $0.925w$. Centre *E* on $g_i^{\mathfrak{C}}$ and perform masking by $g_i^{\mathfrak{C}} = g_i^{\mathfrak{C}} \times E$. Calculate

$$z_i^{\mathfrak{C}} = \sum_{k=1}^{h} \sum_{j=1}^{w} g_i^{\mathfrak{C}}(j,k) \tag{3.19}$$

Rejection Criteria: $z_i^{\mathfrak{C}} < \mu\left(\{z_i^{\mathfrak{C}}\}_{i=1}^{N}\right) - 1.5\sigma\left(\{z_i^{\mathfrak{C}}\}_{i=1}^{N}\right)$

3.6.2.3 TSF 3 – SIFT analysis (SIFT)

This TSF operates similar to TSF 4 (Temp) of $\mathbb{X}^1$ because it aims to exploit the predominance of pedestrian instances. However, the rejection logic must be relatively more complex due to higher correlation between the pedestrian and non-pedestrian instances. The scale-invariant feature transform or SIFT [127] is a powerful feature extraction algorithm that extracts object keypoints and computes their descriptors. The SIFT features of two objects can be matched to measure their similarity. If each sample is matched with every other sample using SIFT features, two useful trends will surface (see Figure 3.14). Firstly, as pedestrians are the dominant class, the pedestrian instances will have a much larger number of matches. Secondly, difficult non-pedestrian instances may also have a large number of matches; however, if the actual scores of their matches are examined, there should be extremely few matches with high scores. Implementation details are as follows:

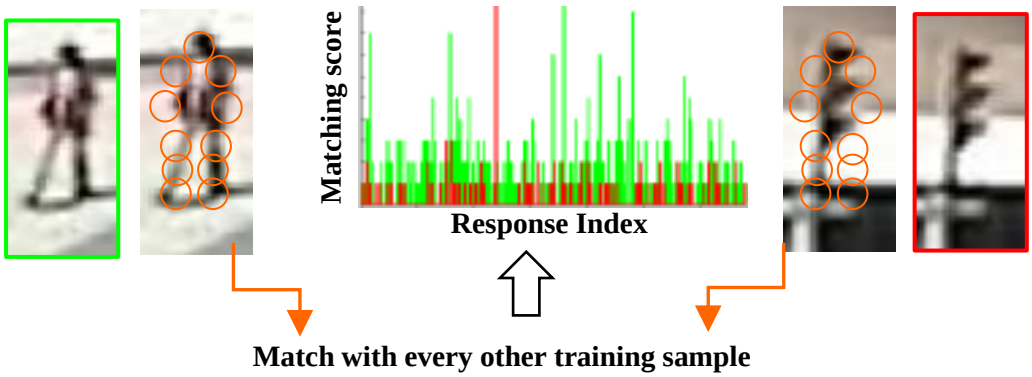


Figure 3.14: Visualization of the SIFT TSF. The green plot shows the matching scores between the pedestrian instance and all the other samples based on SIFT features. Similarly, the red plot shows the matching scores for the non-pedestrian instance. The non-pedestrian instance is a difficult one due to its similarity with pedestrians. Notice how even though it matches with a good number of samples, the non-pedestrian instance has only one high matching score. Comparatively, the pedestrian instance has several high matching scores.

Number of pruners: 2, configuration: series

Given a sample, x_i^{ϵ}, denote its height by *h*, its width by *w*. Form a grid of pixel locations, $\mathbb{G} = \{(\mathbf{x}d - 3\,, \mathbf{y}d - 3)\}_{\mathbf{x}=1}^{a}{}_{\mathbf{y}=1}^{b}$, such that *a* = *w/d* and *b*=*h/d*. *d* is set to 6 pixels. Create custom SIFT keypoint frames, centred at each pixel location in $\mathbb{G}$. Set the scale for every custom keypoint frame to 3, and leave the orientation unspecified. For every x_i^{ϵ}, extract the set of keypoint descriptors, $\mathrm{K}_i^{\epsilon} = \{k_i^{\epsilon}(t)\}_{t=1}^{k\_count}$ from the above custom keypoint frames by applying SIFT. Note that $k\_count = ab$. Create the similarity matrix $\mathrm{S}^{\epsilon} = \{s_{ij}^{\epsilon}\} \in \mathbb{R}^{N\times N}$, where s_{ij}^{ϵ} is in itself a 2-D array that contains indices of the matching keypoints between K_i^{ϵ} and K_j^{ϵ} . s_{ij}^{ϵ} is generated by searching for the closest matching keypoint descriptor in K_j^{ϵ} for every keypoint descriptor in K_i^{ϵ}. The closeness is measured by computing the L2 norm between two keypoints. Denote $s_{ij}^{\epsilon}\_rows$ as the number of matches between x_i^{ϵ} and x_j^{ϵ}, such that $0 \leq s_{ij}^{\epsilon}\_rows \leq k\_count$, where 0 indicates no matches found and k_count indicates that all descriptors were matched. For the *n*th match entry in s_{ij}^{ϵ}, $s_{ij}^{\epsilon}(n,1)$ is the index of the descriptor in K_i^{ϵ} and $s_{ij}^{\epsilon}(n,2)$ is the index of the descriptor in K_j^{ϵ} that matched each other, i.e. $k_i^{\epsilon}(s_{ij}^{\epsilon}(n,1))$ matches $k_j^{\epsilon}(s_{ij}^{\epsilon}(n,2))$.

Pruner 1 implementation

[CBA: inter-sample similarity extent, feature: SIFT Keypoints]

For every sample x_i^{ϵ}, calculate the number of samples that it matched with by computing

$$z_i^{\epsilon} = \sum_{j=1}^{N} m_{ij}^{\epsilon} \tag{3.20}$$

$$\text{where}, m_{ij}^{\epsilon} = \begin{cases} 1\text{ , if } s_{ij\_rows}^{\epsilon} > 0 \\ 0\text{ , if } s_{ij\_rows}^{\epsilon} = 0 \end{cases}$$

Rejection Criteria: $z_i^{\epsilon} < \mu\left(\{z_i^{\epsilon}\}_{i=1}^{N}\right) - 1.5\sigma\left(\{z_i^{\epsilon}\}_{i=1}^{N}\right)$

Pruner 2 implementation

[CBA: false similarity extent, feature: SIFT Keypoints]

For every sample x_i^{ϵ}, find the highest number of matches with another sample by computing

$$m_i^{\epsilon} = \max(\{s_{ij\_rows}^{\epsilon}\}_{j=1}^{N}) \tag{3.21}$$

Rejection Criteria: $\left(z_i^{\epsilon} < \mu\left(\{z_i^{\epsilon}\}_{i=1}^{N}\right)\right) \cap \left(m_i^{\epsilon} > \mu\left(\{m_i^{\epsilon}\}_{i=1}^{N}\right) + 1.5\sigma\left(\{m_i^{\epsilon}\}_{i=1}^{N}\right)\right)$

3.6.3 Training the final detector

The confidence scores of the detection responses, $\mathfrak{R}$, can be utilized to increase the labelling reliability of $\mathbb{X}^2$. A confidence barrier is applied such that those detection responses with confidence scores greater than ($\mu^{conf} + \sigma^{conf}$) cannot be rejected, where μ^{conf} and σ^{conf} are the mean and standard deviation of the confidence scores of $\mathfrak{R}$, respectively. A high confidence score by the pedestrian detector, $\mathbf{Ж}^{RP}$, implies a very strong probability that the sample under consideration is a pedestrian and is therefore allowed to override the decisions of TSFs to reject such a sample. The segregated pedestrian instances and the

rejects are augmented to the training set as the final positives and final negatives, respectively, to train the final detector, $\mathbf{Ж}^{Final}$, which is ultimately assigned as the scene-specific detector, $\mathbf{Ж}^{§}$, trained by VAT.

4 Experimental Results

In real-world surveillance environments, there is a continuous incoming stream of video frames from the surveillance cameras. Therefore, if VAT were to be applied to a target surveillance environment, it would utilize a sequence of video frames to train the scene-specific pedestrian detector, which could then be deployed to detect pedestrians in subsequent frames. As a concrete example, a surveillance system executes VAT at 10.00 a.m. on a particular target scene. VAT takes 1 hour to finish, and generates the scene-specific pedestrian detector. From 11.00 a.m. onwards, the system can perform pedestrian detection on that target scene using the scene-specific pedestrian detector trained by VAT.

The performance of the developed VAT framework is thoroughly evaluated using 10 static video surveillance datasets with different levels and combinations of scene-specific difficulties. To model real-world scenarios as described above, each dataset is split into training and testing sets. For each dataset, VAT is applied to the training set, which is created from the former portion of the dataset, and the trained scene-specific pedestrian detector is subsequently evaluated on the testing set, which is constructed from the latter portion of the same dataset. VAT comprises of various modules and training stages (see Figure 3.3); therefore detailed performance analysis of these components is necessary to make the evaluation of the VAT framework complete and to provide reasons, if necessary, for the performance of the final scene-specific pedestrian detector **Ж**[§]. Accordingly, the performed experiments were:

- Evaluation of labelling performance for both the designed oracles, Oracle-1 and Oracle-2, as well as the individual TSFs comprising each oracle.
- Evaluation of the detectors generated by each stage of the VAT framework, and comparison of the final scene-specific detectors trained using VAT against detectors that are generic, manually trained or dependent on pre-trained generic detectors for scene-specific training.
- Comparison of VAT against several state-of-the-art scene-specific approaches on the two most commonly used datasets for testing scene-specific pedestrian detectors.

4.1 Datasets

Figure 4.1 illustrates sample frames from the datasets. Detailed specifications of the datasets, including training and testing splits, are presented in Table 4.1 and the range and distributions of pedestrian heights in each testing set are depicted using box plots in Figure 4.2. In the subsequent descriptions, the datasets are grouped according to their difficulty level.

Hard (see Figure 4.1a)

***1) MIT Traffic* (MIT)** [93]: It is a far-field video of a traffic intersection. The main challenge is that the pedestrians are very small relative to the video frame size due to the large distance between the camera and scene.

2) *CUHK Square* (CUHK) [94]: It is the most commonly used dataset for testing scene-specific pedestrian detectors. It captures a pedestrian square at a smaller camera tilt angle. A crucial point is that it is very difficult to detect pedestrians in the upper half of the video frames because there is significant degradation in image contrast as the distance from the camera increases.

3) *MONASH Frontgate* (MONASH): This dataset was entirely constructed by us. We captured the scene at the front entrance of MONASH University, Malaysia Campus. The challenges of this dataset are the slanted orientations of the pedestrians and the inter-pedestrian occlusion scenarios simulated by a group of actors.

Extremely hard (see Figure 4.1b)

We discover and introduce three extremely challenging datasets for testing scene-specific pedestrian detection.

4) *QMUL Roundabout* (QMUL-R) [102]: It captures a traffic roundabout at low resolution and has the worst image quality amongst all the test datasets.

***5) QMUL Junction* (QMUL-J)** [92] : It captures a traffic junction and is the most difficult dataset due to the combination of poor image quality, low resolution, small pedestrian scales and severe inter-pedestrian occlusions as well as pedestrian-vehicle occlusions

a

MIT CUHK MONASH

b

QMUL-R QMUL-J KWSI

c

PETS-01 PETS-02 PETS-03 PETS-04

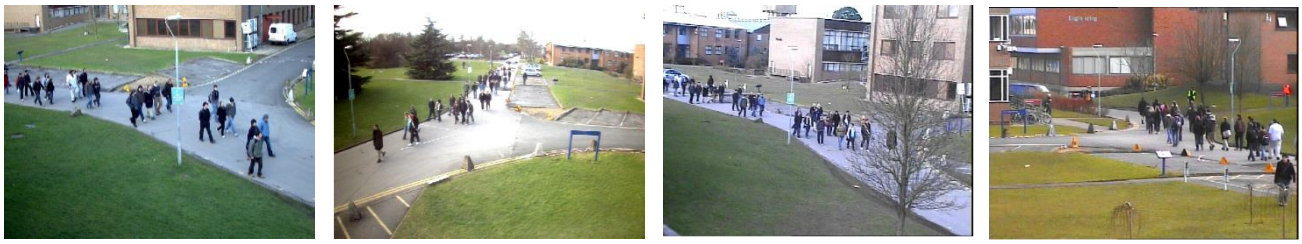


Figure 4.1: Datasets for experimental evaluation of VAT. a) Hard datasets, b) Extremely hard datasets and c) Medium datasets. Zoom in for a better assessment of scene-specific factors like pedestrian scale or image quality. Note that size difference between different datasets in this figure is not an accurate representation of their true image sizes. For the real image size, refer to the video resolution in Table 4.1.

6) Karl-Wilhelm-Straße Intersection **(KWSI)** [103] : It captures a traffic intersection from a bird's eye view similar to MIT, but with a much larger camera tilt angle and poorer image quality.

Medium (see Figure 4.1c)

PETS 2009 [95] is a relatively easier dataset because of better image quality and larger pedestrian scales (see Figure 4.2). Therefore, to make it more challenging and to increase the overall

Table 4.1: Specifications of experimental video surveillance datasets

Dataset	Video Resolution	Training Set		Testing Set		Image Quality	Occlusion Level	Overall Difficulty
		#Frame	#Pedestrian	#Frame	#Pedestrian			
MIT	720×480	420	1573	100	481	Average	Mild	***
CUHK	720×576	352	2087	100	666	Average	Mild	***
MONASH	640×480	360	801	120	626	Average	Moderate	***
QMUL-R	360×288	620	N/A	100	184	Poor	Mild	*****
QMUL-J	360×288	441	N/A	110	1270	Poor	Heavy	*****
KWSI	704×568	121	N/A	78	786	Poor	Mild	****
PETS-01	768×568	397	N/A	81	1168	Good	Moderate	**
PETS-02	768×568	397	N/A	81	2049	Good	Heavy	**
PETS-03	768×568	397	N/A	81	1921	Good	Moderate	**
PETS-04	768×568	397	N/A	81	1628	Good	Heavy	**

difficulty, we select one of the clips with heavy inter-pedestrian occlusion levels – S1.L1, at timestamp 13-59. We perform experiments on all 4 views of this clip.

7) PETS 2009 View 1 **(PETS-01):** This is the easiest of the 10 datasets; the pedestrians are large and clear, and the occlusion levels are lower relative to the other three PETS views because the scene is captured from a side-view.

8) PETS 2009 View 2 **(PETS-02):** This dataset captures the scene from a frontal view and consequently has heavy occlusions. Also, image clarity is poor near the upper portion of the video frames due to excessive brightness levels.

9) PETS 2009 View 3 **(PETS-03):** It captures the scene from a similar view-point as PETS-01, but from slightly further. However, it is far more challenging because a portion of the scene is obstructed by trees.

10) PETS 2009 View 4 **(PETS-04):** It captures the scene from a rear view, but compared to PETS-02, the camera has a smaller tilt angle and is positioned closer to the scene.

Note that all datasets, except PETS-01 – PETS-04, have the added difficulty of various moving vehicles in the scenes (see Figure 4.1). For MIT and CUHK, the training and testing sets are prepared as per the original authors. For MONASH, KWSI, QMUL-R and QMUL-J, the training and testing sets were prepared by sampling frames from the former and latter portions of the same videos, respectively. However, for all PETS datasets, due to limited scene variation in S1.L1, frames were sampled from it to prepare the testing sets only; the training sets were prepared by sampling frames from a separate clip – S0.CC, at timestamp 12-34. Excluding MIT and CUHK, the ground truth for the testing sets have been

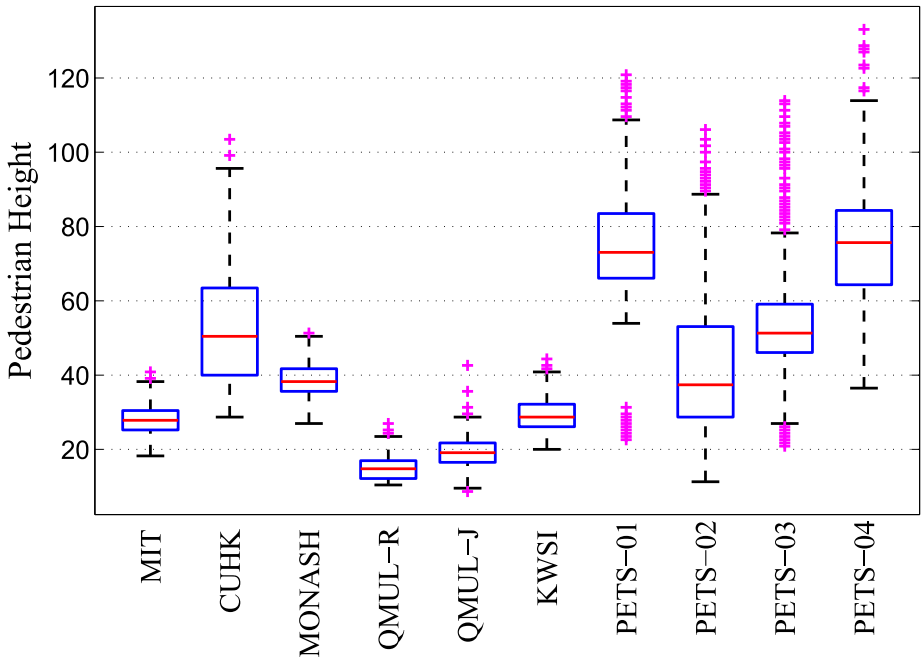


Figure 4.2: Range and distribution of pedestrian heights (in pixels) for all testing sets. Magenta crosses denote outliers

manually annotated by us (see number of pedestrians annotated in Table 4.1). In terms of the annotations for the training sets, once again the ground truth from the original authors were used for MIT and CUHK, whereas the ground truth for MONASH was manually annotated by us. However, manual annotation demands extensive amounts of time and effort. Therefore, training annotations for the remaining datasets are currently unavailable (see Table 4.1).

The 'overall difficulty' reported in Table 4.1 is a subjective metric determined by assessing critical scene-specific factors like image quality, pedestrian scale, video resolution, viewpoint, scene illumination, occlusion levels and background complexity. One of the most important factors is the image quality, which is primarily determined by the sharpness and contrast levels in the image. By comparing QMUL-R and MIT in Figure 4.1, it can be observed that the tree regions in MIT appear clear and detailed due to good sharpness and contrast levels, while the tree regions in QMUL-R have very poor detail due to poor image quality. Pedestrian scale (refer to Figure 4.2) has a profound effect on the difficulty level of a dataset - if the pedestrian scale is smaller than 30 pixels, the difficulty level can significantly increase despite good image quality and resolution (such as MIT). This is because, at such small scales, the quantity and quality of the extracted features (of the pedestrian/non-pedestrian instance)

utilized by the detector for classification degrades markedly. Lower resolution can further amplify the difficulty of a dataset that is already hampered by small scale, for reasons similar to that mentioned for small scale previously. The viewpoint affects the orientation of the pedestrians in the surveillance scene. The pedestrians in MONASH dataset have a slanted orientation, which adds to the difficulty of the dataset. Inconsistent illumination can ruin the image contrast, consequently causing an increase in missed detections. Notice the upper portion of CUHK – the excessive illumination adds unnecessary glare that affects the clarity of pedestrians in that portion of the image. PETS-02 has similar illumination problems towards the top portion of the image. In terms of occlusion levels in a dataset, they can be categorized as mild if there are pedestrians that are less than 10% occluded, moderate if there are pedestrians that are 10% -50% occluded and severe if there are pedestrians that are more than 50% occluded. Note that these occlusions could be between different pedestrians or between pedestrians and vehicles/non-moving structures. Lastly, the background complexity is a crucial factor which is determined by the extent to which the background (all regions in the image that is not occupied by pedestrians) resembles pedestrians. A more complex background would result in a higher false alarm rate. The most influential object class that increases background complexity is vehicles; thus, their absence substantially reduces background complexity which in turn, results in a considerable drop in false alarm rate.

Based on the explanations of the factors that affect the difficulty level, QMUL-R and QMUL-J are the most difficult datasets (extremely hard, 5 * in Table 4.1) due to poor image quality, small scale, low resolution and complex background. Additionally, QMUL-J has severe occlusions as well. KWSI is also categorized as “extremely hard” (4 * in Table 4.1) because of poor image quality, small scale and complex background. Each of the datasets categorized as “hard” (3 * n Table 4.1) have complex backgrounds and one additional complication – MIT has small scale, CUHK has inconsistent illumination and MONASH has non-standard viewpoint. Finally, all the PETS datasets have been categorized as “medium” (2 * in Table 4.1) because they do not have any particular complication except occlusions.

All 10 datasets have been made publicly available online in a single repository[128], under the GNU General Public License(GPL) 3.0.

4.2 Pedestrian detectors evaluated with VAT

VAT is designed to be compatible with any pedestrian detector that is based on any of the popular supervised learning algorithms like SVM, AdaBoost or deep neural networks. This is because it does not attempt to make any modifications to the selected learning algorithm itself; rather it dictates the quality of the trained detector by controlling what training samples the learning algorithm receives.

There are practical limitations of implementing VAT based on deep neural networks in real-world application scenarios. For a detailed discussion on these limitations, refer to section 5.3. Thus, deep neural networks are not tested with VAT; rather, the experimental evaluation focuses on training and testing pedestrian detectors that are based on popular real-time classifiers suitable for practical applications, namely SVM and AdaBoost.

Three different pedestrian detectors are trained and tested. This is done to demonstrate the compatibility of the VAT framework with various detectors, as well as investigate and compare the performances of various detectors when trained using the VAT framework. The feature-classifier combinations of each evaluated pedestrian detector are as follows:

***1) HOG + Linear SVM* (HOG)** [40]: For pedestrian detection, Histogram of Oriented Gradients or HOG, remains the best performing single feature, and HOG + Linear SVM is the most widely used combination.

2) *HOG-LBP + Linear SVM* (HOG-LBP) [129]: While HOG is apt for extracting shape information, local binary patterns (LBP) [130] are more suited for texture. It has been shown [34] that a combination of HOG and LBP performs consistently much better than HOG alone. The purpose of testing a second detector that is also based on SVM is to explore if the richer feature set of HOG+LBP, compared to HOG only, translates to a better scene-specific pedestrian detector using VAT relative to the generic and manual counterparts

3) *Aggregate Channel Features + AdaBoost* (ACF) [44]: ACF is a variant of the original Integral Channel Features (ICF) [41], and is one of the top-performing AdaBoost based pedestrian detectors.

4.3 Implementation details

4.3.1 Detector parameters

Full frame detection is performed using a multiscale sliding-window paradigm, with a model window size of $w \times h$ = 36×84 pixels, detection stride of 4 pixels and 8 scales/octave. Individual detector parameters are as follows:

HOG : 6×6 pixels/cell, 2×2 cells/block, block overlap 50%, 9 bins/cell unsigned, L2-Hys normalization, feature vector dimensionality is (36/6)×(84/6)×(2×2×9) = 3024.

HOG-LBP : HOG parameters are same as above. For LBP, 12×12 pix/cell, 1×1 cells/block, no block overlap, 58 uniform quantized patterns, feature vector dimensionality is (36/12)×(84/12)×58 = 1218 are used. Dimensionality of concatenated HOG-LBP feature vector is 3024+1218 = 4242.

ACF : Total of 10 feature channels comprising of normalized gradient magnitude (1 channel), HOG (6 channels) and LUV (3 channels). 3×3 pixels/block, feature vector dimensionality is (36/3)×(84/3)×10 = 3360, 2048 depth-two decision trees are trained using AdaBoost.

For HOG and LBP feature extraction, the MATLAB interface of VLFeat library [131] was utilized. The SVM classifiers for HOG and HOG-LBP were trained using LIBLINEAR [132], with C = 0.01 and L2-loss L2-regularization. ACF was implemented using the open source MATLAB toolbox from the authors [44].

4.3.2 Bootstrapping parameters

As explained in subsection 3.4.1, bootstrapping can be done on all frames at once, one frame at a time or segments of frames and repeated till convergence. To determine the optimal bootstrapping strategy, one dataset from each category of difficulty was randomly selected: MIT, QMUL-J and PETS-02 from the hard, extremely hard and medium groups, respectively. Using HOG as the detector, $\mathbf{Ж}^{Initial}$ from the Inception stage was bootstrapped with hard negatives using different number of frames to generate $\mathbf{Ж}^{HN}$. In all experiments, it turned out that three rounds of repetition were sufficient for all kinds of approaches. $\mathbf{Ж}^{HN}$ is evaluated on the testing set to report the miss rate. Note that as the rounds progress and more

Table 4.2: Miss rates of $\mathcal{Ж}^{HN}$ after bootstrapping with all frames, repeated three rounds.

Dataset	Round #	Number of frames			
		25	**50**	**75**	**100**
MIT	1	0.39	**0.43**	**0.45**	**0.47**
	2	**0.41**	-	-	-
	3	-	-	-	-
QMUL-J	1	0.83	0.85	**0.88**	**0.90**
	2	0.87	**0.89**	-	-
	3	**0.88**	-	-	-
PETS-02	1	**0.37**	**0.38**	**0.39**	**0.39**
	2	-	-	-	-
	3	-	-	-	-

Table 4.3: Miss rates of $\mathcal{Ж}^{HN}$ after bootstrapping with one frame at a time, repeated three rounds. Reported miss rates are after every round.

Dataset	Round #	Number of frames			
		25	**50**	**75**	**100**
MIT	1	0.23	0.26	0.26	0.28
	2	0.27	0.31	0.30	0.31
	3	0.28	0.30	0.32	0.32
QMUL-J	1	0.66	0.72	0.73	0.79
	2	0.71	0.74	0.79	0.76
	3	0.75	0.78	0.77	0.79
PETS-02	1	0.23	0.29	0.35	0.35
	2	0.26	0.29	0.33	0.34
	3	0.26	0.31	0.35	0.34

hard negatives are acquired, the miss rate will increase until the detector reaches convergence.

Table 4.2 shows the miss rates of the $\mathcal{Ж}^{HN}$ when bootstrapped with all frames at once. Bootstrapping is deemed to have converged (entry in bold) if no hard negatives are obtained in any particular round – in that case no more values are reported for that round. Three important trends can be observed. Firstly, easier datasets converge faster (compare the number of rounds needed by the three datasets for 25 frames) compared to harder ones. Secondly, utilizing a larger number of frames also results in faster convergence; however, the miss rates are also greater, which indicates greater suppression of the detection rates. Lastly, using this approach guarantees convergence.

Table 4.3 shows the miss rates of the $\mathcal{Ж}^{HN}$ when bootstrapped one frame at a time. For any given number of frames (25, 50, 75 or 100), the detector is bootstrapped on one frame at time, but the miss rate is only evaluated at the end of each round when all the frames have been processed. Two

Table 4.4: Different number of frames spit into batches of segment size 5. Miss rates of $\mathcal{K}^{HN}$ are reported after bootstrapping with each segment, repeated three rounds.

Dataset	Round #	Number of frames at fixed segment size														
25 Frames, segment size = 5																
MIT	1	0.15			0.20			0.24			0.25			0.21		
	2	0.23			0.25			0.26			0.28			0.29		
	3	0.25			0.25			0.30			0.29			0.29		
QMUL-J	1	0.21			0.51			0.63			0.69			0.73		
	2	0.69			0.65			0.69			0.75			0.75		
	3	0.72			0.74			0.76			0.75			0.77		
PETS-02	1	0.19			0.25			0.28			0.31			0.30		
	2	0.31			0.30			0.33			**0.35**			**0.35**		
	3	**0.35**			-			-			-			-		
50 Frames, segment size = 5																
MIT	1	0.15	0.20	0.24	0.25	0.21	0.26	0.29	0.29	0.28	0.32					
	2	0.30	0.31	0.33	0.32	0.32	0.33	0.32	0.33	0.31	0.33					
	3	0.32	**0.33**	**0.33**	**0.33**	-	-	-	-	-	-					
QMUL-J	1	0.21	0.51	0.63	0.69	0.73	0.73	0.76	0.74	0.78	0.72					
	2	0.75	0.78	0.79	0.79	0.76	0.77	0.88	0.80	0.78	0.80					
	3	0.79	0.80	0.80	**0.80**	**0.80**	**0.80**	-	-	-	-					
PETS-02	1	0.19	0.25	0.28	0.31	0.30	0.32	**0.35**	**0.35**	**0.35**	-					
	2	-	-	-	-	-	-	-	-	-	-					
	3	-	-	-	-	-	-	-	-	-	-					
75 Frames, segment size = 5																
MIT	1	0.15	0.20	0.24	0.25	0.21	0.26	0.29	0.29	0.28	0.32	0.34	0.35	0.34	0.36	0.36
	2	0.37	0.35	0.37	0.37	0.34	0.36	**0.37**	**0.37**	**0.37**	-	-	-	-	-	-
	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
QMUL-J	1	0.21	0.51	0.63	0.69	0.73	0.73	0.76	0.74	0.78	0.72	0.79	0.81	0.83	0.83	0.81
	2	0.83	0.83	0.81	0.82	0.83	0.81	0.82	0.82	0.81	0.82	**0.83**	**0.83**	**0.83**	-	-
	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
PETS-02	1	0.19	0.25	0.28	0.31	0.30	0.32	**0.35**	**0.35**	**0.35**	-	-	-	-	-	-
	2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	3	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

crucial differences emerge compared to the results from Table 4.2. Firstly, convergence is not reached even after three rounds; rather the miss rates actually appear to fluctuate in successive rounds. Secondly, for every combination of Dataset-Number of frames, the miss rate at the end of three rounds in Table 4.3 is significantly lower than the corresponding miss rate achieved after convergence in Table 4.2. Overall, though this approach does not guarantee convergence, the decline in detection rate is much lower compared to the previous approach.

Table 4.4 shows the miss rates when the total frames are split into batches of 5 frames, and the detector is bootstrapped on each batch or segment. For better confidence, bootstrapping is deemed to have converged if no hard negatives are acquired for three consecutive segments (entry in bold). If

Table 4.5: 50 frames spit into batches of different segment sizes. Miss rates of $\mathcal{H}^{HN}$ are reported after bootstrapping with each segment, repeated three rounds

Dataset	Round #	Number of segments									
Segment size = 5											
MIT	1	0.15	0.20	0.24	0.25	0.21	0.26	0.29	0.29	0.28	0.32
	2	0.30	0.31	0.33	0.32	0.32	0.33	0.32	0.33	0.31	0.33
	3	0.32	**0.33**	**0.33**	**0.33**	-	-	-	-	-	-
QMUL-J	1	0.21	0.51	0.63	0.69	0.73	0.73	0.76	0.74	0.78	0.72
	2	0.75	0.78	0.79	0.79	0.76	0.77	0.88	0.80	0.78	0.80
	3	0.79	0.80	0.79	**0.80**	**0.80**	**0.80**	-	-	-	-
PETS-02	1	0.19	0.25	0.28	0.31	0.30	0.32	**0.35**	**0.35**	**0.35**	-
	2	-	-	-	-	-	-	-	-	-	-
	3	-	-	-	-	-	-	-	-	-	-
Segment size = 10											
MIT	1	0.19		0.25		0.29		0.31		0.31	
	2	0.30		0.32		0.32		**0.33**		**0.33**	
	3	**0.33**		-		-		-		-	
QMUL-J	1	0.28		0.64		0.73		0.78		0.77	
	2	0.77		0.78		0.78		0.80		**0.81**	
	3	**0.81**		**0.81**		-		-		-	
PETS-02	1	0.31		**0.35**		**0.35**		**0.35**		-	
	2	-		-		-		-		-	
	3	-		-		-		-		-	
Segment size = 15											
MIT	1	0.23		0.3		0.32		0.32			
	2	**0.35**		**0.35**		**0.35**		-			
	3	-		-		-		-			
QMUL-J	1	0.34		0.71		0.76		0.79			
	2	0.80		0.80		0.81		**0.82**			
	3	**0.82**		**0.82**		-		-			
PETS-02	1	**0.35**		**0.35**		**0.35**		-			
	2	-		-		-		-			
	3	-		-		-		-			

bootstrapping is merely performed on newer segments of frames, the continuous exposure to newer hard negatives may prevent convergence. That is why the concept of repeating bootstrapping on the same frames for a pre-determined round is necessary - it allows the detector to reach convergence and complete the bootstrapping procedure rather than continuing indefinitely. The results from Table 4.4 suggests that 25 frames may be too less to cause convergence for harder datasets in 3 rounds. However, for 50 frames or more, convergence is guaranteed.

The most significant finding was that this approach does not only guarantee convergence, but suppresses the detection rate lesser, particularly for harder datasets. Compared to the figures for 50 frames in Table 4.2, the miss rates for MIT, QMUL-J and PETS-02 using 50 frames with segment size

5 is 10%, 9% and 3% lower, respectively in Table 4.4. 50 frames was found to be optimal; using 75 frames did achieve convergence earlier, but also caused the miss rate to rise a bit further.

Despite achieving convergence, considerable fluctuations can be seen in the progression of the miss rates (see miss rates for 50 frames in Table 4.4). Therefore, additional segment sizes of 10 and 15 were tested, while keeping the total number of frames at 50. The miss rates are reported in Table 4.5. While segment size 15 also guaranteed more stable convergence without fluctuations, segment size 10 was selected because it achieved the best balance between stable convergence and minimum miss rates.

Accordingly, for Step 2.1 outlined in Algorithm 1, the finalized parameter settings for bootstrapping are : **number of frames k = 51, number of segments s = 5, segment size $fr = (k-1)/s$ = 10, number of rounds r = 3.**

4.4 Evaluation criteria

Assuming pedestrians as positives denoted by P and non-pedestrians as negatives denoted by N, true positives are denoted by TP, true negatives by TN, false positives by FP and false negatives by FN. For easier comprehension during comparative discussions, the standard classification terminologies of accuracy, recall, precision, specificity and negative predictive value of the oracles are reworded as $\mathcal{X}$Acc, $\mathcal{X}$Rec+, $\mathcal{X}$Pre+, $\mathcal{X}$Rec- and $\mathcal{X}$Pre-, respectively, and calculated as:

$$\mathcal{X}\text{Acc} = (TP+TN)/(TP+TN+FP+FN) \tag{4.1}$$

$$\mathcal{X}\text{Rec+} = TP/(TP+FN) \tag{4.2}$$

$$\mathcal{X}\text{Pre+} = TP/(TP+FP) \tag{4.3}$$

$$\mathcal{X}\text{Rec-} = TN/(TN+FP) \tag{4.4}$$

$$\mathcal{X}\text{Pre-} = TN/(TN+FN) \tag{4.5}$$

For individual TSF performance metrics, the correctly rejected samples are denoted by TN^{TSF} and incorrectly rejected samples by FN^{TSF}, and the following are calculated:

$$\text{TSF-Pre} = TN^{TSF}/(\ TN^{TSF} + FN^{TSF}) \tag{4.6}$$

$$\text{TSF-Con} = TN^{TSF}/(\ TN+FP) \tag{4.7}$$

TSF-Pre indicates the rejection precision of the TSF and TSF-Con measures the contribution of the TSF in terms of the percentage of total negatives it is able to reject.

For detector evaluation, the PASCAL measure is employed, which stipulates that a detection response is a true positive only if the area of intersection between a detection bounding box and a ground truth bounding box is at least 50% of their union. As most state-of-the-art methods report their detection performances using receiver operating characteristic (ROC) curves of recall rates against false positive per image (FPPI), the same is utilized when benchmarking. However, for detailed evaluation of the VAT detectors, detection error trade-off (DET) curves of miss rate against FPPI using log-log plots similar to [34] are generated, as they are more linear compared to ROC curves.

4.5 Performance evaluation of oracles

4.5.1 Overall oracle performance

Figure 4.3 displays the changes (on the primary axis) in the number of pedestrians (+ves) and non-pedestrians (-ves) for all datasets, as they pass through the TSFs of the oracles, $\mathcal{O}^1$ and $\mathcal{O}^2$. For each dataset, the overall oracle performance metrics (according to Section 4.4) are plotted along the secondary axis. Regardless of the detector being used, the input samples to $\mathcal{O}^1$ are always the same for a particular dataset because they are obtained by motion-based sample acquisition during Inception. In contrast, the input samples to $\mathcal{O}^2$ for a particular dataset are detection responses acquired by applying the detector $\mathcal{H}^{RP}$ during Finalization, and are different for each implemented detection algorithm. As three different detection algorithms are tested (refer to Section 4.2), there are three corresponding separate plots for $\mathcal{O}^2$ (Figure 4.3b – 4.3d), while only one for $\mathcal{O}^1$ (Figure 4.3a). Additionally, while $\mathcal{O}^2$ labels both passed and rejected samples as pedestrians and non-pedestrians respectively, $\mathcal{O}^1$ labels only the passed samples as pedestrians, while ignoring the rejects (which are re-visited during bootstrapping). Therefore, only

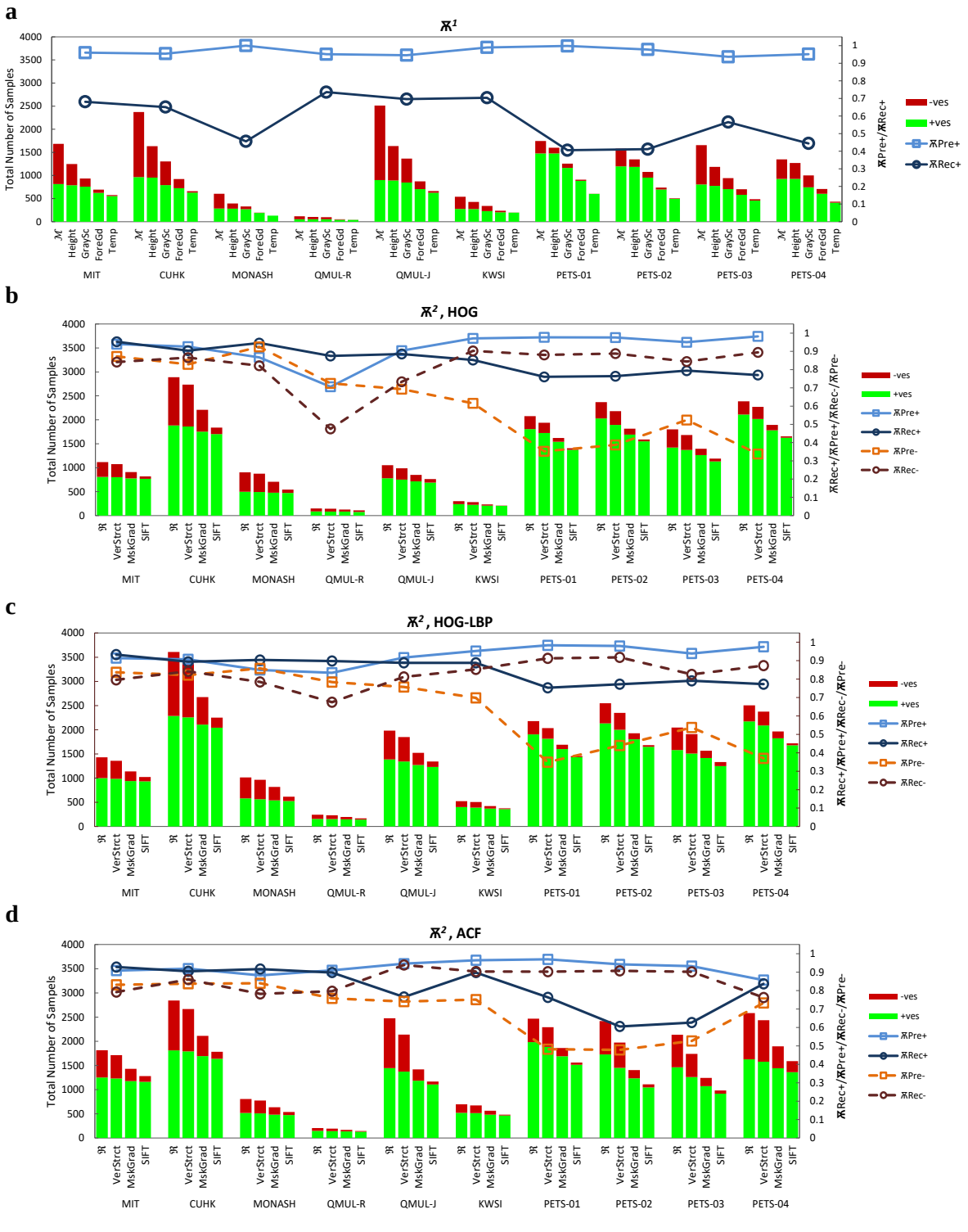


Figure 4.3: Performance of oracles on all datasets. For every dataset, each stacked bar (except the 1st one) indicates the total number of remaining samples after passing the respective TSF, as a combination of pedestrians instances or +ves (green portion of stacked bar) and non-pedestrian instances or -ves (red portion of stacked bar). For $\mathcal{K}^1$, the 1st bar indicates the number of motion regions ($\mathcal{M}$) acquired during Inception. For $\mathcal{K}^2$, the 1st bar indicates the number of detection responses ($\mathfrak{R}$) from the detector $\mathcal{K}^{RP}$ during Finalization. For all datasets, note that last bar also indicates the number of samples passed and labelled as pedestrians by the oracle.

Table 4.6. Complete Oracle-1 statistics.

		MIT	CUHK	MONASH	QMUL-R	QMUL-J	KWSI	PETS-01	PETS-02	PETS-03	PETS-04	Mean
P / N		812/871	965/1409	285/317	53/63	899/1615	277/261	1476/268	1197/383	806/846	925/424	
$TN^{TSF}+FN^{TSF}$ / TSF-Pre / TSF-Con	Height	436/.94/0.47	739/0.99/0.52	210/0.98/0.65	14/0.93/0.21	877/0.99/0.54	110/0.96/0.41	145/1/0.54	232/0.95/0.57	469/0.93/0.51	78/1/0.18	0.97/0.46
	GrayScale	314/0.9/0.32	333/0.51/0.12	62/0.84/0.16	6/1/0.1	273/0.84/0.14	89/0.48/0.16	345/0.09/0.12	272/0.15/0.1	245/0.72/0.21	270/0.33/0.21	0.59/0.16
	ForeGd	241/0.44/0.12	380/0.82/0.22	136/0.4/0.17	43/0.77/0.52	491/0.71/0.22	101/0.76/0.3	347/0.19/0.25	339/0.24/0.21	240/0.48/0.14	295/0.52/0.36	0.53/0.25
	Temp	116/041/0.05	264/0.64/0.12	64/0.06/0.01	12/0.75/0.14	211/0.63/0.08	41/0.8/0.13	308/0.08/0.09	233/0.13/0.08	214/0.42/0.11	274/0.3/0.2	0.42/0.1
TP+FP/X̄Pre+/X̄Rec+		576/0.96/0.68	658/0.95/0.65	130/1/0.46	41/0.95/0.74	662/0.95/0.7	197/0.99/0.7	601/0.99/0.41	504/0.98/0.42	487/0.94/0.57	432/0.95/0.44	0.97/0.58

Table 4.7. Complete Oracle-2 Statistics.

			MIT	CUHK	MONASH	QMUL-R	QMUL-J	KWSI	PETS-01	PETS-02	PETS-03	PETS-04	Mean
HOG	P / N		807/309	1883/1008	498/404	88/61	779/274	238/62	1807/268	2029/340	1422/380	2110/276	
	$TN^{TSF}+FN^{TSF}$ / TSF-Pre / TSF-Con	VerStrct	42/0.88/0.12	155/0.85/0.13	27/0.74/0.05	7/0.57/0.07	66/0.58/0.14	19/0.32/0.1	136/0.4/0.2	189/0.28/0.16	119/0.56/0.18	116/0.2/0.08	0.54/0.12
		MskGrad	165/0.84/0.45	527/0.8/0.42	168/0.92/0.38	15/0.93/0.23	137/0.72/0.36	49/0.65/0.52	318/0.42/0.5	366/0.45/0.49	289/0.63/0.48	376/0.37/0.5	0.67/0.43
		SIFT	91/0.93/0.28	371/0.86/0.32	164/0.96/0.39	18/0.61/0.18	87/0.74/0.23	23/0.78/0.29	215/0.21/0.17	225/0.37/0.25	205/0.36/0.19	239/0.36/0.31	0.62/0.26
	TP+FP/X̄Pre+/X̄Rec+		818/0.94/0.95	1838/0.93/0.9	543/0.87/0.95	109/0.71/0.88	763/0.9/0.89	209/0.97/0.85	1406/0.98/0.76	1589/0.98/0.76	1189/0.95/0.79	1655/0.98/0.77	0.92/0.85
	TN+FN/X̄Pre-/X̄Rec-		298/0.87/0.84	1053/0.83/0.87	359/0.92/0.82	40/0.73/0.48	290/0.69/0.73	91/0.62/0.9	669/0.35/0.88	780/0.39/0.89	613/0.52/0.84	731/0.34/0.89	0.63/0.81
	X̄Acc		0.922	0.891	0.89	0.711	0.846	0.863	0.776	0.782	0.805	0.785	0.827
HOG-LBP	P / N		1001/431	2285/1323	582/432	157/86	1387/596	403/122	1903/276	2133/415	1581/465	2172/332	
	$TN^{TSF}+FN^{TSF}$ / TSF-Pre / TSF-Con	VerStrct	73/0.79/0.13	228/0.88/0.15	47/0.64/0.07	10/0.6/0.07	134/0.68/0.15	20/0.4/0.07	146/0.39/0.21	198/0.33/0.16	137/0.47/0.14	128/0.34/0.13	0.55/0.13
		MskGrad	221/0.79/0.41	706/0.78/0.42	145/0.84/0.29	36/0.86/0.36	325/0.78/0.42	81/0.75/0.5	343/0.38/0.47	422/0.53/0.54	343/0.73/0.54	410/0.35/0.44	0.68/0.44
		SIFT	116/0.95/0.26	422/0.85/0.27	202/0.92/0.43	28/0.75/0.24	180/0.78/0.24	48/0.73/0.29	233/0.28/0.24	247/0.37/0.22	235/0.29/0.15	246/0.41/0.3	0.63/0.26
	TP+FP/X̄Pre+/X̄Rec+		1022/0.91/0.93	2252/0.91/0.89	619/0.85/0.9	169/0.83/0.90	1344/0.92/0.89	376/0.95/0.89	1457/0.98/0.75	1681/0.98/0.77	1331/0.94/0.79	1720/0.98/0.77	0.93/0.85
	TN+FN/X̄Pre-/X̄Rec-		410/0.84/0.8	1356/0.82/0.84	395/0.86/0.78	74/0.78/0.67	639/0.76/0.81	149/0.7/0.85	722/0.35/0.91	867/0.44/0.92	715/0.54/0.83	784/0.37/0.87	0.65/0.83
	X̄Acc		0.892	0.875	0.853	0.819	0.865	0.88	0.773	0.796	0.799	0.786	0.834
ACF	P / N		1252/562	1811/1033	519/289	147/59	1445/1031	520/177	1983/484	1728/690	1462/674	1627/955	
	$TN^{TSF}+FN^{TSF}$ / TSF-Pre / TSF-Con	VerStrct	100/0.79/0.14	177/0.89/0.15	32/0.72/0.08	13/0.69/0.15	339/0.79/0.26	24/0.79/0.1	175/0.43/0.16	444/0.37/0.24	397/0.49/0.29	147/0.67/0.1	0.66/0.17
		MskGrad	283/0.81/0.41	554/0.82/0.44	141/0.82/0.4	24/0.75/0.31	717/0.74/0.51	108/0.71/0.44	430/0.55/0.49	568/0.62/0.51	495/0.62/0.46	539/0.75/0.42	0.72/0.44
		SIFT	152/0.91/0.25	330/0.84/0.27	96/0.92/0.3	25/0.8/0.34	251/0.69/0.17	81/0.79/0.36	301/0.41/0.26	296/0.36/0.16	262/0.4/0.16	307/0.74/0.24	0.69/0.25
	TP+FP/X̄Pre+/X̄Rec+		1279/0.91/0.93	1789/0.92/0.91	539/0.88/0.92	145/0.91/0.90	1169/0.95/0.77	484/0.96/0.9	1561/0.97/0.76	1110/0.94/0.61	982/0.93/0.63	1589/0.86/0.84	0.92/0.81
	TN+FN/X̄Pre-/X̄Rec-		535/0.83/0.79	1061/0.84/0.86	269/0.84/0.78	47/0.76/0.8	1307/0.74/0.94	213/0.75/0.9	906/0.48/0.9	1308/0.48/0.91	1154/0.53/0.9	993/0.73/0.76	0.7/0.85
	X̄Acc		0.886	0.889	0.869	0.869	0.838	0.899	0.791	0.692	0.714	0.809	0.825

Figure 4.4: Examples of the types of non-pedestrian instances rejected by the TSFs of $\bar{X}^1$. Each column illustrates the instances rejected in 10 datasets by a single pruner of a TSF. TSFs with multiple pruners are assigned dedicated columns for each pruner. Only 10 examples are displayed in each montage – hence black spaces indicate there were less than 10 instances rejected by that pruner. Zoom in for better clarity.

Figure 4.5: For each dataset, a set of 100 instances from the samples passed by $\mathcal{X}^1$ and labelled as pedestrian instances. Black spaces indicate less than 100 instances were passed. Zoom in for better clarity.

$\mathcal{X}$Rec+ and $\mathcal{X}$Pre+ are evaluated for $\mathcal{X}^1$ (see Figure 4.3a). Complete numerical performance statistics for the oracles and their individual TSFs are reported in Table 4.6 and Table 4.7.

For every dataset in each plot (10 datasets×4 plots = 40 overall oracle results) in Figure 4.3, it can be clearly observed that starting from the input sample set (first stacked bar), the number of non-pedestrian instances or -ves steadily declines as they pass each subsequent TSF, until the percentage of -ves is very small in the samples passed by the oracle (last stacked bar) as pedestrians. This ubiquitous pattern suggests that the designed oracles are highly consistent in effectively segregating pedestrians from sample sets comprising pedestrian and non-pedestrian instances, in a wide variety of environments.

To reinforce the above observations, $\mathcal{X}^1$ is considered as an example and detailed illustrations of the samples that are rejected and passed by $\mathcal{X}^1$ are depicted in Figure 4.4 and Figure 4.5, respectively. In order to extensively judge the consistency and effectiveness with which $\mathcal{X}^1$ segregates pedestrians from sample sets consisting of pedestrian and non-pedestrians instances, in different surveillance environments, Figure 4.3a, 4.4 and 4.5, and Table 4.6 can be examined as such. For each dataset

1) Refer to Figure 4.3a to visualize how the non-pedestrians are progressively rejected as the samples pass through each TSF. The overall recall and precision can be compared to other datasets.
2) For the exact number of samples rejected by each TSF, the precision of rejection and the percentage of total non-pedestrians removed by that TSF, refer to Table 4.6.
3) To view the type of samples rejected by each TSF, refer to Figure 4.4.
4) To view the type of samples that ultimately pass all the TSFs, and are therefore labelled by $\mathcal{X}^1$ as pedestrian instances, refer to Figure 4.5. Take particular note of the high percentage of the correctly labelled pedestrian instances.

For $\mathcal{X}^1$, $\mathcal{X}$Pre+ is typically above 95%, with an average of 97% (See Table 4.6 and view Figure 4.5). This high precision of $\mathcal{X}^1$ is unaffected by the size of the input sample set ($\mathcal{M}$) or the dataset (see Table 4.6). As $\mathcal{X}^1$ is designed for very high-precision labelling of pedestrians, the recall, $\mathcal{X}$Rec+ is relatively lower, with an average of 58%. $\mathcal{X}$Rec+ of $\mathcal{X}^1$ is usually worse for datasets with a lower percentage of -ves in the input sample set, which is common amongst datasets without non-pedestrian

moving objects, such as PETS-01 – PETS-04 (see Figure 4.3a). This is because the presence of fewer –ves means the outlier ranges in the distribution of the computed CBA scores (see Figure 3.3) would actually encompass some +ves; inevitably, these +ves are rejected when the rejection criteria are applied by the pruners of each TSF (see Figure 4.4 for visualization). To view more samples labelled by $\mathcal{X}^1$ as pedestrians, refer to Appendix A.

Compared to $\mathcal{M}$, the input sample set $\mathfrak{R}$ to $\mathcal{X}^2$ (Figure 4.3b - 4.3d) usually consists of more +ves and fewer –ves, as they are acquired using the detector $\mathcal{H}^{RP}$. $\mathcal{X}$Pre+ of $\mathcal{X}^2$ is still very high, but generally lower than $\mathcal{X}^1$, with an average of 92-93% for every detector (see Table 4.7). Note that $\mathcal{X}^2$ is designed to have lower $\mathcal{X}$Pre+; hence, it comprises of fewer TSFs compared to $\mathcal{X}^1$ even though $\mathfrak{R}$ includes harder negatives than $\mathcal{M}$. This is necessary because unlike $\mathcal{X}^1$ that ignores the rejects, $\mathcal{X}^2$ must label the rejects as non-pedestrians, and a high-precision setting would mean that too many pedestrians would be incorrectly rejected causing the accuracy, $\mathcal{X}$Acc, to drop. There are two important correlations that exist between the four metrics for $\mathcal{X}^2$. It is important to understand them because unlike $\mathcal{X}^1$ whose performance solely depends on $\mathcal{X}$Pre+ (because it ignores the rejects and focuses only on high-precision labelling of pedestrian instances), the performance of $\mathcal{X}^2$ is measured by the accuracy $\mathcal{X}$Acc, which can be impacted by each of these metrics.

- A higher $\mathcal{X}$Pre+ means fewer non-pedestrians or –ves were labelled as pedestrians. This directly means more –ves were correctly labelled as non-pedestrians, resulting in a higher $\mathcal{X}$Rec-.
- Correspondingly, lower $\mathcal{X}$Rec+ means more pedestrians or +ves were incorrectly rejected. This directly means more +ves were incorrectly labelled as non-pedestrians, resulting in a lower $\mathcal{X}$Pre-.

Both these correlations can be verified from Figure 4.3b, 4.3c and 4.3d, particularly for the PETS datasets, where $\mathcal{X}$Pre- drops as a consequence of lower $\mathcal{X}$Rec+. The trends of all four performance metrics are consistent across all three plots of $\mathcal{X}^2$, indicating that the oracle performance is not influenced by detector choice. However, oracle performance can fluctuate when too few samples are available, as is the case for HOG on QMUL-R (see Figure 4.3b).

Overall, taking the above correlations into account, since $\bar{X}$Pre+ is always high, the accuracy of X^2 is dependent on the labelling precision $\bar{X}$Pre-, of non-pedestrians. For MIT, CUHK and MONASH, $\bar{X}$Pre- and $\bar{X}$Acc are in the range of 82-92% and 85-92%, respectively. For the extremely challenging datasets QMUL-R, QMUL-J and KWSI, the $\bar{X}$Pre- and $\bar{X}$Acc have lower scores in the range of 62-78% and 82-90%, respectively. However, for the 4 PETS datasets, poorer scores of $\bar{X}$Rec+ causes $\bar{X}$Pre- and $\bar{X}$Acc to drop considerably to 34-54% and 69-81%, respectively. For exact scores, refer to Table 4.7. See Appendix B for more illustration of samples labelled as pedestrians and non-pedestrians by X^2.

4.5.2 Individual TSF performances

To facilitate intra-TSF and inter-TSF performance comparisons across the 10 datasets, for each oracle, the TSF-Pre against TSF-Con of all its TSFs on every dataset is plotted, as shown in Figure 4.6. As before, there is a single plot for X^1 (Figure 4.6a) and three separate plots for X^2 (Figure 4.6b – 4.6d). Various implications of the subsequent observations are discussed in subsection 4.8.3. Overall, for both oracles, it can be observed that the TSF-Con scores of each individual TSF across the 10 datasets are clustered along the x-axis, such that the TSFs can be ranked in terms of the range of their TSF-Con scores (refer to the plotted TSF means for quick ranking, identified by the filled markers). As TSF-Con denotes the impact of a TSF by measuring the percentage of the total -ves the TSF is able to reject, the presence of these clusters (see Figure 4.6a – 4.6d) and furthermore, the homogeneity of the clustering despite applying X^2 to different $\Re$ acquired by different detectors (compare Figure 4.6b, 4.6c and 4.6d) substantiates the stability of the configured TSF sequence within each oracle. Incorrect choice or sequence of TSFs is likely to have caused the TSFs to function inconsistently for different $\Re$ acquired by different detectors, resulting in a more haphazard pattern in the plots rather than exhibiting such clusters.

Except for the Height filter of X^1, TSF-Pre scores may appear dispersed along the y-axis without any noticeable pattern (see Figure 4.6a – 4.6d). Nonetheless, TSF-Pre of any given TSF actually depends on the ratio of the number of -ves that can be rejected by that TSF to the total number of remaining

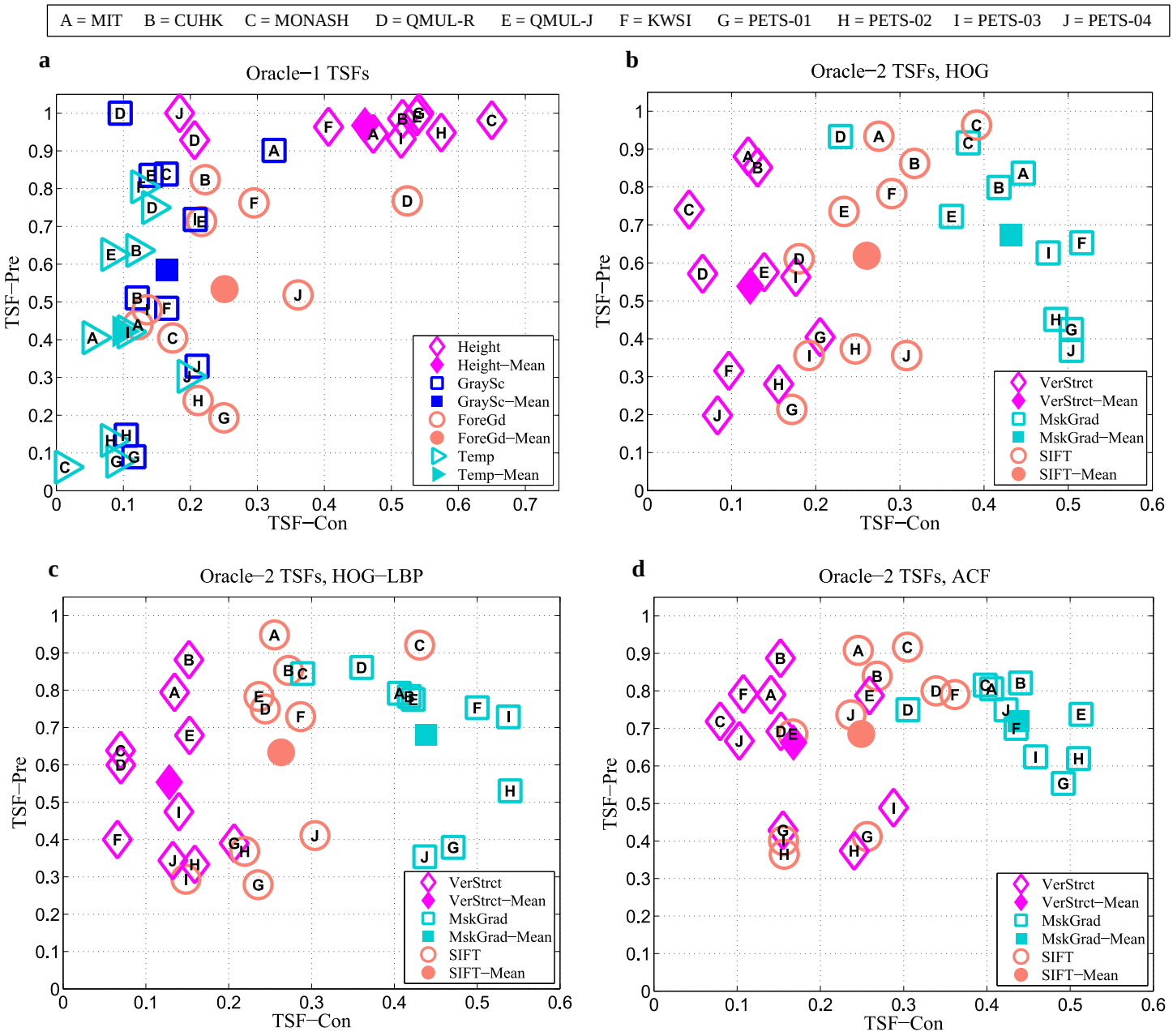


Figure 4.6: Individual TSF Performances on all datasets. Filled markers represent the average performance of each TSF.

samples. For example, as GrayScale (blue squares in Figure 4.6a) evaluates grayscale variance, its rejection precision is much higher (TSF-Pre > 70% in Figure 4.6a) for datasets with poorer brightness levels due to the presence of more –ves with insufficient grayscale variance. However, for datasets with better brightness, such as CUHK (B) or KWSI (F) in Figure 4.6a, there are fewer -ves that can be rejected by GrayScale, resulting in much lower TSF-Pre (compare blue squares with letters B and F with others).

Additionally, TSF-Pre scores will most likely be below average when the percentage of total remaining -ves has already dropped significantly, because this automatically implies the aforementioned condition of fewer -ves being available for rejection by subsequent TSFs. This can be confirmed by carefully cross-checking the number of -ves of the preceding stacked bar in Figure 4.3 for TSFs whose TSF-Pre score is below 50% in Figure 4.6.

According to Figure 4.6a, Height is, by far, the most effective TSF in $\mathcal{X}^1$ with highly precise and consistent rejection ability, as demonstrated by its TSF-Pre > 90% on every dataset (see Table 4.6 for specific scores). The precision of the other three TSFs are dependent on the conditions discussed in the previous paragraph. By examining the TSF-Con scores, it can be deduced that usually, Height also rejects the most -ves, ForeGd rejects slightly more -ves than GraySc, and Temp rejects the least. It is reiterated here that the lowest TSF-Con scores of the Temp TSF does not indicate poor performance; but rather the presence of fewer non-pedestrian instances that can be rejected by Temp. Collective analysis of all TSFs on each dataset in Figure 4.6a shows that following:

- The top TSF precisions are achieved for two of the extremely hard datasets, QMUL-R (D) and QMUL-J (E), indicated by TSF-Pre > 60% for every TSF.
- The hard dataset CUHK (B) and the other extremely hard dataset KWSI (F) follow closely behind at second, indicated by TSF-Pre > 50% for every TSF.
- The worst TSF precisions are for PETS-01 (G) and PETS-02 (H), indicated by TSF-Pre < 30% for 3 out of the 4 TSFs (see Figure 4.6a).

Compared to $\mathcal{X}^1$, it can be uniformly observed in Figure 4.6b, 4.6c and 4.6d that for $\mathcal{X}^2$, the TSF-Pre scores are less dispersed along the y-axis. With respect to the range of the TSF-Con scores of each TSF across the 10 datasets, there is clearer separation between the TSFs along the x-axis. As evident from the TSF-Con scores in Figure 4.6b – 4.6d, amongst the three TSFs, usually, MskGrad rejects the most -ves and VerStrct rejects the least. Overall, regardless of the chosen detector, the following trends are observed:

- The top TSF-Pre scores are achieved by the hard datasets MIT (A), CUHK (B) and MONASH (C), usually > 70% for every TSF.

- The TSF-Pre scores of the extremely hard datasets QMUL-R (D), QMUL-J (E) and KWSI (F) are at a close second, usually > 60% for every TSF.
- The worst TSF-Pre scores belong to the PETS datasets (G-H), usually < 50% for most TSFs. Notice how the TSF-Pre scores of all the PETS datasets increases for ACF relative to HOG-LBP and HOG, as shown in Figure 4.6d due to the presence of more -ves (see stacked bars in Figure 4.3d and compare against Figure 4.3b and 4.3c).

4.6 Performance evaluation of VAT detectors

For a comprehensive evaluation of the VAT framework, the experimental results of the detectors listed in Table 4.8 are compared. The detectors from every stage of VAT were tested to assess how the performance improves as VAT progresses. SS, VAT+Generic and VAT+Generic-NS were trained to study how the dependence on a generic detector influences scene-specific training in target environments of varying difficulty, in comparison to VAT. For each of the 10 datasets, the detectors listed in Table 4.8 were tested for each of the three pedestrian detection algorithms – HOG, HOG-LBP and ACF. However, Manual-Initial and Manual-Final were tested only on MIT, CUHK and MONASH, as the training sets of the remaining datasets could not be annotated due to time constraints. Wherever available, detection results of the manually trained detectors serve as the upper bound in each dataset and represent the target performance that the proposed VAT strives to reach. DET curves for the hard datasets (MIT, CUHK & MONASH), the extremely hard datasets (QMUL-R, QMUL-J & KWSI) and the medium datasets (PETS-01 – PETS-04) are shown in Figure 4.7, Figure 4.8 and Figure 4.9, respectively, and some qualitative detection results are displayed in Figure 4.14, Figure 4.15 and Figure 4.16, respectively. Between Figure 4.7, 4.8 and 4.9, the DET curves of over 250 evaluated detectors is presented. For any subsequent numerical performance comparisons, the reported miss rate is at FPPI=1. We present a detailed analysis of the results below.

Table 4.8: List of detectors tested for performance evaluation of VAT

Generic	Trained using the labelled source dataset, INRIA.
SS	Basic semi-supervised approach that uses INRIA dataset, and iteratively retrains a scene-specific detector with augmented confident positives and negatives obtained by applying the Generic detector to the target scene
VAT+Generic	Trained by executing VAT, but commences operation by applying Generic detector to the target scene to acquire potential training samples instead of relying solely on motion detection. Uses INRIA dataset + target training samples acquired by VAT
VAT+Generic-NS	Similar to VAT+Generic, but NO SOURCE training samples are utilized. Only the target training samples acquired by VAT are used
Manual-Initial	Trained using manually labelled pedestrians from the target scene as positives + negatives sampled from $\mathbb{E}_1^{-ve}$ as per Step 1.4 of Algorithm 1.
Manual-Final	Trained by bootstrapping Manual-Initial with hard negatives, according to Step 2.1 of Algorithm 1
VAT-Initial	$\mathbf{Ж}^{Initial}$, generated by the Inception stage of VAT , with initial training samples
VAT-HN	$\mathbf{Ж}^{HN}$, generated by the Bootstrapping stage of VAT, with augmented hard negatives
VAT-RP	$\mathbf{Ж}^{RP}$, generated by the Bootstrapping stage of VAT, with augmented rejected positives
VAT-Final	$\mathbf{Ж}^{Final}$, generated by the Finalization stage of VAT, with augmented final training samples. $\mathbf{Ж}^{Final}$ is the ultimate output of VAT, and represents the scene-specific pedestrian detector.

4.6.1 VAT progression

Overall, the following observations can be made in almost every plot of Figure 4.7 and Figure 4.8 (see subsection 4.8.5 for explanation of observations) – A) VAT-HN achieves a significant drop in miss rate compared to VAT-Initial, B) the performance improvement attained by VAT-RP over VAT-HN is marginal and C) VAT-Final achieves a further, marked enhancement in performance but the achieved drop in miss rate relative to VAT-RP is smaller than that between VAT-HN and VAT-Initial. However, except a few plots (Figure 4.9g, 4.9j, 4.9h, 4.9c and 4.9l), the DET curves in Figure 4.9 suggest that full-blown VAT is generally not necessary to improve detector performance on easier datasets like PETS and dependence on a generic detector may suffice (refer to subsequent subsection 4.6.3 on Comparisons with SS, VAT+Generic and VAT+Generic-NS for more details). The performance improvement of VAT-Final over VAT-Initial is negligible for some PETS datasets (see Figure 4.9a, 4.9b and 4.9k).

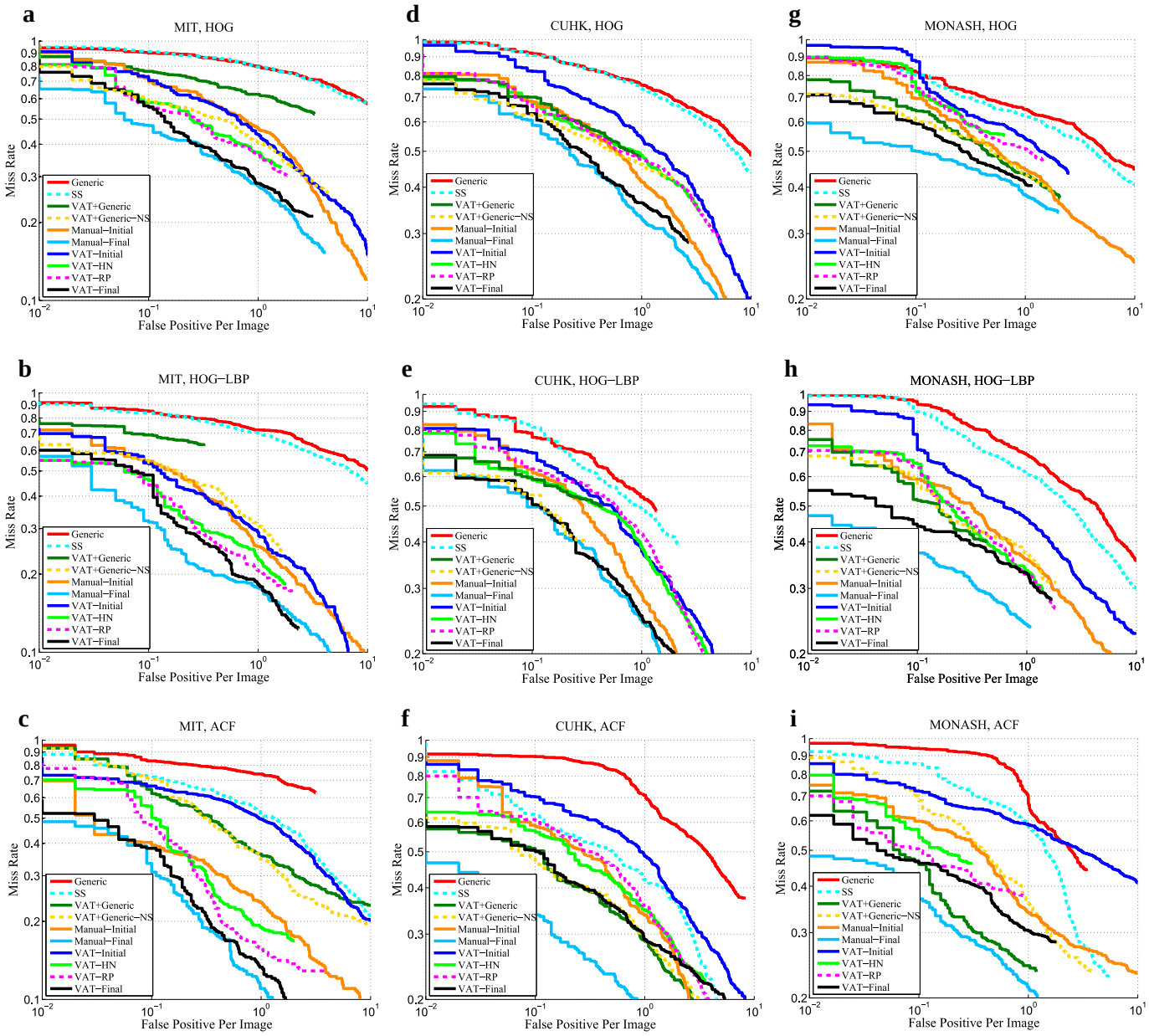


Figure 4.7: VAT performance evaluation results for MIT, CUHK and MONASH

4.6.2 Comparison with generic detectors

Generally, scene specific detectors trained by VAT (VAT-Final) outperform offline trained generic detectors (Generic) by huge margins. The largest performance gaps, of upto 60% (see Figure 4.7c) are achieved on the hard datasets. The performance differences in Figure 4.8 are still substantial, but due to the sheer difficulty of these datasets, the achievable margins are lower, with the highest being approximately 40% (see Figure 4.8c). Amongst the PETS datasets, large performance gaps are still

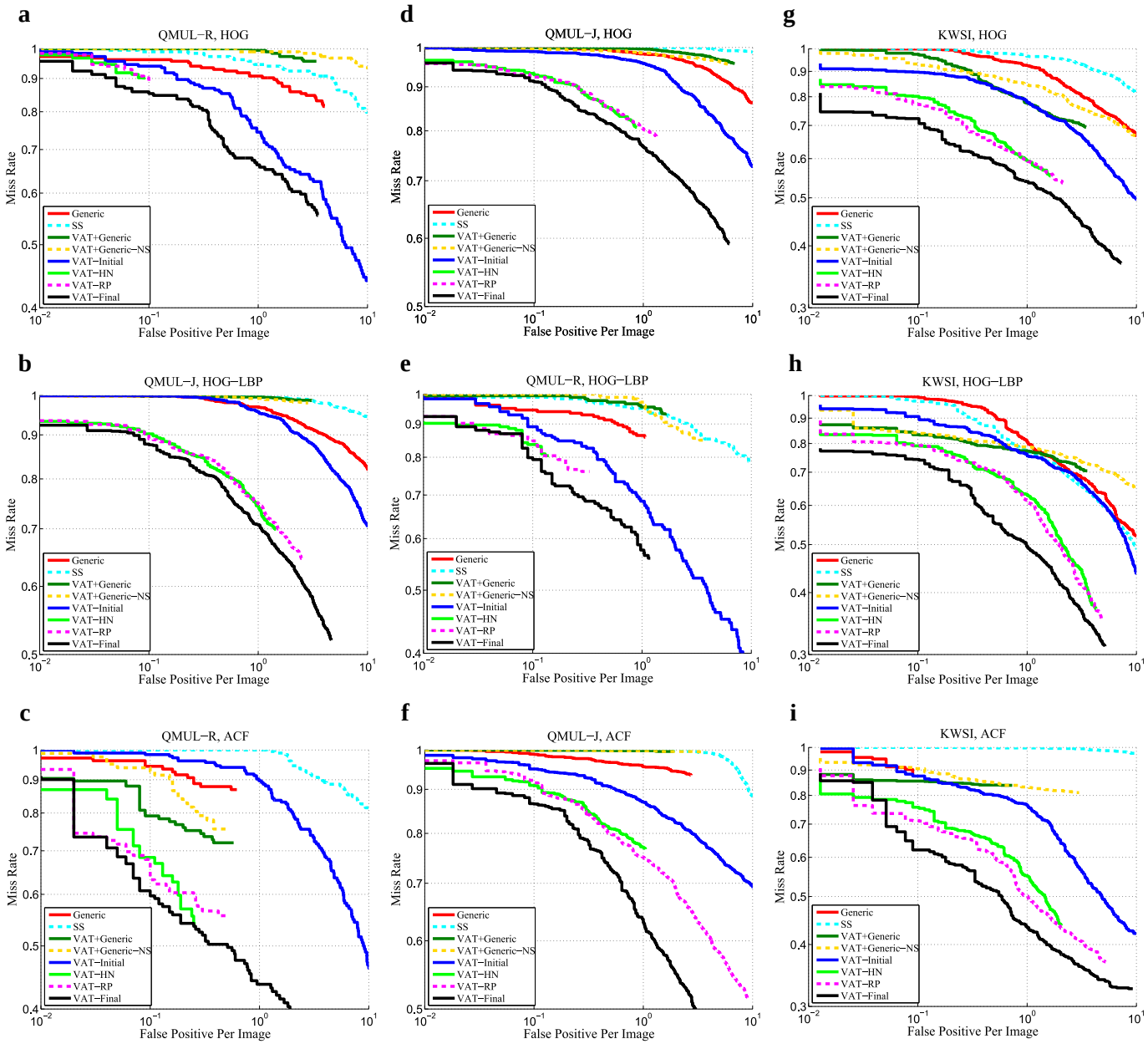


Figure 4.8: VAT performance evaluation results for QMUL-R, QMUL-J and KWSI

achieved for HOG and HOG-LBP based detectors but the margins between generic detectors based on ACF and their VAT counterparts are far smaller, with some of them less than 5% (see Figure 4.9c, 4.9i and 4.9l.)

4.6.3 Comparison with methods that depend on pre-trained generic detector

Hard Datasets. SS achieves minimal improvement over Generic, unless the detection algorithm is

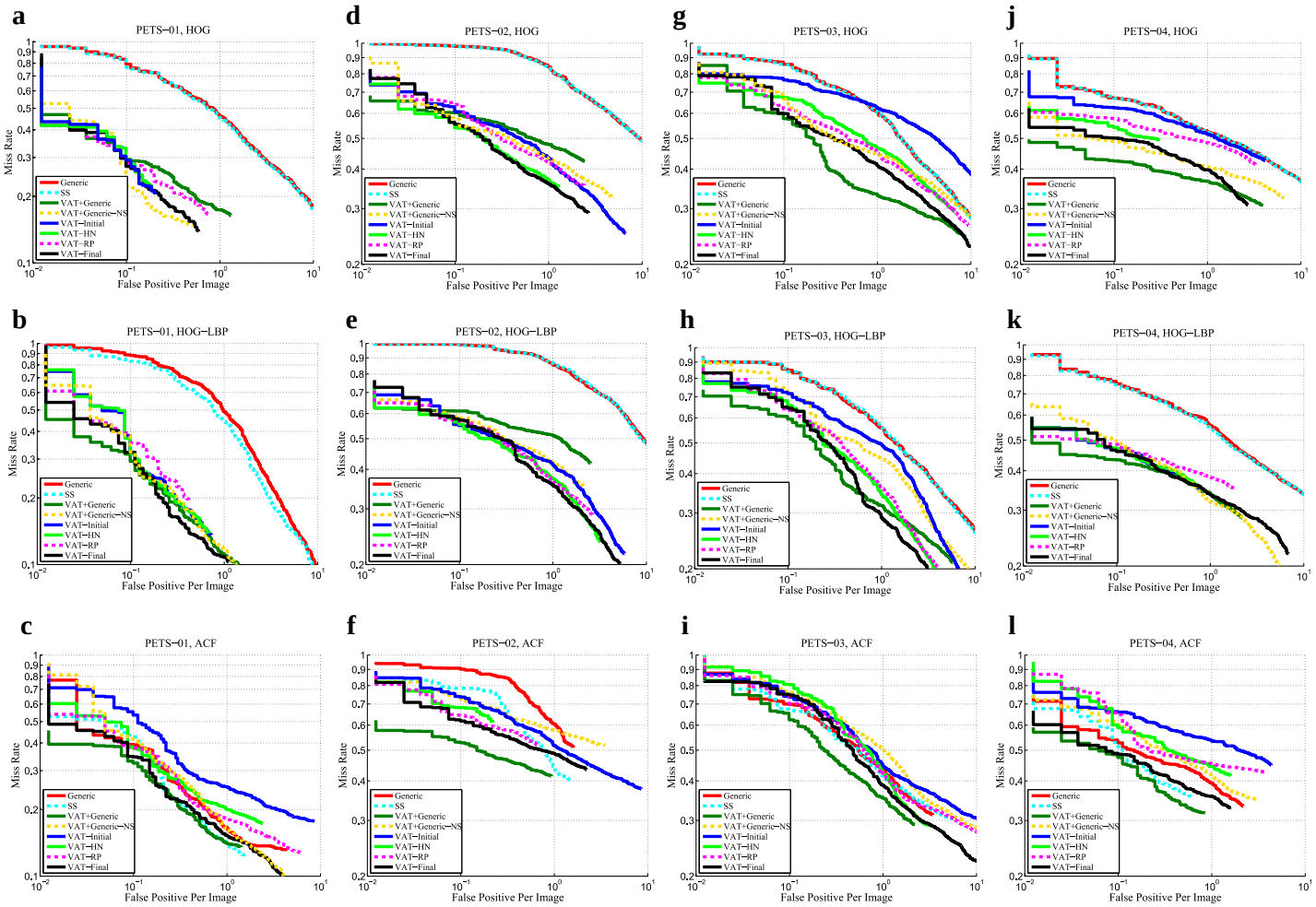


Figure 4.9: VAT performance evaluation results for PETS-01, PETS-02, PETS-03 and PETS-04

ACF. (see bottom row of Figure 4.7). A lower miss rate of VAT+Generic-NS (source samples not used in training) compared to VAT+Generic (INRIA source samples included in training) indicates that there is dataset shift due to the inclusion of INRIA source samples (see Figure 4.7a, 4.7b). If the opposite is true and VAT+Generic is lower, it can be interpreted that the utilization of source samples is actually beneficial to the detector training (see Figure 4.7i.). On MIT, VAT-Final consistently achieves a miss rate of at least 10% lower than these three detectors (see Figure 4.7a-4.7c), On CUHK, the performance of VAT-Final is either better (see Figure 4.7d, 4.7e) or equal to VAT+Generic and VAT+Generic-NS (see Figure 4.7f). As for MONASH, the miss rate of VAT-Final is lower than VAT+Generic and VAT+Generic-NS for HOG and HOG-LBP, but VAT+Generic achieves 7% lower miss rate than VAT-Final when using ACF.

Extremely hard datasets. Except Figure 4.8h, it can be seen that the performance of SS actually degrades relative to Generic. On QMUL-R, the miss rate of VAT-Final is 25-40% lower compared to SS, VAT+Generic and VAT+Generic-NS. It can be seen that VAT+Generic and VAT+Generic-NS worsen compared to Generic for HOG and HOG-LBP (see Figure 4.8a, 4.8b) and achieve slight performance improvement when ACF is used (see Figure 4.8c). A similar performance gap exists between VAT-Final and these detectors on QMUL-J, but the miss rates of VAT+Generic and VAT+Generic-NS are the worst amongst the 10 datasets (see Figure 4.8d - 4.8f). On KWSI, VAT+Generic and VAT+Generic-NS achieve similar or lower miss rates relative to Generic; yet, the miss rates remain very poor and VAT-Final still maintains large performance margins of 30-40%(see Figure 4.8g – 4.8i). Overall, the performance of these detectors are abysmal, which suggests that dependence on generic detectors in extremely challenging target environments is highly likely to result in poor scene-specific training.

Medium datasets. The performance of SS relative to Generic is identical to the hard datasets. Except a few cases (Figure 4.9d, 4.9e), VAT+Generic and VAT+Generic-NS generally achieve similar or lower miss rates compared to VAT-Final. More notably, VAT+Generic achieves the lowest miss rates on a number of datasets (see Figure 4.9g, 4.9j, 4.9f, 4.9i and 4.9l), indicating the similarity between the distributions of INRIA and PETS datasets. Overall, the competitive performance of VAT-Final on easier datasets like PETS, as evident from all plots of Figure 4.9, validates the wide applicability of the VAT framework. Furthermore, the lower miss rates of VAT+Generic and VAT+Generic-NS reaffirms that on datasets like PETS that are less prone to dataset shift, dependence on a generic detector is unlikely to have any negative impact on scene-specific training.

4.6.4 Comparison with manually trained detectors

On MIT, the performance of VAT-Final is almost as good as Manual-Final (see Figure 4.7a – 4.7c). Notice how VAT-Initial is much worse than Manual-Initial in Figure 4.7c, but performance improves as VAT progresses and VAT-Final converges to Manual-Final. VAT achieves similar convergence on CUHK for HOG and HOG-LBP (see Figure 4.7d, 4.7e) but for ACF, a performance margin of 10% exists between VAT-Final and Manual-Final (see Figure 4.7f). On MONASH, despite the apparent

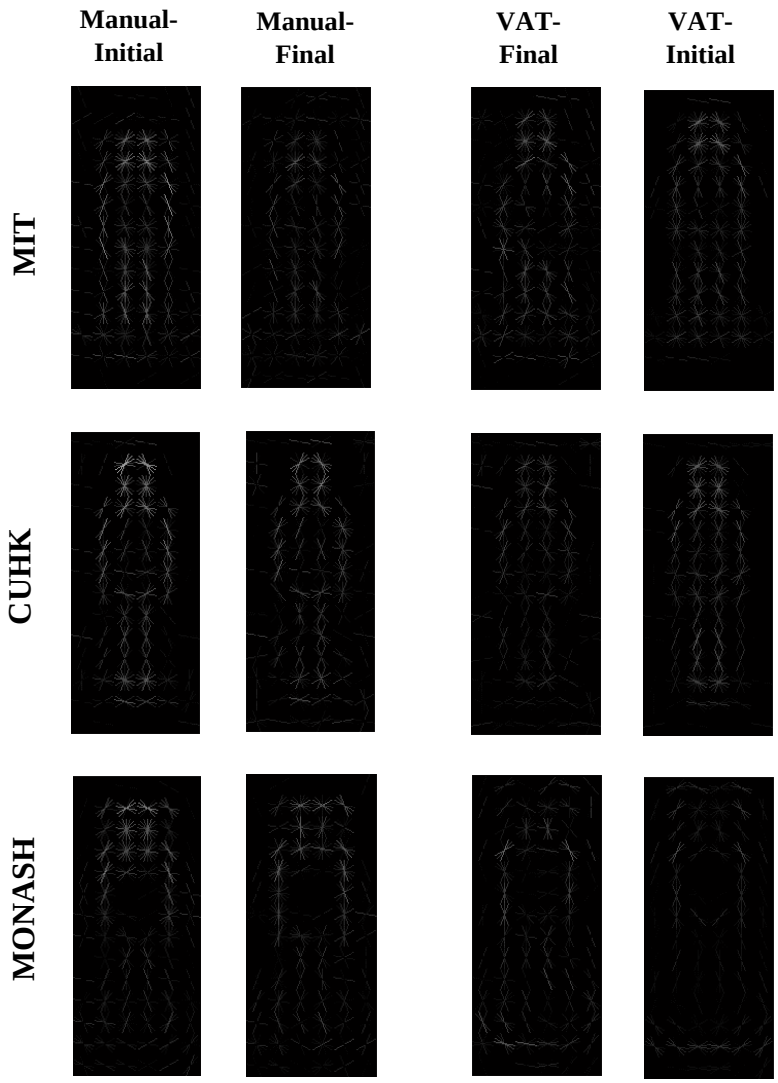


Figure 4.10: Visualization of the model weights for HOG-SVM detectors. Detectors for a particular dataset are arranged row-wise and detectors of the same type are arranged column-wise. For optimal viewing, zoom in at least 200%. For full clarity, position eye level above top of the screen and look at a downwards angle

detector improvement as VAT progresses (see Figure 4.7g – 4.7i), performance gaps of 2%, 9% and 9% remain between VAT-Final and Manual-Final, for HOG, HOG-LBP and ACF, respectively.

To visualize the "closeness" of VAT detectors to manually trained detectors, pictorial renditions of the model weights for HOG detectors trained using VAT are compared with manually trained detectors in Figure 4.10. It can be seen that initial detectors trained manually (Manual-Initial) are rather crude for all datasets – the weights are too focused along the centre of the pedestrian structure. These detectors will have high false alarm rate in the presence of upright structures like poles, traffic signals, vehicle parts or building gates, which have similar concentration of edges along the centre. As the

detector is retrained with hard negatives, the weights are spread out and redistributed to more discriminative locations (see Manual-Final).

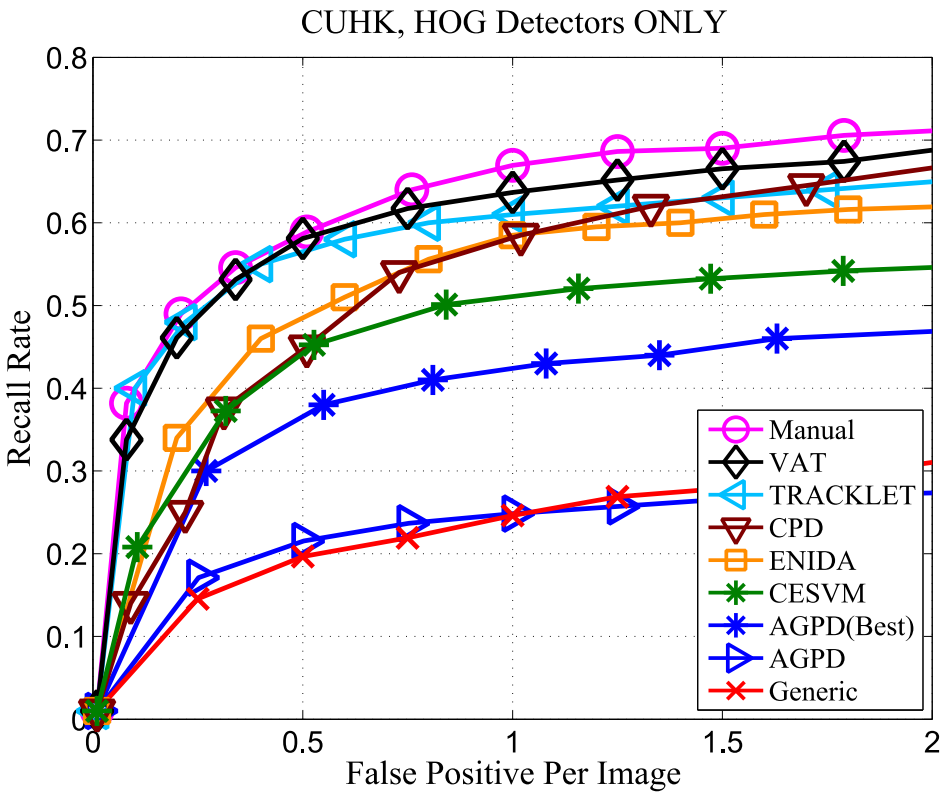
The initial detector trained by VAT during Inception (VAT-Initial) shows similar issues of limited discriminative ability as Manual-Initial. As VAT progresses to completion, careful observation shows that the position of reassigned weights in VAT-Final have high similarity to those in Manual-Final. This may explain why the performance of VAT detectors converges to that of manually trained detectors. For CUHK and MONASH, it can be seen that the intensity of the weights do have some differences but the location of the weights are very similar.

For MIT, the comparison is intriguing – the weight distribution of VAT-Final is visually more discriminative compared to Manual-Final, as it appears to capture the pedestrian structure more accurately. However, this can be explained by the fact there is a large number of very small scale and blurry pedestrians in the training set for MIT. As manual training incorporates all such instances, the model gets fine-tuned accordingly – notice how the pedestrian rendition for Manual-Final is smaller than VAT-Final due to influence from such small instances. Contrastingly, VAT may have missed such difficult instances; therefore, even though the trained model seems more representative of pedestrians, VAT-Final ultimately would miss very small, blurry instances during testing, resulting in slightly lower performance compared to Manual-Final.

4.7 Comparison with state-of-the-art

VAT is compared against the state-of-the-art on the two most commonly used datasets for benchmarking scene-specific training approaches, namely CUHK and MIT. To the best of our knowledge, the comparison against the state-of-the-art shown in Figure 4.12 is the most extensive to-date, with the largest number of scene-specific training approaches considered. However, though it can be readily assessed which approach achieves the higher performance on a given dataset, it is difficult to conclude which approach is actually the better one because while some are based on just HOG, others utilize the far superior CNN. Therefore, an additional comparison of all available scene-specific training approaches that are based on the commonly used HOG is performed and presented in Figure 4.11. For

a



b

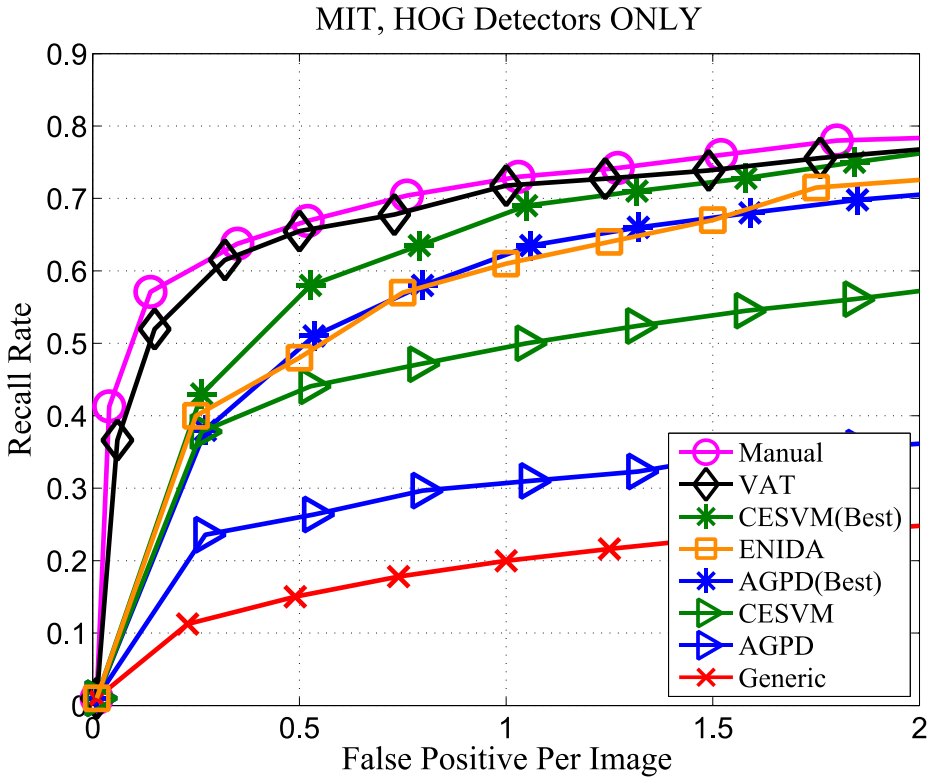
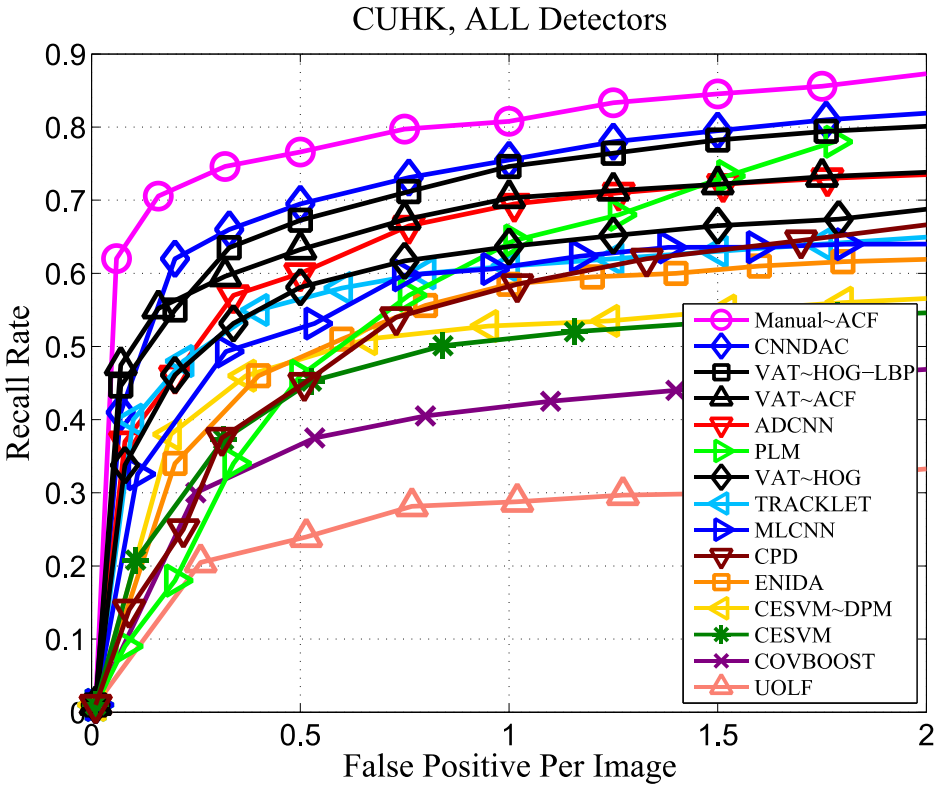


Figure 4.11: Comparison with state-of-the-art scene-specific training approaches based on HOG

a



b

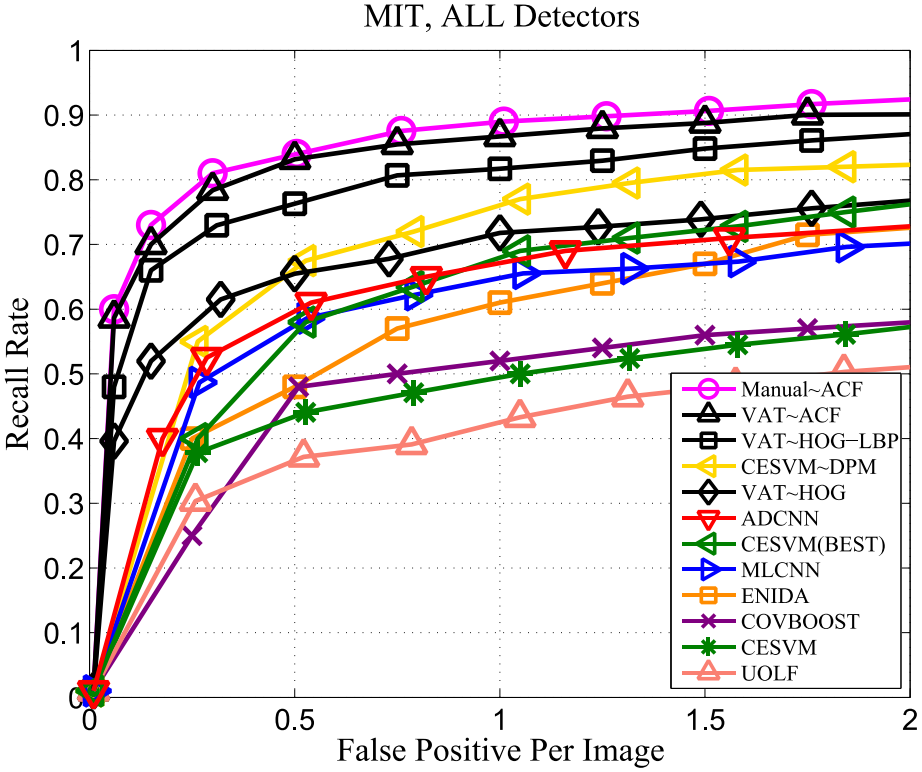


Figure 4.12: Comparison with all state-of-the-art scene-specific training approaches

any subsequent numerical performance comparisons, the detection rates at FPPI=1 are reported.

First, brief descriptions of the HOG-based methods compared in Figure 4.11 is provided. AGPD [93] transfers a generic pedestrian detector to the target scene using confident target samples acquired by exploiting multiple context cues. CESVM [75] is the extension of AGPD that assigns confidence scores to the acquired target samples, weighs source samples based on their similarity to target samples and then incorporates all of them in a Confidence-Encoded SVM. AGPD and CESVM are iterative training approaches; accordingly, AGPD(Best) and CESVM(Best) represent the converged detectors. These are often excluded during benchmarking, but have been included here for a comprehensive comparison. TRACKLET [83] obtains trajectories of the detection candidates and then applies multiple criteria to determine their labels. ENIDA [78] also employs tracking, but the tracks are labelled by spatio-temporal verification, done by applying the generic detector to classify each patch in a track. CPD [84] projects source and potential target samples onto a shared attribute subspace, and trains an attribute classifier which subsequently labels the target samples. Lastly, VAT represents the detector VAT-Final, based on HOG.

Next, the approaches compared in Figure 4.12 are described. VAT is compared against three state-of-the-art deep model based approaches – CNNDAC [86] , ADCNN[89] and MLCNN [82]. CNNDAC uses a modulating neural network to dynamically adjust the final layer of pre-trained CNN and generates a proprietary classifier for every candidate window. ADCNN transfers a pre-trained CNN by selecting useful kernels layer wise, and then embeds context information to enhance localization ability. MLCNN trains a deep-model by optimizing a multi-objective function that jointly learns discriminative features, their distributions and scene-specific visual patterns. PLM [72] treats object locations as latent variables and solves them with a progressive latent model that formulates the iterative steps of object discovery, spatial regularization and label propagation. CovBoost [74]shifts features to the most discriminative locations and scales and updates the weak classifier coefficients. UOLF [133] uses background subtraction to select target samples and CESVM~DPM applies the CESVM framework to DPM. Note that the approaches from Figure 4.11 are included again in the comparison here for the sake of completeness (AGPD is excluded, as CESVM is its extension). VAT~HOG, VAT~HOG-LBP and VAT~ACF represent the VAT-Final detectors based on HOG, HOG-LBP and ACF respectively.

According to the ROC curves of Figure 4.11a and 4.11b, VAT achieves the best performance amongst the HOG-based approaches on both CUHK and MIT, respectively. Moreover, it can also be observed that on both CUHK and MIT, the performance achieved by VAT detectors are very close to the upper bound represented by the manually trained detectors. On CUHK (see Figure 4.11a), the detection rate of VAT is the highest at 63.7%, and TRACKLET is second at 61%. On MIT (see Figure 4.11b), VAT is once again the highest at 71.8%, and CESVM(Best) is second at 69%. These results, where all assessed methods utilized the same detection algorithm, strongly indicate that VAT is a highly effective scene-specific training approach.

Figure 4.12a and 4.12b show that the performances of VAT detectors are amongst the best when compared with all the state-of-the-art scene specific training approaches. Of all the assessed methods on CUHK, only PLM, ADCNN and CNNDAC are competitive with VAT detectors. Though PLM gets 64.5% detection rate, which is just 0.9% higher than VAT~HOG, it is important to note how the performance of PLM drops almost linearly compared to VAT~HOG for FPPI<1 (see Figure 4.12a). On CUHK, CNNDAC achieves the highest detection rate at 75.5%, with VAT~HOG-LBP being a very close second at 74.6%. VAT~ACF is third at 70.28%, with ADCNN closely behind at 69.5%. All three VAT detectors outperform CESEVM~DPM and MLCNN on CUHK. On MIT (see Figure 4.12b), VAT detectors outperform all state-of-the-art approaches, except CESVM~DPM. VAT~ACF and VAT~HOG-LBP achieve the best detection rates of 86.7% and 81.7% respectively, which are both, remarkably, the highest ever reached on MIT. CESVM~DPM is third at 77% and is 4.7% lower than VAT~HOG-LBP, with VAT~HOG placing fourth at 71.8%. The deep model based approaches, ADCNN and MLCNN, are both outperformed by all three VAT detectors on MIT.

4.8 Discussions

In this section, the significance of the reported results is discussed and some final insights are provided.

4.8.1 Significant factors to consider in comparisons with state-of-the-art

Firstly, according to Figure 4.12a, CNNDAC and ADCNN achieve the 1st and 4th ranks, respectively, on CUHK while VAT~HOG-LBP and VAT~ACF place 2nd and 3rd, respectively. However, it is crucial to note that CNNDAC and ADCNN are based on CNN, which is the absolute state-of-the-art in pedestrian detection and thus, a far superior detection algorithm compared to ACF or HOG-LBP[49, 51]. Yet, 75.5% achieved by CNNDAC is only marginally better than the 74.6% achieved by VAT~HOG-LBP on CUHK. CNNDAC[86] and ADCNN[89] have already been shown to outperform various representative state-of-the-art generic CNN pedestrian detectors such as R-CNN[134], Fast R-CNN[135] and Faster R-CNN[136]. Table 4.9 compares the detection rates of the three VAT detectors against some of these state-of-the-art generic CNN pedestrian detectors, as well as the scene-specific CNN pedestrian detectors from Figure 4.12. On CUHK, VAT~HOG-LBP is almost as good as the top-performing CNN approach, while on MIT, each of the three VAT detectors outperforms all CNN based

Table 4.9: Comparison of the three VAT detectors against state-of-the-art approaches based on CNN. Methods that perform scene-specific pedestrian detection are marked with an asterisk (*). Those without an asterisk are generic pedestrian detectors. VAT detectors are highlighted in grey.

CUHK	
Method	Detection Rate
MCDNN[137]	42.0%
R-CNN[134]	42.6%
Fast R-CNN[135]	56.1%
DCNN[138]	60.5%
*MLCNN[82]	62.0%
VAT~HOG	**63.7%**
Faster R-CNN[136]	65.8%
*ADCNN[89]	69.2%
VAT~ACF	**70.3%**
VAT~HOG-LBP	**74.6%**
*CNNDAC[86]	75.5%

MIT	
Method	Detection Rate
R-CNN[134]	-
Fast R-CNN[135]	-
Faster R-CNN[136]	-
*CNNDAC[86]	-
MCDNN[137]	23.0%
DCNN[138]	43.1%
*MLCNN[82]	64.9%
*ADCNN[89]	66.8%
VAT~HOG	**71.8%**
VAT~HOG-LBP	**81.7%**
VAT~ACF	**86.7%**

approaches. The detection rate of the top-performing VAT detector, VAT~ACF is 20.1% higher than the top-performing CNN approach, ADCNN. The highly competitive performance of VAT detectors compared to the state-of-the-art CNN based approaches in Table 4.9 strongly indicate the following:

- VAT is a highly effective approach for training scene-specific pedestrian detectors
- Focusing on optimal exploitation of target samples may be more useful than developing more complex pedestrian detectors or adaptation algorithms for training better scene-specific pedestrian detectors.

Secondly, there are several pedestrians in the upper third region of the frames in CUHK but this region is excluded during detector evaluation by all the compared methods. We tackle a much harder detection challenge by searching this region in our evaluations (see detection responses on CUHK in Figure 4.14).

Thirdly, a critical observation is the performance gap between VAT detectors and ADCNN. For CUHK, the detection rate of the top performing VAT detector VAT~HOG-LBP is 5.4% higher than ADCNN; however, on MIT, VAT~ACF is 20.1% higher than ADCNN. The primary difference between MIT and CUHK is small scale, and is likely to be the influential factor for such a bigger difference in performance on MIT compared to CUHK (20.1% against 5.4%). It has been reported [89] that CNN based approaches have difficulty detecting small scale pedestrians. Therefore, there is a high probability that the performance gap would be larger on the extremely difficult datasets which have the additional complexities of poor resolution or image quality on top of small scale.

Lastly, CUHK and MIT are the most commonly used difficult datasets for evaluating scene-specific training approaches and both were created by [94]; therefore, the reliability of approaches that evaluate one dataset but exclude the other can be logically questioned. From Figure 4.11a and 4.11b, it can be observed that CPD and TRACKLET are two of the best performing approaches on CUHK, but were not tested on MIT. Similarly, CNNDAC and PLM were evaluated on CUHK (see Figure 4.12a), but not MIT (see Figure 4.12b). It is reiterated that evaluation of a scene-specific approach on a large number of datasets of varying difficulty is important and necessary to arrive at a consensus on its performance.

4.8.2 Effect of selected pedestrian detection algorithm

Between the two SVM based detectors, the richer feature set of HOG-LBP enables it to outperform HOG on every single dataset in terms of A) detection rate of VAT-Final (see Figure 4.13a) and B) performance progression of VAT measured by the performance gap between VAT-Final and VAT-Initial (compare HOG against HOG-LBP for Figure 4.7-4.9). With respect to both the afore-mentioned

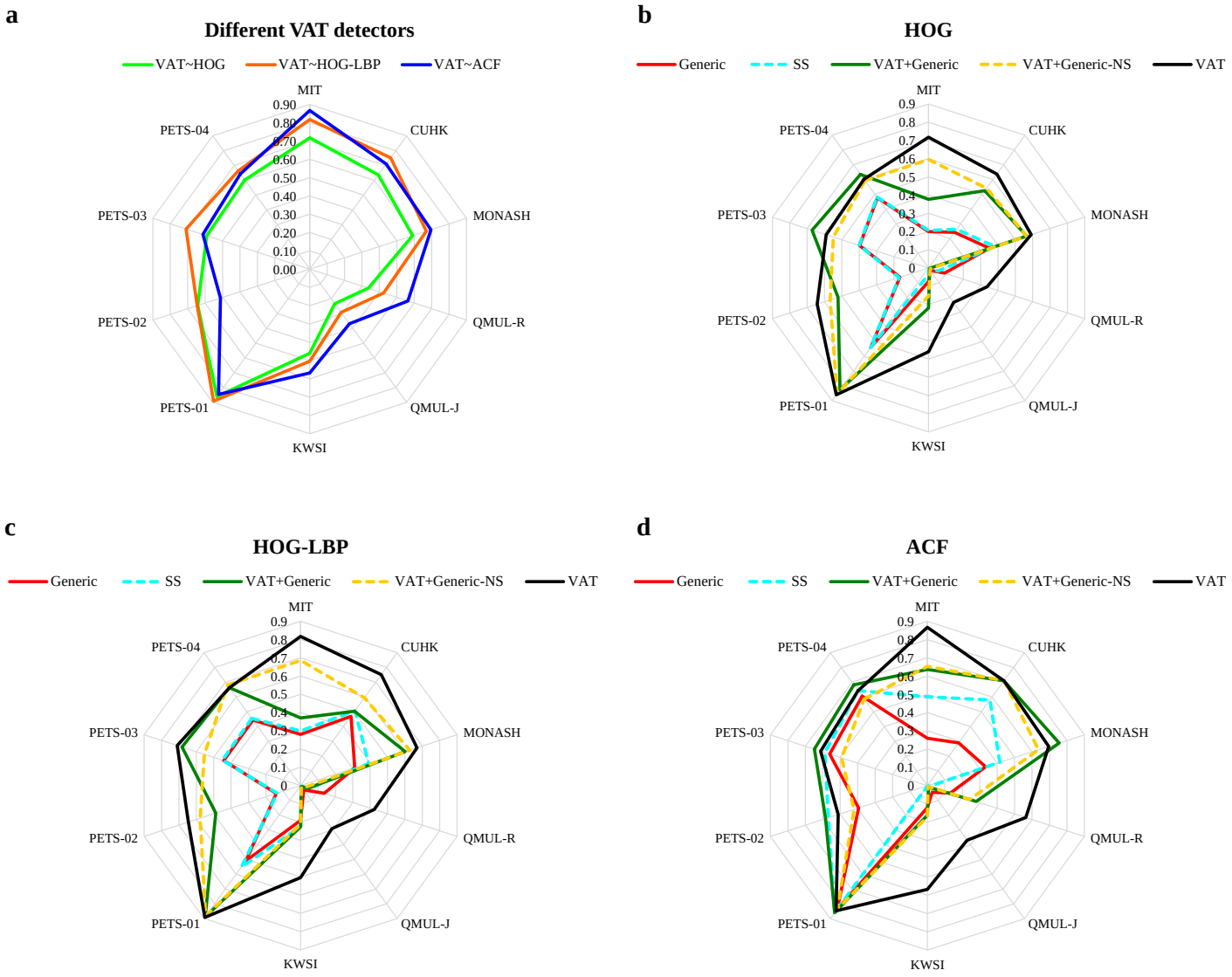


Figure 4.13: Comparison of detection rates of different detectors on all datasets. a) Comparison of HOG, HOG-LBP and ACF detectors trained using VAT. Comparison of VAT against different tested methods based on b) HOG, c) HOG-LBP and d) ACF. Manually trained detectors are not included because they are always the top-performing and are only available for MIT, CUHK and MONASH.

criteria, ACF, in turn, outperforms HOG-LBP but only on datasets with smaller pedestrian sizes - MIT, MONASH, QMUL- R, QMUL-J and KWSI (compare VAT~ACF against VAT~HOG-LBP in Figure 4.13a). For CUHK and PETS-01 – PETS-04 with larger pedestrian size (refer to Figure 4.2 for pedestrian sizes), HOG-LBP outperforms ACF (compare VAT~ACF against VAT~HOG-LBP in Figure 4.13a), which is consistent with the reported performance advantage of HOG- LBP over ACF in literature [34], on large-scale pedestrians. Further verification of lower ACF performance relative to HOG-LBP on CUHK and PETS-01 – PETS-04 can be done by comparing the number of $\Re$ acquired by their respective $\mathbf{Ж}^{RP}$ (compare Figure 4.3c and 4.3d). Lower $\Re$ results in much lower number of labelled pedestrians by X^2 for ACF compared to HOG-LBP (see Table 4.7), which may be a primary reason that VAT-Final is unable to converge to Manual-Final on CUHK (see Figure 4.7f), when ACF is being used. It must also be noted that the greater improvements of SS, VAT+Generic and VAT+Generic-NS relative to Generic when using ACF instead of HOG (compare the bottom row with the top and middle rows in Figure 4.7 and Figure 4.9) indicates that ACF is less susceptible to dataset shift compared to SVM based detectors.

4.8.3 Interpretation of TSF performance scores

TSF-Pre is an absolute metric that measures the efficiency of the TSF in terms of the percentage of correctly rejected samples, while TSF-Con is a comparative metric that measures the contribution of the TSF towards rejecting –ves in terms of the percentage of total –ves rejected by that TSF. TSF-Con is relative to other TSFs, and higher score indicates that there is a higher number of the type of –ves that can be rejected by that TSF. Based on these facts, multiple deductions can be made. TSFs in the top right portion in the plots of Figure 4.6 achieve the best performance and are the most essential (high TSF-Pre + high TSF-Con), while TSFs in the bottom left portion perform the worst and are redundant (low TSF-Pre + low TSF-Con). TSFs in the top-left corner are extremely accurate because they have high TSF-Pre scores even though there are few –ves that can be rejected (indicated by low TSF-Con scores). If majority of the TSFs have low TSF-Pre scores on a dataset, then it is impractical to apply the oracle to that dataset. It is straightforward to estimate the overall performance of an oracle on any given dataset from Figure 4.6 by making the following observations:

Figure 4.14: Qualitative detection results of best-performing detection algorithm when trained with VAT, for difficult datasets. Green bounding boxes denote correct detections, red bounding boxes denote incorrect detections and blue bounding boxes denote missed detections. Zoom in 300% for optimal viewing

- if the TSF-Pre scores of the TSFs are mostly high for a dataset, that would mean most TSFs have high precision of rejecting –ves. This equates to a high overall rejection precision of the oracle for –ves, represented by $\mathcal{X}$KPre- (percentage of correctly rejected non-pedestrians). According to the correlation discussed in subsection 4.5.1, a high $\mathcal{X}$KPre- will consequently result in a high $\mathcal{X}$KRec+.
- if the summation of the TSF-Con scores is closer to 1, that would mean most of –ves have been successfully rejected. This equates to a high recall for –ves, represented by $\mathcal{X}$KRec- (total number of correctly rejected non-pedestrians). According to the correlation discussed in subsection 4.5.1, a high $\mathcal{X}$KRec- will consequently result in a high $\mathcal{X}$KPre+.

QMUL-R, ACF

QMUL-J, ACF

KWSI, ACF

Figure 4.15: Qualitative detection results of best-performing detection algorithm when trained with VAT for extremely difficult datasets. Green bounding boxes denote correct detections, red bounding boxes denote incorrect detections and blue bounding boxes denote missed detections. Zoom in 300% for optimal viewing

4.8.4 Oracle performances

Based on the deductions in subsection 4.8.3, it is evident from Figure 4.6a that some TSFs are redundant (focus on region where TSF-Pre and TSF-Con are both < 0.3), mostly for PETS datasets with fewer – ves. Nevertheless, a few redundant TSFs cannot affect $\mathcal{X}^1$ adversely because the rejects are not used for training $\mathcal{H}^{Initial}$ and only the overall precision of $\mathcal{X}^1$ matters. Therefore, by observing the trend in Figure 4.3a, the values in Table 4.6 of $\mathcal{X}$Pre+ and particularly, the examples of labelled pedestrian instances in Figure 4.5, it can be reasoned that $\mathcal{X}^1$ performs optimal labelling on a wide variety of target scenes. On the other hand, poor TSF performance strongly influences the statistics of $\mathcal{X}^2$, which can be verified by observing the low scores of $\mathcal{X}$Pre- and $\mathcal{X}$Rec+ of the PETS datasets on Figure 4.3b, 4.3c and 4.3d as a

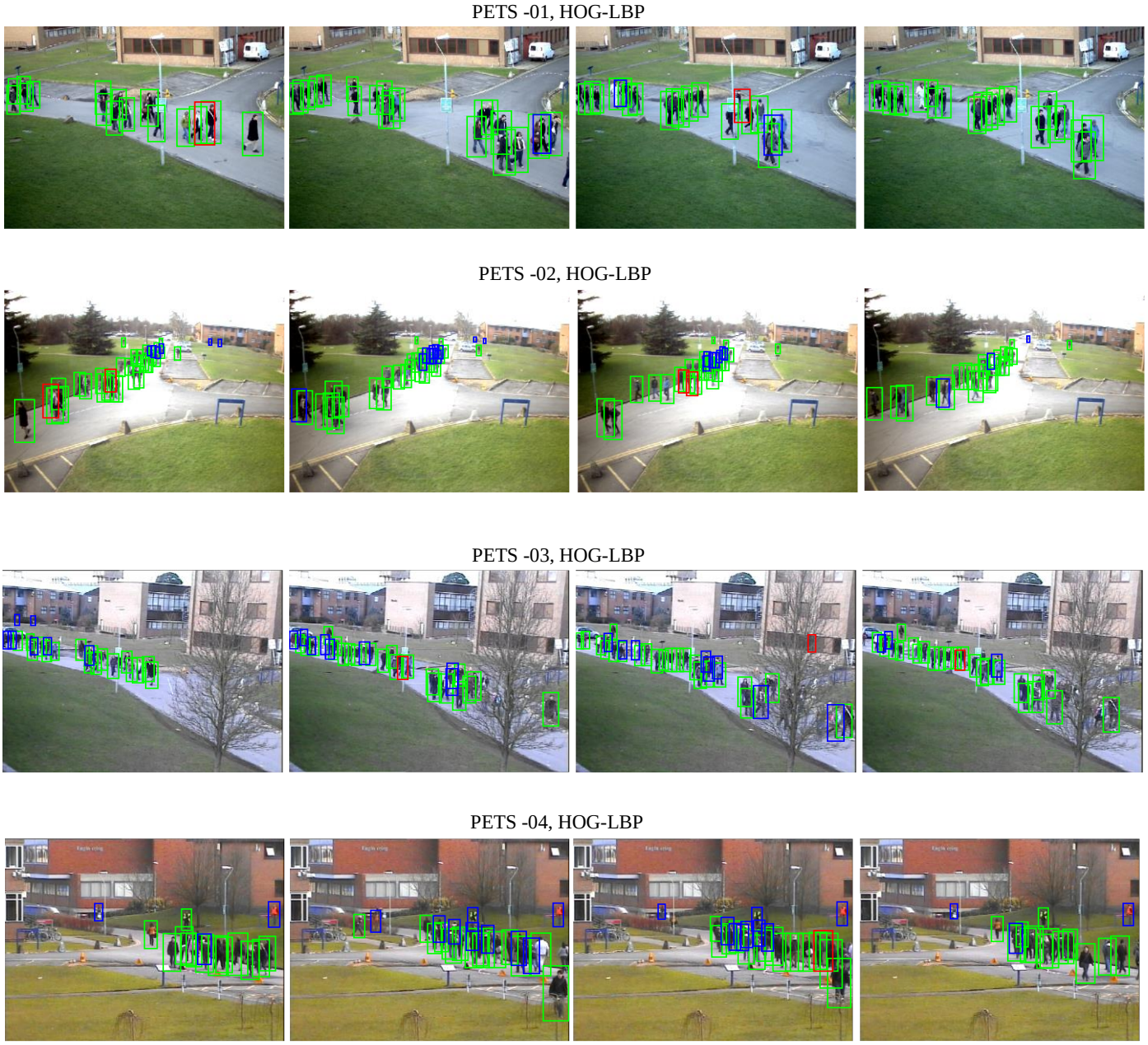


Figure 4.16: Qualitative detection results of best-performing detection algorithm when trained with VAT, for medium datasets. Green bounding boxes denote correct detections, red bounding boxes denote incorrect detections and blue bounding boxes denote missed detections. Zoom in 300% for optimal viewing.

result of low TSF-Pre scores in Figure 4.6b, 4.6c and 4.6d, respectively. Even for the hard/extremely hard datasets, $\mathbb{X}^2$ achieves accuracies of less than 90% (see Table 4.7), which equates to a possibly objectionable, albeit small, number of incorrectly labelled samples. Hence, unlike $\mathbb{X}^1$, the labelling performance of $\mathbb{X}^2$ needs further improvement in order to reach optimal labelling capability.

Table 4.10: Statistics of $Ж^{HN}$ applied to the Oracle-1 rejects

		MIT	CUHK	MONASH	QMUL-R	QMUL-J	KWSI	PETS-01	PETS-02	PETS-03	PETS-04
TN+FN/FN		1107/259	1716/337	472/155	75/14	1852/273	341/82	1145/878	1076/704	1168/350	917/514
TP+FP/ Pre/ Rec	HOG	94/0.99/0.34	223/0.98/0.65	84/0.98/0.53	2/1/0.14	43/0.98/0.15	21/1/0.26	710/0.96/0.77	584/0.94/0.78	313/0.94/0.84	508/0.94/0.93
	HOG-LBP	147/1/0.57	298/0.98/0.87	112/0.97/0.7	5/1/0.34	99/1/0.36	47/1/0.57	757/0.97/0.83	640/0.93/0.85	375/0.86/0.92	550/0.91/0.98
	ACF	186/1/0.72	345/0.95/0.97	104/0.98/0.66	9/1/0.64	250/0.98/0.9	74/0.99/0.89	728/0.95/0.79	683/0.93/0.9	421/0.81/0.97	356/0.92/0.64

4.8.5 VAT performances

Bootstrapping with hard negatives reduces the FPPI but it is usually at the expense of increased miss rate (see subsection 4.3.2). Therefore, even though the lower curves of VAT-HN relative to VAT-Initial suggest its achieved performance improvement (see Figure 4.7 and Figure 4.8), the detection rate of VAT-HN are clearly diminished compared to VAT-Initial (compare the lowest points of both curves). VAT-RP is trained to boost this reduced detection rate by bootstrapping the rejected positives as hard positives (compare the lowest points of VAT-RP and VAT-HN on Figure 4.7 and Figure 4.8). The statistics reported in Table 4.10 indicate that acquisition of rejected positives from $Neg_{Ж^1}$ is generally accurate. The higher detection rate ensures the acquisition of a higher number of $\Re$ when VAT-RP is applied to the target scene during Finalization. However, the trade-off between miss rate and false alarm rate means the FPPI of VAT-RP increases as its miss rate falls (preventing a gap to form between the curves of VAT-RP and VAT-HN), causing the performance improvement to be marginal, as reported earlier. This explains why the curves of VAT-Final are distinctly lower than VAT-RP – the augmented final positives reduces the miss rate further, but the augmented final negatives simultaneously reduces the FPPI.

Given the labelling errors of $Ж^2$ discussed previously, VAT-Final still achieves drops in miss-rate in most datasets and even converges to Manual-Final in some cases (see Figure 4.7a, 4.7c, 4.7d and 4.7e). This clearly indicates the robustness of the VAT framework despite the presence of few incorrectly labelled samples. However, a more thorough study is still required to better understand the correlation between labelling errors and VAT performance, and to determine the maximum performance achievable by VAT when labelling is optimal for both oracles.

By observing the 30 plots from Figures 4.7 – 4.9, it can be concluded that VAT does not fail, as VAT-Final never worsens relative to VAT-Initial, but is clearly more worthwhile on certain datasets.

The detection rates of all tested detectors as per table 4.8, on all datasets, are summarized using spider graphs in Figure 4.13b, 4.13c and 4.13d for HOG, HOG-LBP and ACF, respectively, where VAT corresponds to VAT~Final detector for each of the three detection algorithms. Overall, the following final interpretations can be made from Figure 4.13b – 4.13d

- On easier target scenes like PETS-01 - PETS-04, VAT does not achieve significant detector improvement relative to other approaches and may be considered overkill, regardless of the detection algorithm. Nonetheless, its performance remains competitive indicating its applicability in such easier scenes.
- On more challenging target scenes (MIT, CUHK and MONASH), VAT is the optimal approach for training scene-specific pedestrian detectors (except on MONASH based on ACF).
- On extremely challenging scenarios with prominent dataset shift (QMUL-R, QMUL-J and KWSI), VAT achieves substantial performance gaps over other approaches. It can be seen that in such scenarios, scene-specific approaches like VAT+Generic and VAT+Generic-NS that depend on pre-trained generic detectors fail, but VAT, relatively, is very effective.

4.8.6 Training time

In real-world visual surveillance environments, different variables control the times taken for different training stages of VAT, causing the total training time for generating the scene-specific pedestrian detector to considerably fluctuate from one scene to another. The number of motion regions ($\mathcal{M}$) to be acquired during the Inception stage and the number of detection responses ($\Re$) to be acquired during the Finalization stage have to be set before commencing training. These parameters act as stopping criteria for the sample acquisition steps of these stages. The values of these parameters are dependent on the scene; for an easier scene with relatively larger pedestrians and better image quality, the values can be lower but for a more difficult scene with poor resolution and small scale, the values will have to be much higher. Even if the same values are set for two different scenes, the acquisition times can be very

different based on how frequently pedestrian instances appear in the video sequences of those two scenes. Additionally, the time taken by the Bootstrapping stage can also vary depending on the number of batches/rounds needed to converge - an environment with a simple background will converge faster than one with a more complicated background.

For standard experimental datasets such as MIT or CUHK (or any other dataset listed in Section 4.1), the number of frames to be utilized for training is usually fixed and no limit is imposed on the number of acquired samples. However, one scene-specific approach may acquire a larger number of target samples than another and consequently require more training time. Under such circumstances, the longer training time cannot be considered a shortcoming if the trained detector has higher performance due to acquisition of more target samples. The same argument applies to the algorithms of different scene-specific training approaches – if one approach has a more complex algorithm with more steps/stages compared to another and therefore achieves superior performance, the quicker training time of the inferior approach is no longer a merit.

Based on the afore-mentioned discussion, there is little meaning in comparing the training times when benchmarking scene-specific training approaches. This could be the reason why scene-specific training approaches in literature do not report training times in their performance evaluations. Therefore, in a similar fashion, training times for VAT are not reported nor compared with other approaches.

5 Applications of VAT

5.1 Commercialized product based on VAT

This section describes the commercialized product that has been developed using VAT. An overview of the industry problem is presented first, followed by a description of the developed solution, and finally, the role of VAT in the developed solution is elaborated.

5.1.1 The security threat of tailgating

An access control system determines who is allowed to enter or exit a secure premise, including where and when they are allowed to enter or exit. When a credential is presented at the reader usually located at the entrance door to a secure premise, the credential information is relayed to a control panel, where the information is compared to a database of stored credentials. Depending on whether a match is found or not, access is granted or rejected. The credentials could be a card number, fingerprint or face. Almost every private facility around the world implements some form of access control.

The ultimate purpose of an access control system is to ensure that access to a secure area is restricted to "authorized" personnel only. Tailgating is the most prevalent security breach that defeats this purpose, consequently jeopardizing the privacy/safety of the information/people in a secure area. This phenomenon occurs when an authorized individual uses standard access techniques (card, fingerprint, face) to gain access to a secure area, and with or without his/her consent or knowledge, an unauthorized person enters the secure area before the door closes. There are three types of tailgating offences:

Classic Tailgating (see Figure 5.1a) occurs when an unauthorized person follows an authorized person to a restricted area without the knowledge of the authorized person. This is usually done with malicious intent.

Piggybacking (see Figure 5.1b) occurs when an unauthorized person tags along with an authorized person into a restricted area, often with the knowledge of the authorized person. This is a very common

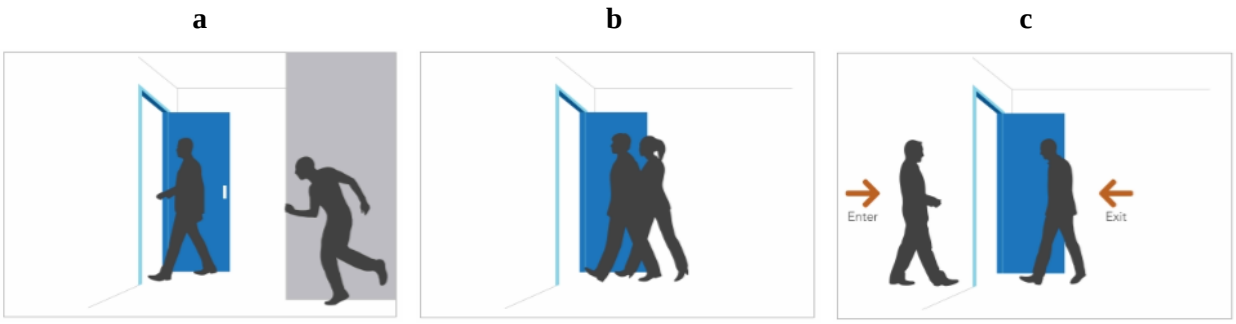


Figure 5.1: Different types of tailgating. a) Classic tailgating b) piggybacking c) crossing

scenario in most workplaces and is normally not as dangerous as classic tailgating.

Crossing (see Figure 5.1c) occurs when an authorized person leaves a secure area, and an unauthorized person seizes the opportunity to enter the secure area before the door closes. This may/may not be malicious depending on whether the "crosser" is a co-worker or an intruder.

Common scenes where tailgating is problematic are high security offices/buildings, server rooms, storage/production rooms, factories and public facilities. Existing solutions are mostly mechanical such as turnstiles, revolving doors, man-traps or lasor sensors. These are extremely expensive, intrusive, require high maintenance, difficult to install and easy to circumvent due to limited intelligence.

5.1.2 Developed anti-tailgating system: ELIDEye EV-100

Figure 5.2: ELIDEye EV-100 anti-tailgate device

ELIDEye EV-100 (see Figure 5.2) is a vision-based anti-tailgating device that can be integrated with any standard access control system to detect and alert against all tailgating offences, hence providing enhanced security. It is installed on the ceiling monitoring the secure area. A detailed illustration is shown in Figure 5.3, with relevant descriptions listed in Table 5.1. Upon installation, it can then be integrated with any standard access control. An example of such an integration is shown in Figure 5.4.

Once integration is complete, anti-tailgating can be executed as follows. When an individual badges his/her card at the entry reader (see Figure 5.4), the access control checks the database. If a match is found, access is granted by unlocking the door, but simultaneously, a signal is sent to the ELIDEye unit (see 'Valid Entry Signal' in Figure 5.4). ELIDEye then checks the actual number of entering people,

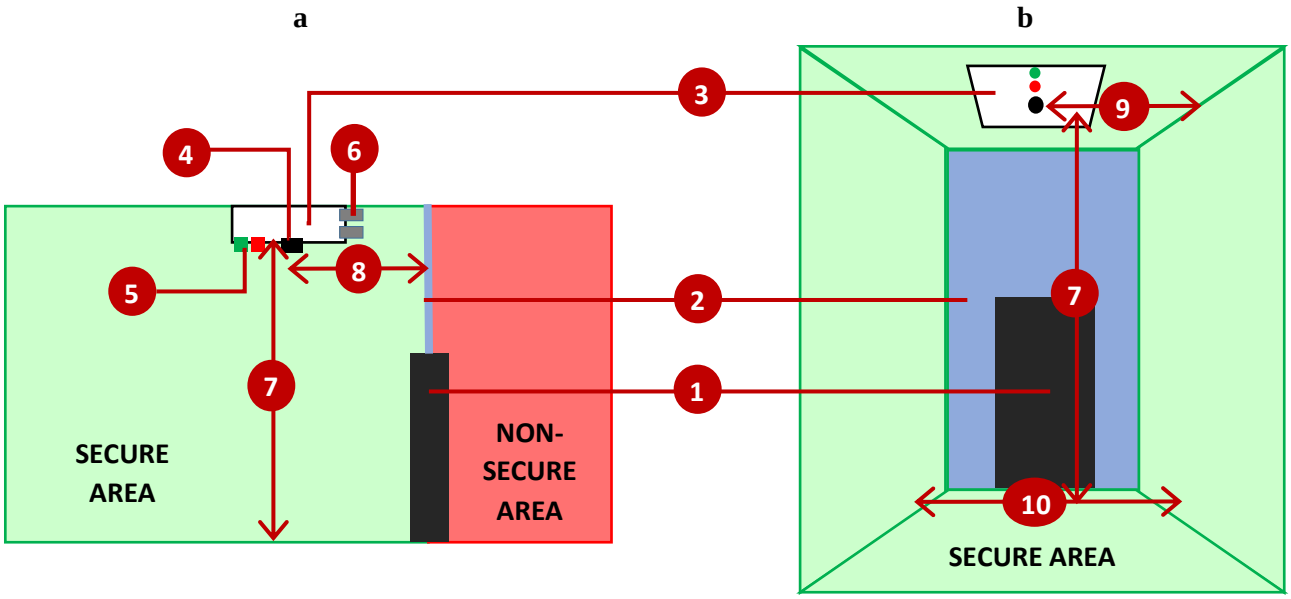


Figure 5.3: Installation of ELIDEye

Table 5.1: Installation descriptions of ELIDEye

No.	Description
1	Door
2	Partition separating secure area from non-secure area
3	ELIDEye unit
4	Camera
5	Green Indicator LED for monitoring system status
6	Various connections – Inputs/Outputs, Power & LAN
7	Device installation height
8	Distance between ELIDEye and Door
9	Distance between ELIDEye and Wall
10	Width of the secure area

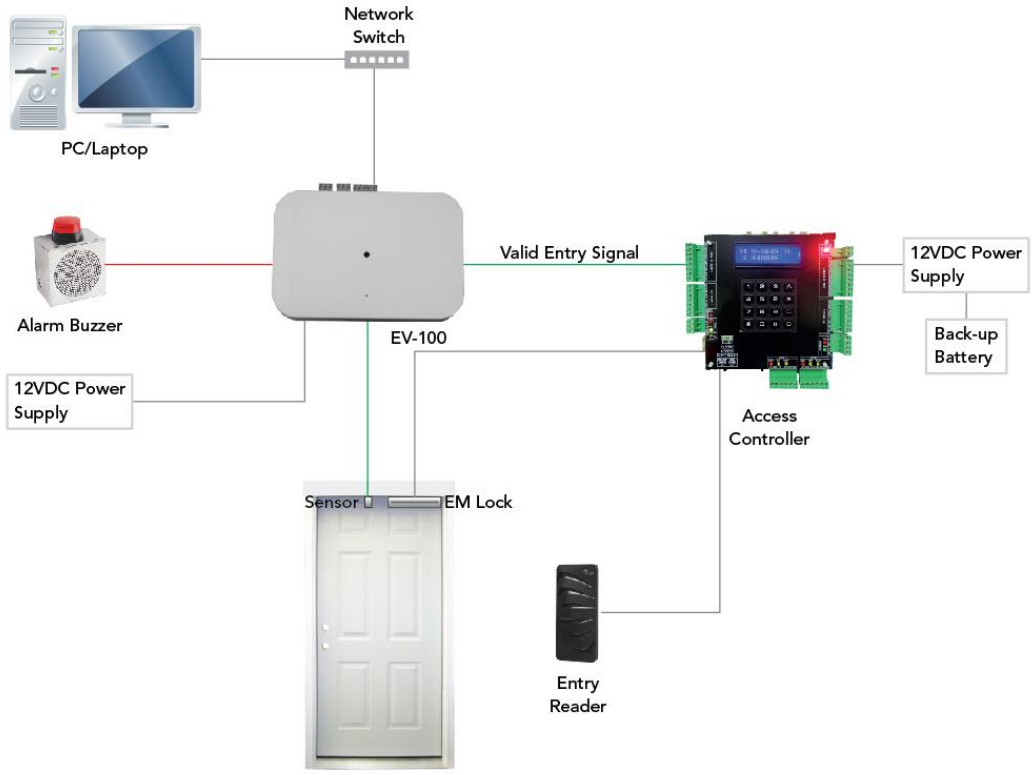


Figure 5.4: System configuration for integration of ELIDEye with standard access controller

detected by the camera, against the number of valid entry signals received from the access controller. If the number of entering people is found to be greater than the number of valid entry signals, the system triggers an alarm or other alert responses as setup by the user.

5.1.3 The role of VAT

To perform anti-tailgating, the primary function of ELIDEye is to count the number of entering and exiting people by determining the direction of moving individuals. The core technologies required to achieve this objective are overhead pedestrian detection and tracking. An illustration is shown in Figure 5.5. As the detection responses from the overhead pedestrian detection stage directly impacts the tracking performance, which ultimately governs the reliability of the system, it is of the highest importance to ensure that the accuracy of the overhead pedestrian detector is optimal.

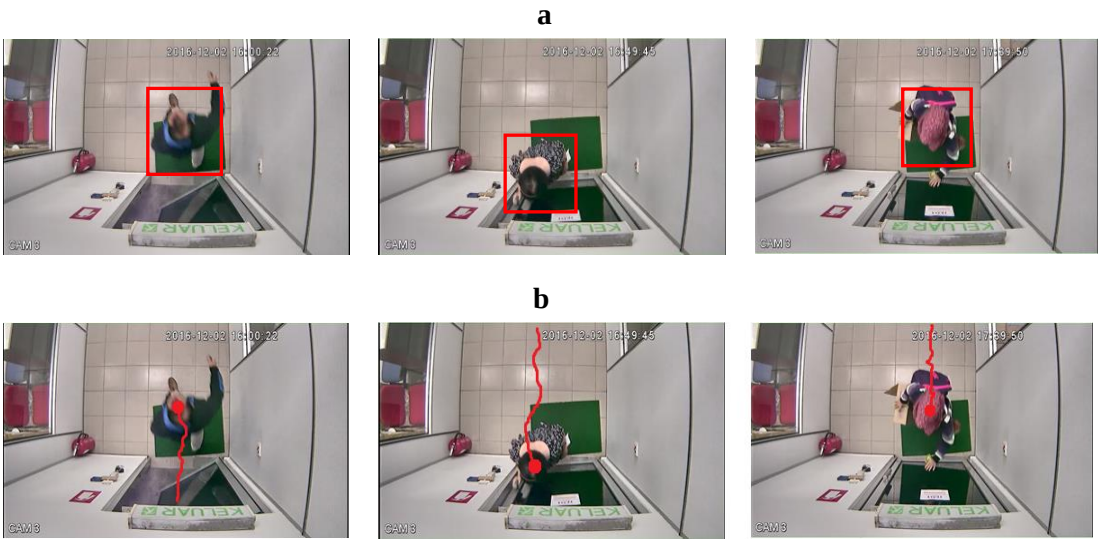


Figure 5.5: Examples of overhead pedestrian a) detection and b) tracking executed by ELIDEye

Reliable overhead pedestrian detection for anti-tailgating can be challenging due to several reasons:

- Complete absence of publicly available overhead pedestrian detectors and datasets for training overhead pedestrian detectors – this not only makes the generation of detectors difficult, but also eliminates the possibility of implementing domain adaptation algorithms that depend on pre-trained detectors.
- Poor lighting and unpredictable variations in illumination
- Limited contrast in the target environment
- Low object discriminability
- Different installation heights
- Cluttered backgrounds with objects highly similar to pedestrian instances.

To tackle the above challenges, overhead pedestrian detectors must be trained for each target environment using target samples acquired directly from the target environment. ELIDEye is designed to achieve this in an autonomous manner for different environments using VAT. The usual procedure is as follows: Once ELIDEye has been installed at the target environment and integrated with the access control system, the user can access ELIDEye from their browser (see Figure 5.4) and run VAT. Once VAT commences, it utilizes the incoming video stream from the camera module to execute the VAT training stages. Usually, it takes approximately 30 minutes to complete in "normally" busy environments

a

b

Figure 5.6: Examples of a) pedestrian instances and b) non-pedestrian instances autonomously acquired and labelled by VAT, from Site 1, during scene-specific training of the overhead-pedestrian detector.

a

b

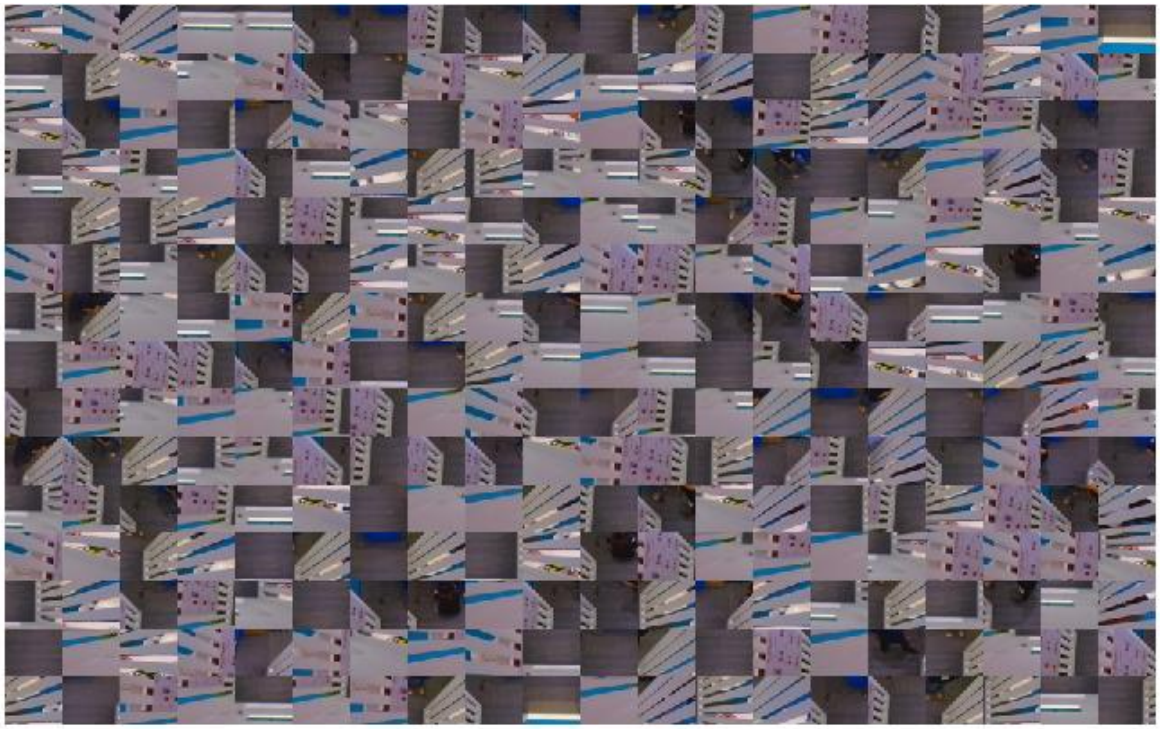


Figure 5.7: Examples of a) pedestrian instances and b) non-pedestrian instances autonomously acquired and labelled by VAT, from Site 2, during scene-specific training of the overhead-pedestrian detector

(offices or factories), but may take upto several hours if the movement in the secure area is less. Once the scene-specific detector is generated, the user is notified that the system is ready for anti-tailgating. Subsequently, when the user activates anti-tailgating, the system actively determines the number of entering/exiting people by performing overhead pedestrian detection (using the scene-specific pedestrian detector trained by VAT) and tracking.

Figure 5.6 and 5.7 depicts examples of the target samples acquired and labelled for training the overhead scene-specific pedestrian detector by VAT, in two different environments. Due to the proprietary nature of the product, exact details such as the specific samples labelled by Inception and Finalization or the types of samples rejected by the individual TSFs cannot be shared.

5.2 Extension to similar industry applications

The applicability of VAT to standard visual surveillance scenarios has been extensively validated in Chapter 4. In this section, some important applications in retail analytics are discussed. Each of these applications relies on overhead pedestrian detection; hence the developed technology for ELIDEye, based on VAT, is readily adaptable to such applications.

People counting at the entrance/exit of a store can be combined with sales data to determine the *store conversion rate*, which indicates the ratio of visitors to buyers. This metric can be valuable for store managers to evaluate the performance of the store. Furthermore, the same statistics can be used to determine the number of people within a building at different times of day; places like malls or entertainment outlets can exploit this information to improve the usage of the space, assess revenue opportunities or determine optimal opening hours.

Hot zone and dwell time is the analysis of where customers spend most of their time. This is equally as important as people counting – while it does not contribute to the evaluation of the sales efficiency discussed previously, it reveals the customers' interests to the retailers and can be utilized to restructure the store layout for optimal product positioning.

***Queue management**:* Long queues are likely to result in frustrated customers who may consequently abort their decision to purchase products. By measuring queue lengths and durations, appropriate actions can be taken, such as allocating extra personnel per queue or opening up a new cash register.

***Direction detection**:* This simple application can protect against massive losses due to theft in retail stores. By detecting people moving in the wrong direction, particularly at blocked entrances/exits in large retail outlets, it eliminates the need for the presence of physical security at such locations.

5.3 Limitations of CNN in real-world applications

Due to its undisputed superiority in classification performance, CNNs are preferred as the go-to algorithm for any image recognition task, particularly for difficult applications like pedestrian detection. This often incites the preconceived notion that selection of any other algorithm other than CNN is a sub-par approach for pedestrian detection. However, there are significant limitations on utilizing CNNs for pedestrian detection in real-world application scenarios. In this section, the major limitations are discussed and it is shown how real-time classifiers like SVM trained with VAT are more suitable for practical deployments.

Practical applications of pedestrian detection require edge computing, which means the pedestrian detectors have to run on devices with limited processing capabilities, close to the source of data (the camera), rather than running on desktop computers or cloud servers. Accordingly, the subsequent discussions are related to the deployment of pedestrian detectors on edge devices.

5.3.1 Training limitations

For training CNNs, two criteria must be fulfilled:

1) Availability of large amounts of training data
2) Hardware architecture with massive processing capability

These two criteria already make it clear that CNNs cannot be trained at the edge; popular architectures like AlexNet [47], GoogLeNet [139], VGG [140] and ResNet [141] need multiple GPUs, several gigabytes of training data and multiple days to generate the CNN. The availability of large amounts of

training data can be satisfied due to the endless stream of incoming video frames in standard surveillance scenarios, but the memory for processing (RAM) and storage (DISK space), as well as the speed of the CPUs are extremely limited.

Even if it is assumed that training can be performed on a desktop to generate the model which can subsequently be deployed to the edge device to merely execute inference, there is still an imposed need to expend on a powerful GPU. NVIDIA's Tesla and Titan GPUs are the most widely used for training CNNs and cost upwards of USD 2000. Cost constraints are a priority in most application scenarios, which makes such GPU-based training feasible to a very small minority who are ready to invest in extravagant computational resources.

The final option is to train the model using cloud services and then deploy to the edge device. However, this cannot be presumed as a one-off transaction. In realistic scenarios, it may be necessary to retrain the model and resend it to the edge. Such repetitive communications renders the whole system highly dependent on the server and mandates the need for stable connectivity. Under any network failures, the edge device will fail to update the model if required. This could be solved if the edge device could retrain the model itself without needing to depend on the central server.

5.3.2 Inference limitations

Two of the state-of-the-art CNN based detectors, with top-performance in terms of speed and accuracy are Faster R-CNN [136] and SA-Fast RCNN [51] - they require Tesla K40 and Titan X GPUs, respectively, for inference. As this computational demand during inference reiterates the limitation discussed in the previous subsection, it can be deduced that both training and inference of the top-performing CNNs on edge devices is not feasible.

To address these limitations, detectors based on simpler CNN architectures with reduced processing demands have been developed such as MobileNet [142] and SqueezeNet [143]. These architectures require less memory and execute much faster due to parallelization in the convolution layers and massive reductions in the number of parameters. However, careful analysis reveals noteworthy losses in accuracy in attempts to achieve faster inference speeds:

- Though these architectures have high information density (accuracy per parameter), their performance is 20% worse than the best CNN models [144]. Furthermore, this difference in performance is on the ImageNet benchmark dataset involving recognition of multiple object classes [145]. Pedestrian detection in challenging scenarios is a much more complex task; hence, there is high likelihood that the performance gaps will widen further for this task.
- None of the top-performing pedestrian detectors reported by the most comprehensive benchmarking for pedestrian detection [61] are based on this efficient architectures; rather, they mostly used VGG or ResNet – this is the strongest indication of the limited accuracy of these approaches that utilize simpler CNN architectures.

5.3.3 Comparison of VAT based on CNN and SVM

VAT is compatible with any detection algorithm – it boils down to which is most suitable for practical applications. Based on the discussions in the previous two subsections, Table 5.2 provides a comparison of implementing VAT with a competitive CNN approach (Faster R-CNN or SA-Fast RCNN) for an anti-tailgating solution against the current implementation of VAT based on SVM in ELIDEye. There is little dispute over the superiority of the pedestrian detector that can be generated using VAT based on CNN compared with VAT based on SVM. Unfortunately, as the statistics in Table 5.2 suggest, the computational demands of CNN make its implementation on edge devices with limited processing capabilities highly infeasible.

Table 5.2: Comparison of implementing VAT with CNN against its implementation with SVM in ELIDEye

		CNN	**SVM**
Training Requirements	Training at edge possible?	NO	YES
	# training samples required	~1000,000	< 3000
	Disk space for training	Several Gb	~10 Mb
	Memory requirements	Needs GPU	1 GB of RAM
	CPU requirements	Needs GPU	1.2 GHz single core (at edge)
	Model size	>100 Mb	~50 Kilobytes
	Training time	Several hours - days	~30 minutes
Inference Requirements	CPU requirements	Needs GPU	1.2 GHz quad-core (at edge)
	Processing speed	5-20FPS	35 FPS

6 Conclusions and Future Work

6.1 Summary of research

This research commenced by identifying the susceptibility of the state-of-the-art generic pedestrian detectors to the dataset shift problem. The progress achieved by existing scene-specific pedestrian detectors was acknowledged, but it was also shown that important practical limitations still existed – specifically, the need for manual labelling was undesirable and affected scalability, while the dependence on pre-trained generic detectors limited the applicability of the scene-specific approaches in more challenging surveillance environments. It was noted that significant effort had already been made to develop newer generic pedestrian detection as well as scene-specific ones to tackle the current limitations. Therefore, this research took a different route – the focus was shifted to design strategies that can generate optimal scene-specific pedestrian detectors through exclusive exploitation of target samples, with autonomy and practical applicability as primary design requirements.

Concretely, a paradigm shift from popular domain adaptation approaches for training scene-specific detectors was proposed to tackle the dataset shift problem as well as address the limitations of scalability and robustness. A Virtually Autonomous Training (VAT) framework was developed that trains optimal scene-specific pedestrian detectors for unseen target surveillance environments without requiring any manual labelling of target samples or utilizing any source dataset or pre-trained generic detector. To perform automatic labelling of target samples, oracles were constructed that can segregate pedestrians from non-pedestrians using a sequential combination of training sample filters designed to reject non-pedestrians. Equipped with these automated oracles for labelling target samples, VAT executes a sequence of three training stages, namely Inception, Bootstrapping and Finalization, that maximize the acquisition of target samples and iteratively improves the classifier to generate the scene-specific pedestrian detector.

Extensive experimental evaluation of the oracles and the VAT framework was carried out on 10 different datasets of varying levels of difficulty – medium, hard and extremely hard. To validate the compatibility of the VAT framework with different real-time classifiers, three different detectors were

trained with VAT, namely, HOG with SVM, HOG-LBP with SVM and ACF with AdaBoost. The performance of the VAT detectors were benchmarked against several state-of-the-art scene-specific approaches, including multiple CNN based methods.

The experimental results showed the capability of the oracles to label target samples in different kinds of surveillance environments. If the objective is the high-precision labelling of pedestrian instances only, as was the case during Inception, Oracle-1 has a precision of > 95% on all datasets. However, if high-precision labelling of both pedestrians and non-pedestrians are required, as was the case during Finalization – then the task is more challenging. The important factor was the presence of insufficient non-pedestrian instances. Overall, for more difficult datasets, Oracle-2 demonstrated better performance but in the case of easier datasets with insufficient non-pedestrian instances, TSFs become redundant as there are few non-pedestrians to reject, consequently causing a decline in the oracle performance. Analysis showed that the overall accuracy of the oracle reduced when more pedestrians were incorrectly rejected due to the presence of fewer non-pedestrians – this can be solved by relaxing the rejection criteria of the pruners.

The performance of VAT was evaluated using multiple criteria. Firstly, the progression of VAT was assessed – it was found that as training progressed, the classifier continuously improved. The improvement was more noticeable for difficult datasets. The final detector from the Finalization stage was always found to have improved relative to the initial detector from Inception stage, indicating the stability of the training framework. Secondly, when compared to generic detectors, VAT always outperformed. Thirdly, the performance of VAT was found to have converged to that of manually trained detectors for most cases. Lastly, when compared to approaches using pre-trained detectors, VAT achieved higher performance on difficult datasets and significantly outperformed on the most difficult datasets, indicating that usefulness of VAT in challenging scenarios.

On two of the most commonly used datasets for benchmarking scene-specific pedestrian detectors, VAT achieved amongst the highest performances, despite being compared to multiple CNN based approaches. On CUHK, VAT based on HOG-LBP was second, behind the top performing CNN based detector by only 0.9%. Additionally, VAT detectors outperformed multiple CNN based detectors on the CUHK dataset. On MIT, VAT based on ACF achieved the highest detection rate reported to date,

and outperformed the top-performing CNN based detector by 20%. These statistics, where CNN based detectors have been shown to be outperformed by inferior detectors trained with VAT, were strong indicators that efficient utilization of target samples is much more effective than designing more powerful models for generating optimal scene-specific pedestrian detectors.

In conclusion, the first objective of developing mechanisms to automatically label target samples was fulfilled by designing oracles. To fulfil the second objective of developing an end-to-end framework that incorporates the oracles from the first objective, and executes a sequence of training stages to generate the scene-specific pedestrian detector when applied to a target scene, VAT was developed. The third objective of extensive experimental evaluation was met by testing all components of the VAT framework on 10 different datasets of varying difficulty, validating the compatibility of VAT with three different detectors and comparing the performance of VAT against several state-of-the-art approaches. The final objective of validating the applicability of VAT in real-world surveillance scenarios was achieved by implementing VAT to develop a commercialized anti-tailgate product.

6.2 Future Work

The proposed VAT framework is a very different approach to existing literature on scene-specific training. In this thesis, the foundation of this approach has been developed. Therefore, various works are planned to extend the capabilities of the VAT framework.

6.2.1 Crowded and dynamic scenes

Exploiting motion to initialize VAT has proven highly effective for surveillance scenarios, but it is likely to be problematic if the pedestrians in the target surveillance environment are so crowded that extraction of solitary pedestrian instances becomes exceedingly difficult. To solve this, the acquisition mechanism must be based on selective search [146] rather than motion detection. Selective search can perform hierarchical segmentation on crowds to propose various possible pedestrian locations, which can then be passed to the oracles for labelling. A similar necessity arises when deploying VAT to dynamic scenes. Dependence on motion makes it difficult to apply VAT to video acquired from moving cameras. By

using selective search, sample proposals can be obtained directly from each frame and eliminate dependence on the spatiotemporal differences across multiple frames. Therefore, using selective search with VAT can extend the current capability of training pedestrian detectors in static surveillance scenes to other application domains like driver assistance systems.

6.2.2 Extension to other objects

The VAT framework has been developed, tested and implemented for pedestrian detection. However, the framework can be applied to other objects by designing the corresponding oracles. For instance, apart from pedestrians, the other common object class of interest in surveillance scenes is vehicles. Therefore, appropriate oracles can be designed to label vehicles in surveillance scenes. Provided separate oracles are available for pedestrians and vehicles, two instances of VAT can be concurrently run within the same IVS system to generate scene-specific detectors for pedestrians and vehicles.

6.2.3 Incorporation of clustering

Clustering aims to partition data into groups based on the similarities between them. The design objectives are usually to maximize the homogeneity within a group and the heterogeneity between groups. One of the most difficult data for clustering techniques is images – a significant amount of work has been done [147]. However, there appears to be no studies on clustering techniques for scene specific pedestrian detection, which is surprising, because it can be applied for unsupervised grouping of target samples into pedestrians and non-pedestrians. Two important studies will be done: a) Evaluation of different clustering techniques for grouping a given set of target samples from different surveillance environments into pedestrians and non-pedestrians b) Analysis of VAT performance by replacing Oracle-2 in finalization with clustering techniques, if part a) reveals competent labelling.

6.2.4 Long-term performance improvements

Surveillance scenes usually have an endless stream of incoming video. This allows for periodic updates of the scene-specific pedestrian detector, which essentially means repeating Stage 2 and Stage 3 of VAT

at pre-defined intervals. Exploiting the availability of new video to improve the detector is not only a rational extension, but probably necessary. This is because the VAT detector is trained within approximate 30 minutes, which is unlikely to encompass the wide range of potential scene variations. As time passes, the natural illumination may fluctuate, the background may change drastically, the lighting can be manually altered, and the most extreme of all, the camera position itself might be changed! Therefore, in real-world scenarios, it is most likely that a one-off training will be insufficient; rather the system must be designed to update itself periodically. For studies on such long-term performance, it is important to gather datasets representing video from target scenes over several days.

6.2.5 VAT on distributed systems

Distributed IVS through a large network of edge devices is the future of intelligent visual surveillance. Each device can execute VAT to generate the optimal scene-specific detector for the target scene that it is responsible for monitoring. However, in addition to distributing the computational load from a centralized server to multiple edge nodes, distributed systems can also implement inter-device communication to enhance performance. Devices can compare their respective target scenes, and if they are deemed to have sufficient similarity, the target samples labelled in one scene by a device can be shared with another device.

6.2.6 Fully Autonomous Training (FAT)

VAT is 'virtually' autonomous because the oracles have to be designed by a human. The next step would be to remove this dependence – either the labelling is done without the oracles or the oracles are designed without human assistance. At this point, it is still unclear how either of them can be accomplished; nonetheless if it can be achieved, then a Fully Autonomous Training (FAT) framework can be created. FAT would be the pinnacle of autonomous training – by merely specifying the object class of interest, the system can be instructed to generate the optimal scene-specific object detector for that particular target environment.

References

[1] Market Research Engine, "Video Surveillance and VSaaS Market by Type (IP based video surveillance and VSaaS, Analog video surveillance and VSaaS), by Component (Video surveillance as a service (VSaaS), Video surveillance software, Video surveillance hardware) and by End User – Global Market Analysis and Forecast 2022," VSM217, 2018. [Online]. Available: https://www.marketresearchengine.com/upcommingreport/video-surveillance-and-vsaas-market

[2] MarketsandMarkets, "Video Surveillance Market by System (Analog, & IP), Offering (Hardware, Software, & Service), Vertical (Commercial, Infrastructure, Military & Defense, Residential, Public Facility, & Industrial), and Geography - Global Forecast to 2023," SE 2873, May 2018. [Online]. Available: https://www.marketsandmarkets.com/Market-Reports/video-surveillance-market-645.html

[3] IHS Markit, "Top Video Surveillance Trends For 2018," December 2017. [Online]. Available: https://ihsmarkit.com/Info/1217/top-video-surveillance-trends-2018.html

[4] J. Chin and L. Lin, "China's All-Seeing Surveillance State Is Reading Its Citizens' Faces," The Wall Street Journal, June 2017. [Online]. Available: https://www.wsj.com/articles/the-all-seeing-surveillance-state-feared-in-the-west-is-a-reality-in-china-1498493020

[5] Research and Markets, "Global Video Surveillance Market Analysis and Forecast (2017-2023) - Focus on Ecosystem (Camera, Monitor, Storage, Software and Service) and Application (Infrastructure, Commercial, Residential, Industrial, Institutional and Others)," B. Research, 4413482, December 2017. [Online]. Available: https://www.researchandmarkets.com/reports/4413482/global-video-surveillance-market-analysis-and

[6] IFSEC GLOBAL, "The Video Surveillance Report 2018," 2018. [Online]. Available: https://www.ifsecglobal.com/video-surveillance-report-2018/

[7] S. A. Velastin, "CCTV Video Analytics: Recent Advances and Limitations," in *Visual Informatics: Bridging Research and Practice. Proceedings First International Visual Informatics Conference*, 2009, pp. 22-34.

[8] N. Haering, P. L. Venetlaner, and A. Lipton, "The evolution of video surveillance: an overview," *Machine Vision and Applications*, vol. 19, pp. 279-290, 2008.

[9] W. Hu, T. Tan, L. Wang, and S. Maybank, "A Survey on Visual Surveillance of Object Motion and Behaviours," *IEEE Transactions on Systems, Man, and Cybernetics - Part C : Applications and Reviews*, vol. 34, no. 3, pp. 334-352, 2004.

[10] J. Candamo, M. Shreve, D. B. Goldgof, D. B. Sapper, and R. Kasturi, "Understanding transit scenes: A survey on human behavior-recognition algorithms," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 206-224, 2010.

[11] T. D. Raty, "Survey on Contemporary Remote Surveillance Systems for Public Safety," *IEEE Transactions on Systems, Man, and Cybernetics - Part C : Applications and Reviews,* vol. 40, no. 5, pp. 493-515, 2010.

[12] X. Wang, "Intelligent multi-camera video surveillance: A review," *Pattern Recognition Letters* vol. 34, 2013.

[13] T. D'Orazio and C. Guaragnella, "A Survey of Automatic Event Detection in Multi-Camera Third Generation Surveillance Systems," *International Journal of Pattern Recognition and Artificial Intelligence,* vol. 29, no. 1, 2014.

[14] D. Elliot. (2010) Intelligent Video Solution: a Definition. *Security*. 46-48.

[15] S. K. Mishra and K. S. Bhagat, "A survey on human motion detection and surveillance," *Internation Journal of Advanced Research in Electronics and Communication Engineering,* vol. 4, pp. 1044-1048, 2015.

[16] R. K. Tripathi, A. S. Jalal, and S. C. Agrawal, "Abandoned or removed object detection from visual surveillance: a review," *Multimedia Tools and Applications,* 2018.

[17] B. Tian, B. T. Morris, M. Tang, Y. Liu, Y. Yao, C. Gou, *et al.*, "Hierarchical and Networked Vehicle Surveillance in ITS: A Survey," *IEEE Transactions on Intelligent Transportation Systems,* vol. 18, no. 1, 2017.

[18] J. Connell, Q. Fan, P. Gabbur, N. Haas, S. Pankanti, and H. Trinh, "Retail video analytics: An overview and survey," in *Proceedings of SPIE - The International Society for Optical Engineering*, 2013.

[19] J. Neves, F. Narducci, S. Barra, and H. Proença, "Biometric recognition in surveillance scenarios: a survey," *Artificial Intellgence Review,* vol. 46, no. 4, pp. 514-541, 2016.

[20] Z.-P. Bian, J. Hou, L.-P. Chau, and N. Magnenat-Thalmann, "Fall Detection Based on Body Part Tracking Using a Depth Camera," *IEEE Journal of Biomedical and Health Informatics,* vol. 19, no. 2, pp. 430-439, 2015.

[21] N. D. Bird, O. Masoud, N. P. Papanikolopoulos, and A. Isaacs, "Detection of loitering individuals in public transportation areas," *IEEE Transactions on Intelligent Tansportation Systems,* vol. 6, no. 2, pp. 167-177, 2005.

[22] R. Arroyo, J. J. Yebes, L. M. Bergasa, I. G. Daza, and J. Almaza´n, "Expert video-surveillance system for real-time detection of suspicious behaviors in shopping malls," *Expert Systems with Applications,* vol. 42, no. 21, pp. 7991-8005, 2015.

[23] Y. Gao, H. Liu, X. Sun, C. Wang, and Y. Liu, "Viloence detection using oriented violent flows," *Image and Vision Computing,* vol. 48, pp. 37-41, 2016.

[24] T. Li, H. Chang, M. Wang, B. Ni, R. Hong, and S. Yan, "Crowded Scene Analysis: A Survey," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 25, no. 3, 2015.

[25] Stratistics MRC, "Video Surveillance-Global Market Outlook (2016-2022)," 2017. [Online]. Available: http://www.orbisresearch.com/reports/index/video-surveillance-global-market-outlook-2016-2022

[26] M. Cristani, R. Raghavendra, A. D. Bue, and V. Murino, "Human behavior analysis in video surveillance: A Social Signal Processing perspective," *Neurocomputing,* vol. 100, pp. 86-97, 2013.

[27] M. Cristani, M. Farenzena, D. Bloisi, and V. Murino, "Background subtraction for automated multisensor surveillance: A comprehensive review," *Eurasip Journal on Advances in Signal Processing,* 2010.

[28] X. Zhang, Y.-H. Yang, Z. Han, H. Wang, and C. Gao "Object Class Detection: A Survey," *ACM Computing Surveys,* vol. 46, no. 1, 2013.

[29] A. Yilmaz, O. Javed, and M. Shah, "Object Tracking: A Survey," *ACM Computing Surveys,* vol. 38, no. 4, 2006.

[30] O. P. Popoola and K. Wang, "Video-based abnormal human behavior recognition - a review," *IEEE TRAnsactions on Systems, Man, and Cybernetics - Part C : Applications and Reviews,* vol. 42, no. 6, pp. 865-878, 2012.

[31] A. B. Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: A review," *Expert Systems with Applications,* vol. 91, pp. 480-491, 2017.

[32] D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf, "Survey of Pedestrian Detection for Advanced Driver Assistance Systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 32, no. 7, pp. 1239-1258, 2010.

[33] M. Enzweiler and D. M. Gavrilla, "Monocular Pedestrian Detection: Survey and Experiments," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 31, no. 12, pp. 2179-2195, 2009.

[34] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection : An Evaluation of the State of the Art," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 34, no. 4, pp. 743-761, 2012.

[35] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned?," in *Computer Vision - ECCV 2014 Workshops. Proceedings: LNCS 8926*, 2015, pp. 613-27.

[36] H. A. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 20, no. 1, pp. 23-38, 1998.

[37] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences,* vol. 55, no. 1, pp. 119-139, 1997.

[38] C. Papgeorgiou and T. Poggio, "A trainable system for object detection," *International Journal of Computer Vision,* vol. 38, no. 1, pp. 15-33, 2000.

[39] P. A. Viola and M. J. Jones, "Robust Real-Time Face Detection," *International Journal of Computer Vision,* vol. 60, no. 2, pp. 91-110, 2004.

[40] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 886-893.

[41] P. Dollar, Z. Tu, P. Perona, and S. Belongpie, "Integral Channel Features," in *Proceedings of British Machine Vision Conference*, 2009.

[42] P. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained Part Based Models," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1627-1645, 2010.

[43] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool, "Pedestrian detection at 100 frames per second," in *2012 IEEE Conference on Computer Vision and Pattern Recognition* 2012, pp. 2903-10.

[44] P. Dollar, "Fast Feature Pyramids for Object Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 8, pp. 1532-1545, 2014.

[45] S. Zhang, R. Benenson, and B. Schiele, "Filtered channel features for pedestrian detection," in *Proceedings on the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 1751-60.

[46] E. Ohn-Bar and M. M. Trivedi, "To Boost or Not to Boost? On the Limits of Boosted Trees for Object Detection," in *International Conference on Pattern Recognition*, 2016.

[47] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, vol. 60, no. 6, 2017.

[48] Z. Cai, Q. Fan, R. S. Feris, and N. Vasconcelos, "A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection," in *ECCV*, 2016, pp. 354-370.

[49] X. Du, M. El-khamy, J. Lee, and L. Davis, "Fused DNN: A deep neural network fusion approach to fast and robust pedestrian detection," in *2017 IEEE Winter Conference on Applications of Computer Vision*, 2017, pp. 953-961.

[50] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan, and X. Wang, "(in press).Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[51] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-Aware Fast R-CNN for Pedestrian Detection," *IEEE Transactions on Multimedia*, vol. 20, no. 4, 2018.

[52] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. D. Lawrence, *Dataset shift in Machine Learning (Neural Information Processing Series)*. Cambridge, MA, USA: MIT Press, 2008.

[53] J. G. Moreno-Torres, T. Raeder, R. Alaiz-Rodriquez, N. V. Chawla, and F. Herrera, "A unifying view on dataset shift in classification," *Pattern Recognition*, vol. 45, pp. 521-530, 2012.

[54] K. K. Htike and D. Hogg, "Adapting pedestrian detectors to new domains: A comprehensive review," *Engineering Applications of Artificial Intelligence*, vol. 50, pp. 142-158, 2016.

[55] Z.-Q. Zhao, P. Zheng, S.-t. Xu, and X. Wu, "Object Detection with Deep Learning: A Review," *Journal of Latex Class Files*, vol. 14, no. 8, 2017.

[56] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian Detection: A Benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Patter Recognition*, 2009.

[57] A. Ess, B. Leibe, and L. Van Gool, "Depth and Appearance for Mobile Scene Analysis," in *Proceedings of IEEE Internation Conference on Computer Vision*, 2007.

[58] C. Wojek, S. Walk, and B. Schiele, "Multi-Cue Onboard Pedestrian Detection," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[59] (2017). *Paris, France - November 6, 2017:A bird`s eye view of a spacious city intersection roundabout with many cars, trucks and buses on in. Pedestrians are waiting green light*. Available: https://www.videoblocks.com/video/paris-france---november-6-2017a-birds-eye-view-of-a-spacious-city-intersection-roundabout-with-many-cars-trucks-and-buses-on-in-pedestrians-are-waiting-green-light-rism03xfgjekqdhag

[60] (2012). *A top down view of Pedestrians.* Available: https://www.google.com/imgres?imgurl=https%3A%2F%2Fi.ytimg.com%2Fvi%2Fp_GKeQvlFlA%2Fmaxresdefault.jpg&imgrefurl=https%3A%2F%2Fwww.youtube.com%2Fwatch%3Fv%3Dp_GKeQvlFlA&docid=dA7NA4igNUkF5M&tbnid=s7Y-s2S3S5WjJM%3A&vet=10ahUKEwi0k7DqvvvdAhWJpY8KHcljBxUQMwg6KAEwAQ..i&w=1280&h=720&bih=626&biw=1366&q=overhead%20view%20pedestrians&ved=0ahUKEwi0k7DqvvvdAhWJpY8KHcljBxUQMwg6KAEwAQ&iact=mrc&uact=8

[61] P. Dollar. *Caltech Pedestrian Detection Benchmark*. Available: http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/

[62] S. Sternig, P. M. Roth, and H. Bischof, "Learning of Scene-Specific Object Detectors by Classifier Co-Grids," in *Proceedings 7th IEEE International Conference on Advanced Video and Signal Based Surveillance* 2010, pp. 408-13.

[63] P. M. Roth, H. Grabner, D. Skocaj, H. Bishof, and A. Leonardis, "On-line conservative learning for person detection," in *Proceedings of the IEEE International Workshop on PETS*, 2005.

[64] O. Javed, S. Ali, and M. Shah, "Online detection and classification of moving objects using progressively improving detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005, pp. 696-701.

[65] M. Munaro and A. Cenedese, "Scene specific people detection by simple human interaction," presented at the IEEE International Conference on Computer Vision Workshops, 2011.

[66] A. J. Joshi and F. Porikli, "Scene-Adaptive Human Detection with Incremental Active Learning," in *Proceedings of the 2010 20th International Conference on Pattern Recognition* 2010, pp. 2760-2763.

[67] Y. Abramson and Y. Freund, "Semi-Automatic Visual Learning (SEVILLE): A Tutorial on Active Learning for Visual Object Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2005.

[68] B. Wu and R. Nevatia, "Improving part based object detection by unsupervised, online boosting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007, pp. 1-8.

[69] Z. Qi, Y. Xu, L. Wang, and Y. Song, "Online multiple instance boosting for object detection," *Neurocomputing*, vol. 74, pp. 1769-1775, 2011.

[70] P. Sharma, C. Huang, and R. Nevatia, "Unsupervised Incremental Learning for Improved Object Detection in a Video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3298-305.

[71] K. K. Htike and D. Hogg, "Weakly supervised pedestrian detector training by unsupervised prior learning and cue fusion in videos," in *ICIP*, 2014, pp. 2338-42.

[72] Q. Ye, T. Zhang, W. Ke, Q. Qiu, J. Chen, G. Sapiro, *et al.*, "Self-Learning Scene-Specific Pedestrian Detectors Using a Progressive Latent Model," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 2057-66.

[73] X. Cao, Z. Wang, P. Yan, and X. Li, "Rapid pedestrian detection in unseen scenes," *Neurocomputing*, vol. 74, pp. 3343-3350, 2011.

[74] J. Pang, Q. Huang, S. Yan, S. Jiang, and L. Qin, "Transferring boosted detectors towards viewpoint and scene adaptiveness," *IEEE Transactions on Image Processing*, vol. 20, no. 5, pp. 1388-1400, 2011.

[75] X. Wang, M. Wang, and W. Li, "Scene-Specific Pedestrian Detection for Static Video Surveillance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 361-374, 2014.

[76] X. Cao, L. Wang, B. Ning, Y. Yuan, and P. Yan, "Pedestrian detection in unseen scenes by dynamically updating visual words," *Neurocomputing*, vol. 119, pp. 232-242, 2013.

[77] X. Cao, Z. Wang, P. Yan, and Y. Li, "Transfer learning for pedestrian detection," *Neurocomputing*, vol. 100, pp. 51-57, 2013.

[78] K. K. Htike and D. Hogg, "Efficient Non-iterative domain adaptation of pedestrian detectors to video scenes 2014," in *2014 22nd International Conference on Pattern Recognition (ICPR)*, 2014, pp. 654-9.

[79] F. Liang, S. Tang, Y. Zhang, Z. Xu, and J. Li, "Pedestrian detection based on sparse coding and transfer learning," *Machine Vision and Applications*, vol. 25, no. 7, pp. 1697-1709, 2014.

[80] D. Vazquez, A. M. Lopez, J. Marin, D. Ponsa, and D. Geronimo, "Virtual and Real World Adaptation for Pedestrian Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 4, pp. 797-809, 2014.

[81] J. Xu, S. Ramos, D. Vazquez, and A. M. Lopez, "Domain Adaptation of Deformable Part-Based Models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 12, pp. 2367-2380, 2014.

[82] X. Zeng, W. Ouyang, M. Wang, and X. Wang, "Deep Learning of Scene-Specific Classifier for Pedestrian Detection," in *Computer Vision - ECCV 2014. 13th European Conference. Proceedings*, 2014, pp. 472-487.

[83] Y. Mao and Z. Yin, "Training a Scene-Specific Pedestrian Detector Using Tracklets," in *2015 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015, pp. 170-6.

[84] X. Zhang, F. He, L. Tian, and L. Wang, "Cognitive pedestrian detector : Adapting detector to specific scene by transferring attributes," *Neurocomputing*, vol. 149, pp. 800-810, 2015.

[85] J. Xu, S. Ramos, D. Vázquez, and A. M. ópez, "Hierarchical adaptive structural SVM for domain adaptation," *International Journal of Computer Vision*, vol. 119, pp. 159-178, 2016.

[86] S. Tang, M. Ye, C. Zhu, and Y. Liu, "Adaptive pedestrian detection using convolutional neural network with dynamically adjusted classifier," *Journal of Electronic Imaging*, vol. 26, 2017.

[87] H. Hattori, N. Lee, V. N. Boddeti, F. Beainy, K. M. Kitani, and T. Kanade, "(in press). Synthesizing a Scene-Specific Pedestrian Detector and Pose Estimatorfor Static Video Surveillance," *International Journal of Computer Vision*, pp. 1-18, 2018.

[88] S. Tang, M. Ye, P. Xu, and X. Li, "(in press). Adaptive pedestrian detection by predicting classifier," *Neural Computing and Applications*, 2017.

[89] X. Li, M. Ye, Y. Liu, and C. Zhu, "(in press). Adaptive Deep Convolutional Neural Networks for Scene-Specific Object Detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[90] H. Detmold, A. Van Den Hengel, A. Dick, A. Cichowski, R. Hill, E. Kocadag, *et al.*, "Estimating camera overlap in large and growing networks," in *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, 2008.

[91] V. S. Kenk, R. Mandeljc, S. Kovačič, M. Kristan, M. Hajdinjak, and J. Perš, "Visual re-identification across large, distributed camera networks," *Image and Vision Computing*, vol. 34, pp. 11-26, 2015.

[92] C. C. Loy, T. Xiang, and S. Gong, "Stream-based Active Unusual Event Detection," in *Proceedings of Asian Conference on Computer Vision*, 2010, pp. 161-175.

[93] M. Wang and X. Wang, "Automatic Adaptation of a generic pedestrian detector to a specific traffic scene," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011, pp. 3401-3408.

[94] M. Wang, W. Lei, and X. Wang, "Transferring a generic pedestrian detector towards specific scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3274-81.

[95] J. Ferryman and A. Shahrokni, "Pets2009: Dataset and challenge," in *Twelfth IEEE Int'l workshop on performance evaluation of tracking and surveillance*, 2009.

[96] K. Nasrollahi and T. B. Moeslund, "Extracting a Good Quality Frontal Face Image from a Low-Resolution Video Sequence," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, no. 10, pp. 1353-62, 2011.

[97] M. Sodanil and C. Intarat, "A Development of Image Enhancement for CCTV Images," in *5th International Conference on IT Convergence and Security (ICITCS)*, 2015.

[98] C. Henderson, S. G. Blasi, F. Sobhani, and E. Izquierdo, "On the impurity of street-scene video footage," in *6th International Conference on Imaging for Crime Prevention and Detection (ICDP-15)*, 2015.

[99] C. Henderson and E. Izquierdo, "Robust Feature Matching in the Wild," in *2015 Science and Information Conference (SAI)*, 2015, pp. 628-37.

[100] A. Tsifouti, S. Triantaphillidou, M.-C. Larabi, E. Bilissi, and A. Psarrou, "A case study in identifying acceptable bitrates for human face recognition tasks," *Signal Processing: Image Communication*, vol. 36, pp. 14-28, 2015.

[101] E. Gaillard, "Police and Surveillance Control Room," Reuters. [Online]. Available: http://www.pyrotechworkspace.com/police-and-surveillance-control-room-2/

[102] D. Russell and S. Gong, "Exploiting periodicity in recurrent scenes," in *Proceedings of the British Machine Vision Conference*, 2008.

[103] M. Haag and H.-H. Nagel, "Tracking of Complex Driving Manoeuvres in Traffic Image Sequences," *Image and Vision Computing*, vol. 16, no. 8, 1998.

[104] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth, "Describing Objects by their Attributes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* 2009, pp. 1778-85.

[105] S. Zhang, R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Towards Reaching Human Performance in Pedestrian Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, 2018.

[106] A. Levin, P. Viola, and Y. Freund, "Unsupervised improvement of visual detectors using co-training," in *Proc. IEEE International Conference on Computer Vision*, 2003, pp. 626-633.

[107] P. Roth, S. Sterning, H. Grabner, and H. Bishof, "Classifier grids for robust adaptive object detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops* 2009, pp. 2727-34.

[108] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-supervised self-training of object detection models," in *Proc. IEEE Workshop on Application of Computer Vision*, 2005, pp. 29-36.

[109] H. Grabner and H. Bischof, "On-line boosting and vision," in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, 2006, pp. 260-267.

[110] H. Grabner, C. Leistner, and H. Bischof, "Semi-supervised On-Line Boosting for Robust Tracking," in *Proceedings 10th European Conference on Computer Vision*, 2008, pp. 234-247.

[111] B. Babenko, M. H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 983-90.

[112] B. Zeisl, C. Leistner, and A. Saffari, "On-line semi-supervised multiple-instance boosting," in *Proceeding of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010, pp. 1879-1886.

[113] Z. Qi, Y. Xu, L. Wang, and Y. Song, "Online Multiple Instance Boosting For Object Detection," *Neurocomputing*, vol. 74, 2011.

[114] Y. Li-ping, T. Huan-ling, and A. Zhi-yong, "Domain Adaptation for Pedestrian Detection Based on Prediction Consistency," *Scientific World Journal*, 2014.

[115] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang, "Designing category-level attributes for discriminative visual recognition," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2013, pp. 771-778.

[116] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-Based Classification for Zero-Shot Visual Object Categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 2, pp. 453-65, 2014.

[117] P. Tome, J. Fierrez, R. Vera-Rodriguez, and M. S. Nixon, "Soft biometrics and their application in person recognition at a distance," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 3, pp. 464-472, 2014.

[118] D. A. Reid, M. S. Nixon, and S. V. Stevenage, "Soft biometrics; human identification using comparative descriptions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1216-28, 2014.

[119] M. Liu, D. Zhang, and S. Chen, "Attribute relation learning for zero-shot classification," *Neurocomputing*, vol. 139, pp. 34-46, 2014.

[120] F. Li, S.-R. Zhou, J.-M. Zhang, D.-Y. Zhang, and L.-Y. Xiang, "Attribute-based knowledge transfer learning for human pose estimation," *Neurocomputing*, vol. 116, pp. 301-10, 2013.

[121] S. Prokopenko. (2013, 1 July 2018). *Human Figure Proportions – Average Figures – Dr. Paul Richer*. Available: http://www.proko.com/human-figure-proportions-average-richer/#.Wziiv9Izbcs

[122] M.-P. Dubuisson and A. K. Jain, "A Modied Hausdorff Distance for Object Matching," in *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 1994, pp. 566-568.

[123] S. Walk, N. Majer, K. Schindler, and B. Schiele, "New Features and Insights for Pedestrian Detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[124] D. R. Wilson and T. R. Martinez, "The general inefficiency of batch training for gradient descent learning," *Neural Networks*, vol. 16, no. 1, pp. 1429-1451, 2003.

[125] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in *Proceedings of COMPSTAT 2010 - 19th International Conference on Computational Statistics, Keynote, Invited and Contributed Papers*, 2010, pp. 177-186.

[126] M. Li, T. Zhang, Y. Chen, and A. J. Smola, "Efficient mini-batch training for stochastic optimization," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2014, pp. 661-670.

[127] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[128] S. Chowdhury. (2018). *Video surveillance datasets for testing scene-specific pedestrian detectors*. Available: https://osf.io/jy58f

[129] X. Wang, T. X. Han, and S. Yan, "An HOG-LBP Human Detector with Partial Occluion Handling," in *Proceedings of the IEEE International Conference on Computer Vision*, 2009.

[130] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971-987, 2002.

[131] A. Vedaldi and B. Fulkerson, "Vlfeat - An open and portable library of computer vision algorithms," in *MM'10 - Proceedings of the ACM Multimedia 2010 International Conference*, 2008, pp. 1469-1472.

[132] R.-E. Fang, K.-W. Chang, C.-J. Hseieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871-1874, 2008.

[133] V. Nair and J. J. Clark, "An unsupervised, online learning framework for moving object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2004, pp. 317-324.

[134] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 580–587.

[135] R.Girshick, "Fast r-cnn," in *Proceedings of the IEEE Internation Conference on Computer Vision*, 2015, pp. 1440-1448.

[136] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, 2017.

[137] X. Zeng, W. Ouyang, and X. Wang, "Multi-stage contextual deep learning for pedestrian detection," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 121-128.

[138] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proceedings of the Advancements in Neural Information Processing Systems*, 2013, pp. 2553-2561.

[139] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, *et al.*, "Going deeper with convolutions," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[140] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Internation Conference on Learning Representations*, 2015.

[141] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[142] A. Howard, M. Zhu, B. Chen, D. Kalenichenko, T. W. W., M. Andreetto, *et al.*, "Mobilenets:Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[143] F. Iandola, S. Han, M. Moskewicz, K. Ashraf, W. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size," *arXiv preprint arXiv:1602.07360*, 2016.

[144] A. Wong, "NetScore: Towards Universal Metrics for Large-scale Performance Analysis of Deep Neural Networks for Practical On-Device Edge Usage," *arXiv:1806.05512*, 2018.

[145] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, *et al.*, "ImageNet Large Scale Visual Recognition Challenge," *arXiv:1409.0575,* 2015.

[146] K. E. V. d. Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders, "Segmentation as selective search for object recognition " in *Proceedings of the IEEE International Conference on Computer Vision,* 2011, pp. 1879-1886.

[147] S. Wazarkar and B. N. Keshavamurthy, "A survey on image data analysis through clustering techniques for real world applications," *Journal of Visual Communication and Image Representation,* vol. 55, pp. 596-626, 2018.

Appendices

Appendix A: Samples labelled as pedestrians by Oracle-1

This appendix displays a subset of approximately the first 500 instances from the samples labelled as pedestrians by Oracle-1, for all 10 datasets. In the cases where fewer samples are displayed, the displayed samples represent the whole sample set and not a subset

MIT

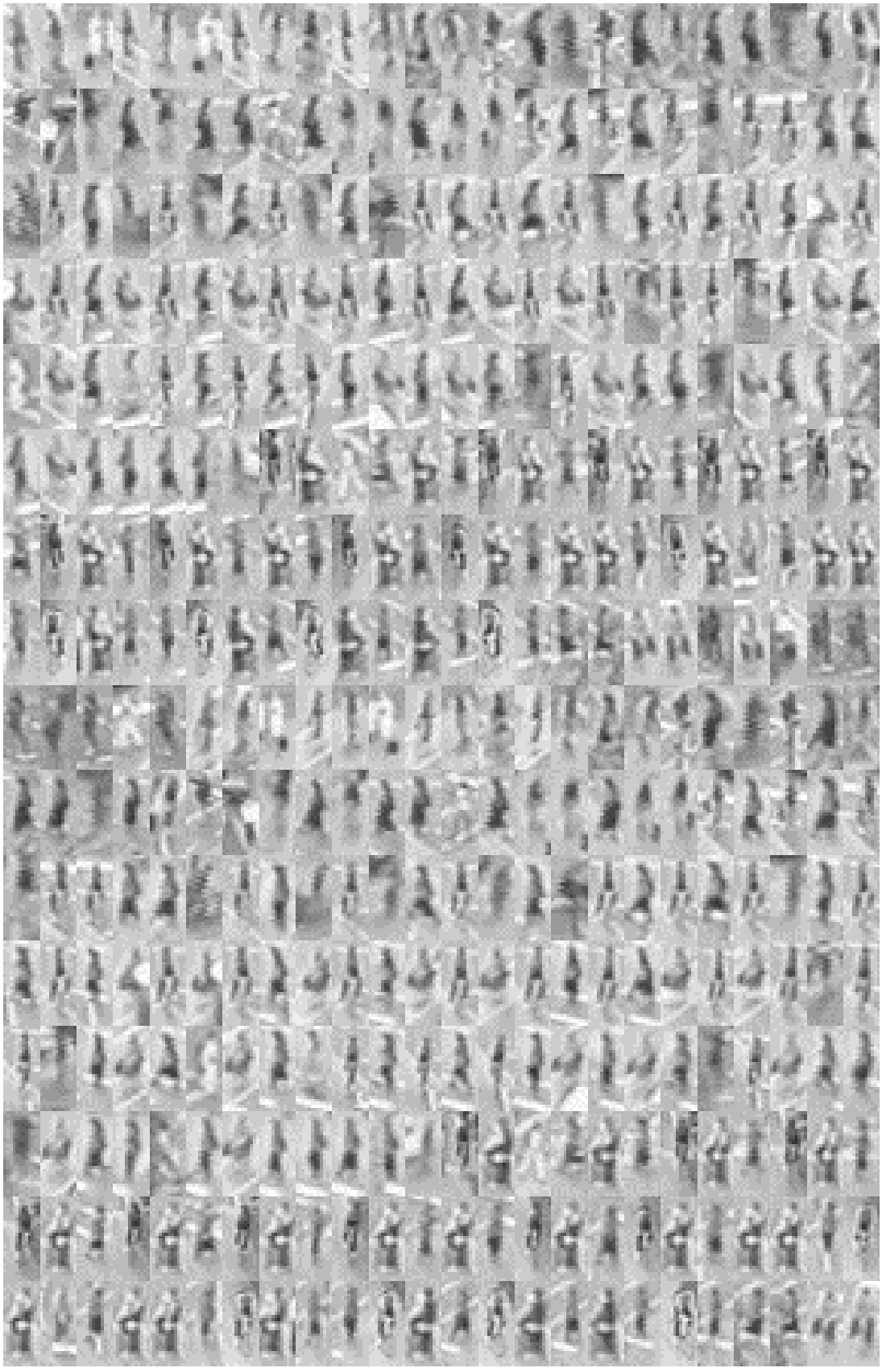
CUHK

MONASH

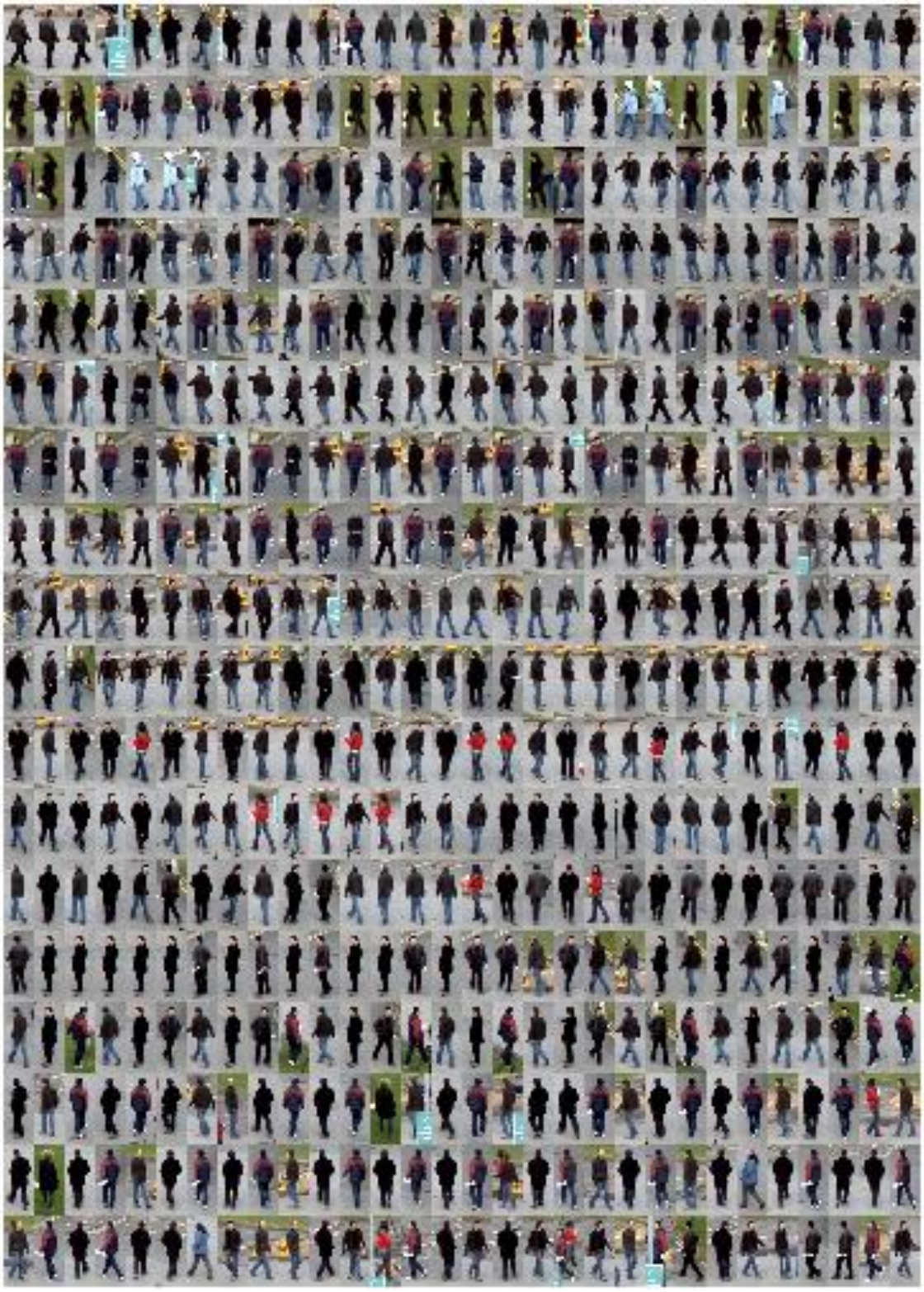
QMUL-R

QMUL-J

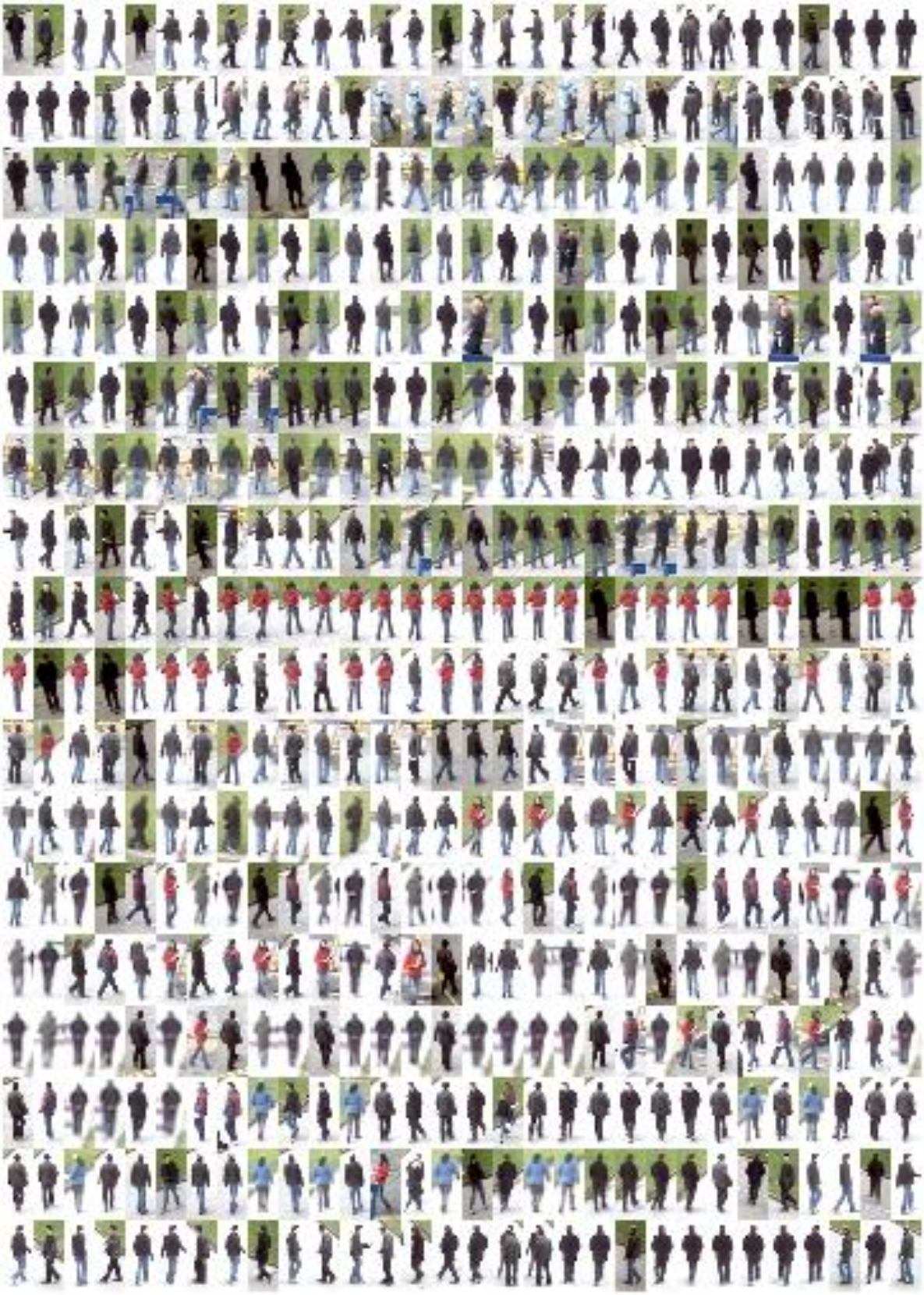
KWSI



PETS-01



PETS-02



PETS-03

PETS-04

Appendix B: Samples labelled by Oracle-2

This appendix displays a subset of approximately the first 250 instances from the samples labelled as pedestrians and non-pedestrians by Oracle-2, for all datasets, excluding the four PETS datasets. In the cases where fewer samples are displayed, the displayed samples represent the whole sample set and not a subset. The detection responses that were fed to Oracle-2 were obtained with HOG.

MIT

Examples of pedestrian instances

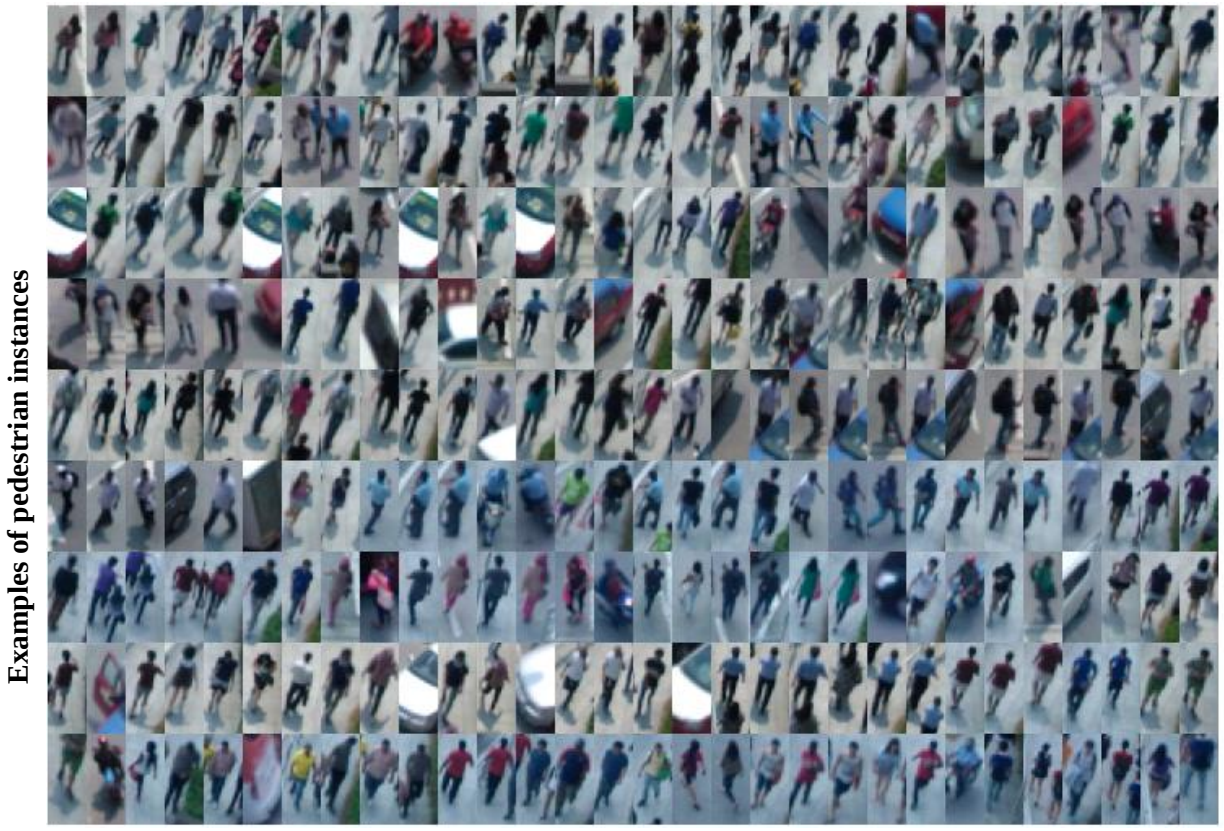
Examples of non-pedestrian instances

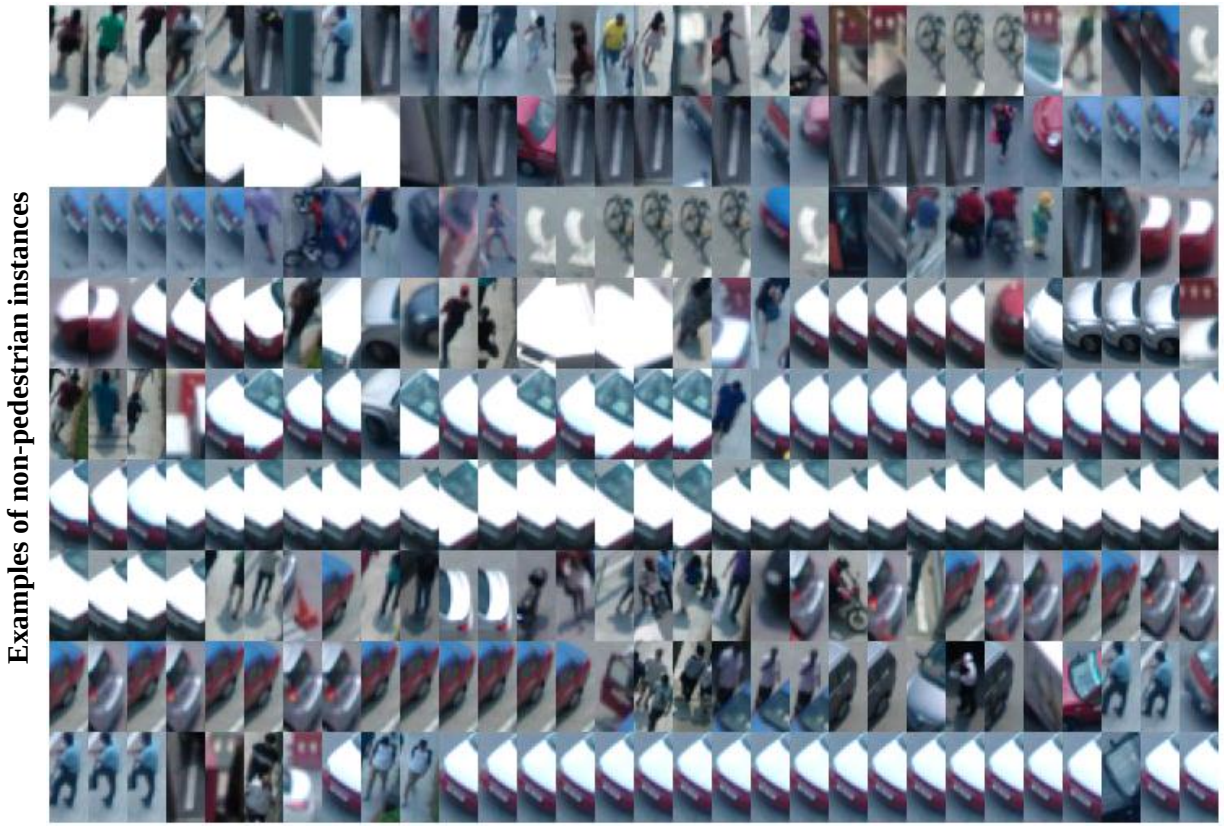
CUHK

Examples of pedestrian instances

Examples of non-pedestrian instances

MONASH

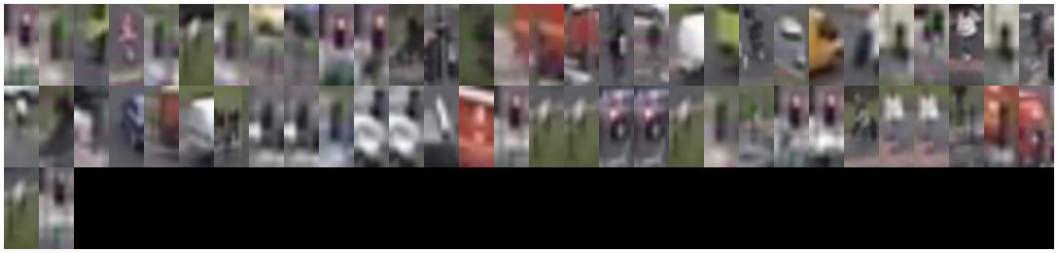

Examples of pedestrian instances


Examples of non-pedestrian instances

QMUL-R

Examples of pedestrian instances

Examples of non-pedestrian instances

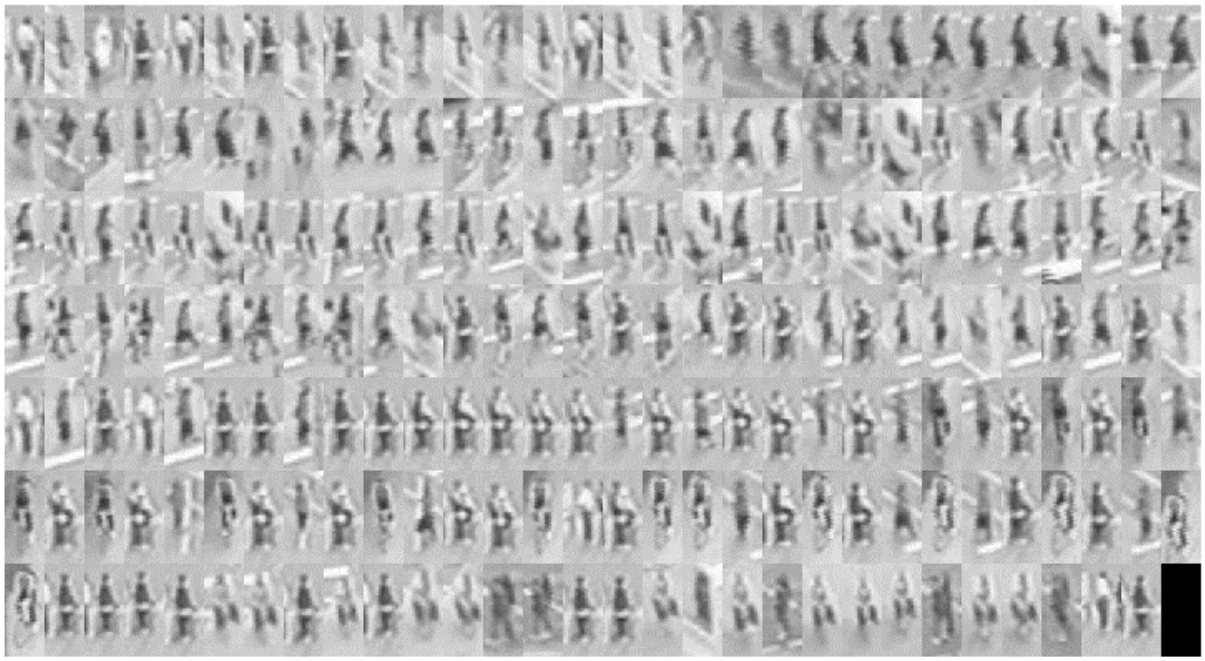


QMUL-J

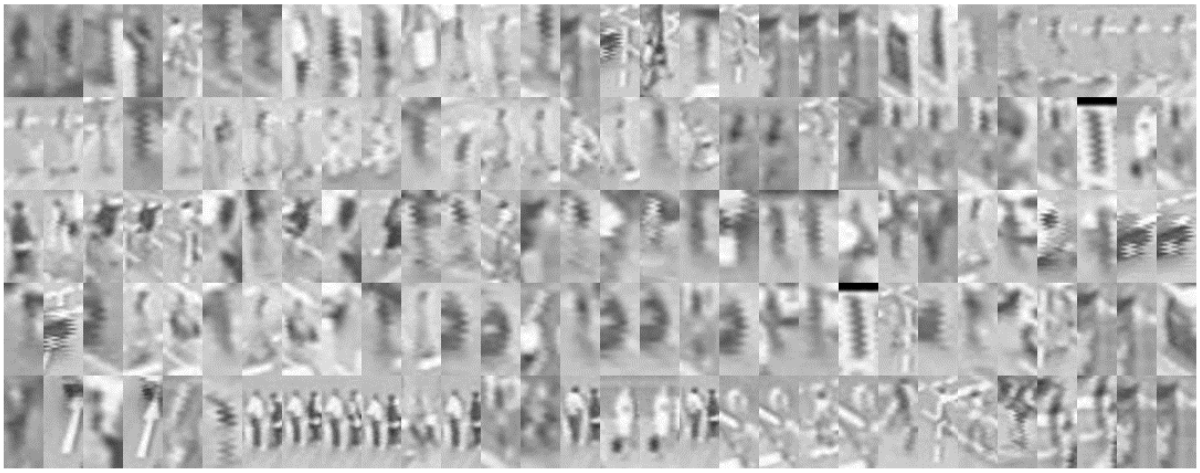

Examples of pedestrian instances


Examples of non-pedestrian instances

Examples of pedestrian instances

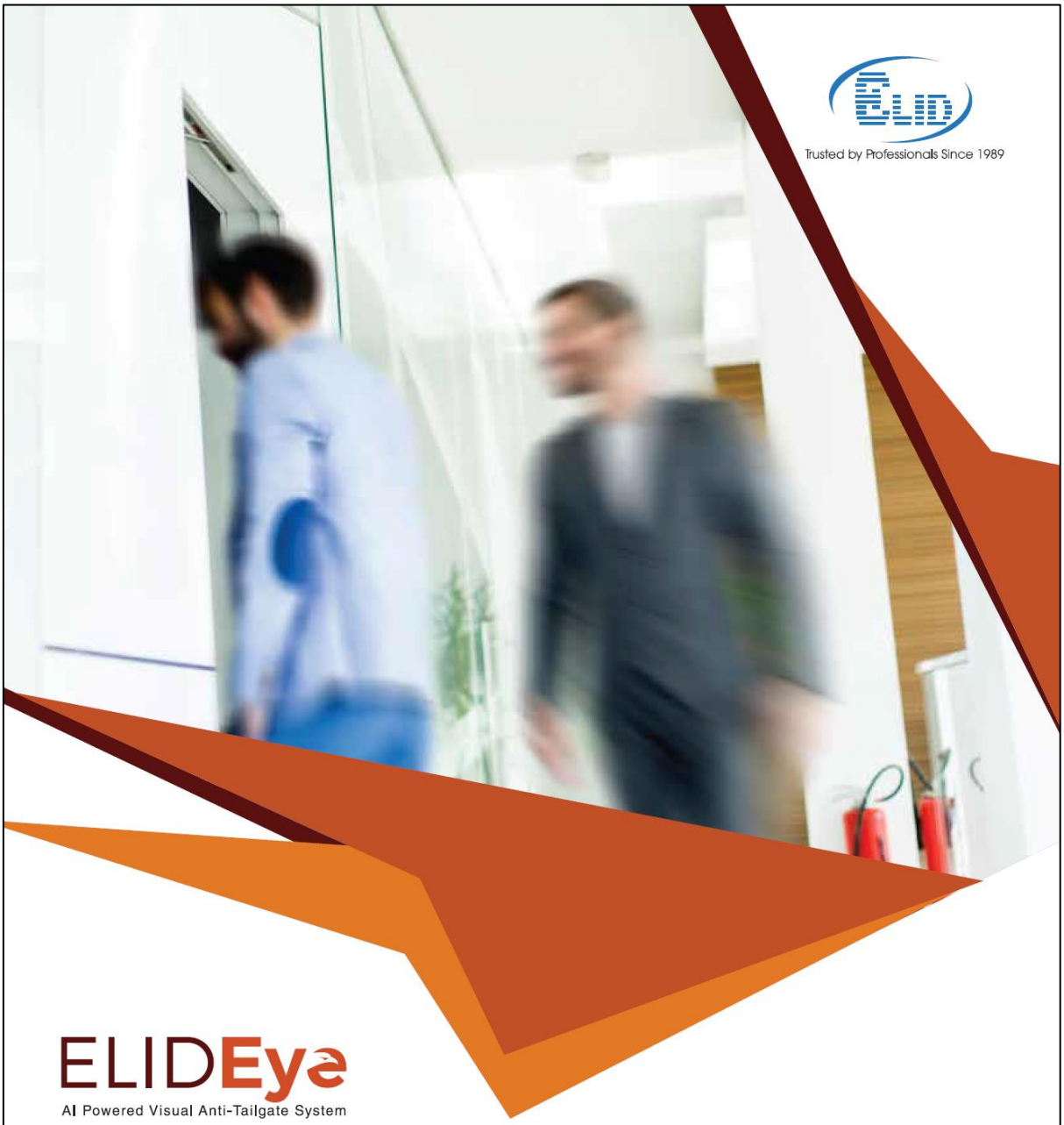


Examples of non-pedestrian instances



Appendix C: ELIDEye EV-100 Brochure

Page – 1 of Brochure



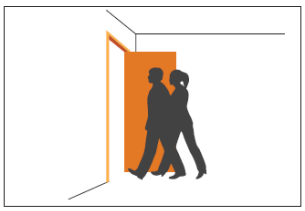
How it works

Once installed, ElidEye autonomously trains and adapts itself, learning detailed information about the environment and appearance of people within the field of view. This is done by applying state-of-the-art computer vision and machine learning algorithms to the video feed acquired by the built-in camera module. Subsequently, ElidEye detects and tracks all people within the field of view, thereby determining whether a person is entering or exiting the secure area. It checks the number of entering people against the number of valid entries and permits only a single authorized person to enter between a door open-door close cycle. If the aforementioned rule is violated, the system triggers an alarm. Equipped with a powerful quad-core processor, ElidEye is able to perform all these complex operations and precisely detect tailgating, piggybacking and crossing, in real-time.

There are three types of tailgate offences

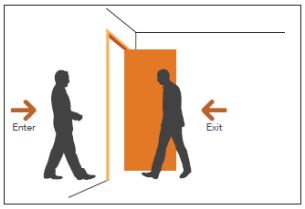
Tailgating

Act of an unauthorized person follows an authorized person to a restricted area without the consent of the authorized person.



Piggybacking

Happens when a person tags along with another person who is authorized to gain entry into a restricted area, or pass a certain checkpoint



Crossing

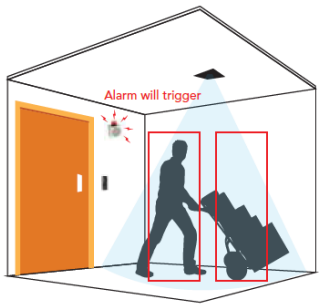
When an authorized person badges at the exit reader to leave a secure area, with or without his/her consent, an unauthorized person takes the opportunity to enter the secure area before the door closes.

Features & Benefits

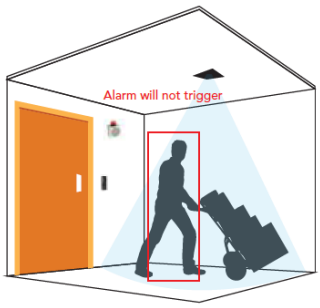
- **Tailgate Detection:** Protection against various security breach scenarios such as tailgating, piggybacking and crossing
- **Enhanced Security :** Provides a high level of security by ensuring only a single person can enter each time an authorized personnel's card is swiped/badged
- **Affordable :** AI-powered rather than using expensive mantrap vestibules, turnstiles or electronic sensor beams
- **Non-intrusive:** Mounted on the ceiling. Employees are unaware of its existence except when an alarm is triggered in response to a tailgating offence
- **Easy Installation and Compatibility:** Effortless integration with most access controllers, requiring only the valid entry and door sensor signals
- **Enforces proper card/fingerprint usage :** Encourages employees to adhere to proper access control protocols
- **Low maintenance costs:** Unlike traditional anti-tailgate systems, there is no hardware that is subject to wear and tear and may require repair/maintenance
- **Stand-alone system:** Operates with full autonomy. No human interaction/intervention required. No control or monitoring is required from a PC

Highlight #1

ElidEye is able to differentiate humans from other moving objects. Therefore, when an employee walks in with a trolley, stroller or luggage, it will only detect the employee. It will not falsely detect the moving trolley as another person.



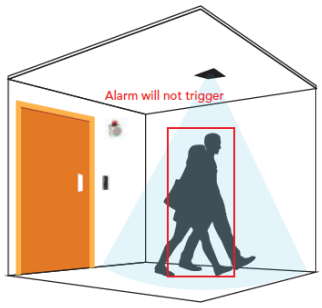
Standard Anti-Tailgate System



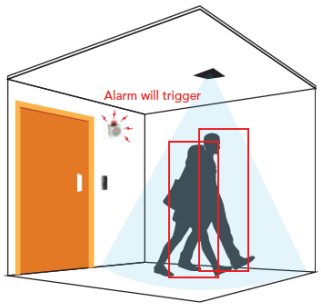
ElidEye

Highlight #2

During piggybacking, two or more people move together closely. It is difficult to isolate and detect the exact number of people during piggybacking. ElidEye has superior ability to detect precise locations of people even if they are close to each other. This feature enables ElidEye to effectively detect piggybacking.



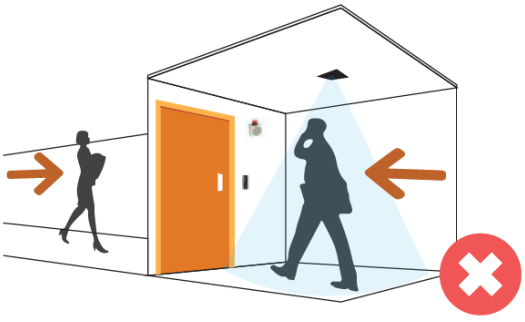
Standard Anti-Tailgate System



ElidEye

Highlight #3

ElidEye is able to track the movement of people. Therefore, it can differentiate between entering and exiting people. It is suited for detecting crossing. Alarm will not trigger if one person enters and another person exits the secure area at the same time.



Standard Anti-Tailgate System does not allow one person enters and another person exits at the same time

ElidEye allows one person enters and another person exits at the same time

Response options

Annunciator – An alarm may be triggered such as a bell, buzzer or a pre-recorded voice clip
Image Capture – Take snapshots of the tailgating offence
Video Recording – Record a short clip of the tailgating offence

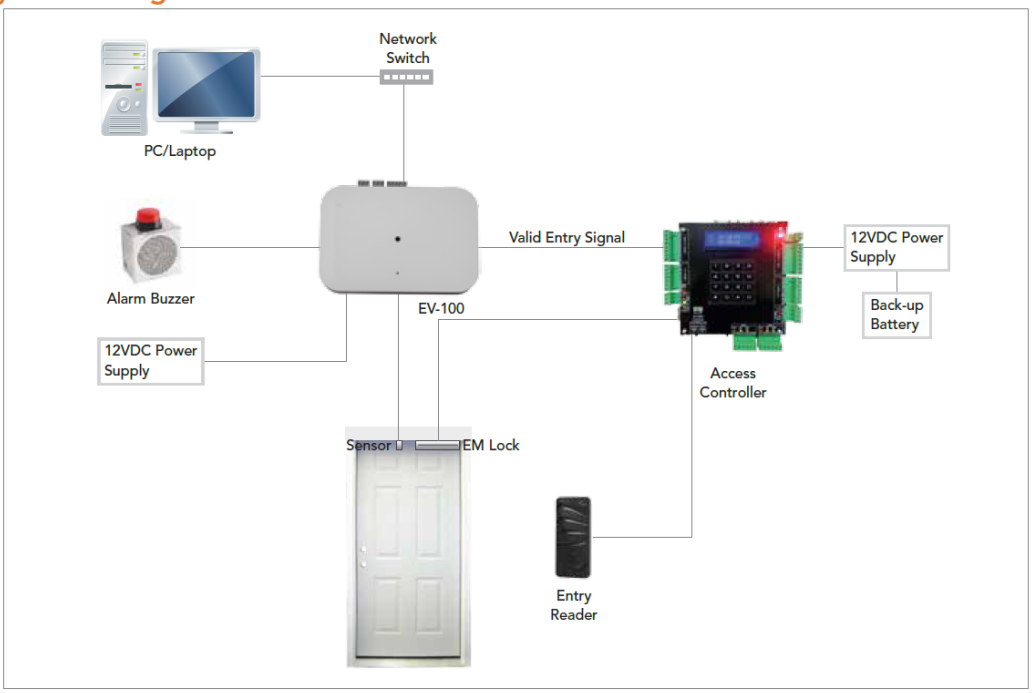
Application scenarios

- High-security offices, institutes and organizations
- Bank server rooms
- Casinos
- Government or military infrastructure
- University laboratories (chemical & biomedical)
- Public facilities like gyms

Specifications

Power	Standard 12V DC
Inputs	Door sensor – recommended voltage is 5V or 12V
	Valid Entry – recommended voltage is 5V or 12V
Output	Annunciator – recommended voltage is 12V, can drive up to 24V
Connectivity	Ethernet / WIFI
Environment	Designed for indoor operation. Outdoor operation may be sub-optimal
Compatibility	Any controller that can provide valid entry signal
Dimensions	200 cm (L) x 135 cm (W) x 40 cm (H)

System configuration



For more information: Check out the website at www.elid.com, or contact our dealers. ELID has a policy of continuous research and development , and reserves the right to change specifications without notice.

www.elid.com

Appendix D: Additional images of ELIDEye EV-100


ELIDEye
AI Powered Visual Anti-Tailgate System
GND IN1 IN2 IN3 IN4
NO1 COM1 NC1 OUTPUT 1
NO2 COM2 NC2 OUTPUT 2
12V GND POWER
USB 1&2
LAN


ELIDEye
AI Powered Visual Anti-Tailgate System