# High-performance Object Detection and Tracking Using Deep Learning

XINYU WANG


Principal Supervisor: Prof. Chunhua Shen
Co-Supervisor: Dr. Lingqiao Liu

A thesis submitted in fulfilment of
the requirements for the degree of
Master of Philosophy

School of Computer Science
Faculty of Engineering, Computer & Mathmatical Sciences
The University of Adelaide
Australia

14 October 2019

# Contents

# Abstract

Human detection and tracking are two fundamental problems in computer vision, which have been cornerstones for many real-world applications such as video surveillance, intelligent transportation systems and autonomous driving. Benefiting from deep learning technologies such as convolutional neural networks, modern object detectors and trackers have been achieving much improved accuracy on public benchmarks. In this work, we aim to improve deep learning based human detection. Our main idea is to exploit semantic context information for human detection by using deeply learned semantic features provided by semantic segmentation masks. These segmentation masks play as an attention mechanism and enforce the detectors to focus on the image regions where potential object candidates are likely to appear. Furthermore, after reviewing some widely used detection benchmarks, we found that the annotation quality for small and crowd objects does not meet to a satisfied standard. Hence, we introduce a new dataset which includes more than 8000 images for detecting small and crowd targets in fixed angle videos. Meanwhile, a baseline detector was proposed to exploit motion channel features for boosting the detection performance. The experimental results show that our proposed approach significantly improve the detection accuracy for the baseline detectors.

In addition to a novel method for object tracking, we propose to transfer the deep feature which is learned originally for image classification to the visual tracking domain. The domain adaptation is achieved via some "grafted" auxiliary networks which are trained by regressing the object location in tracking frames. Moreover, the adaptation is also naturally used for introducing the objectness concept into visual tracking. This removes a long-standing target ambiguity in visual tracking tasks and we illustrate the empirical superiority of the more well-defined task. We also experimentally demonstrate the effectiveness of our proposed tracker on two widely used benchmarks.

# Declaration

I certify that this work contains no material which has been accepted for the award of any other degree or diploma in my name, in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. In addition, I certify that no part of this work will, in the future, be used in a submission in my name, for any other degree or diploma in any university or other tertiary institution without the prior approval of the University of Adelaide and where applicable, any partner institution responsible for the joint-award of this degree.

I acknowledge that copyright of published works contained within this thesis resides with the copyright holder(s) of those works.

I also give permission for the digital version of my thesis to be made available on the web, via the University's digital research repository, the Library Search and also through web search engines, unless permission has been granted by the University to restrict access for a period of time.

## Acknowledgements

First and foremost, I would like to thank my principle supervisor, Prof. Chunhua Shen, for all his patient guidance, encouragement and advice. I have been extremely lucky to have a supervisor who cared so much about my work, and who responded to my questions and queries so promptly. His dedication to research inspired me a lot, and will definitely influence me throughout my research career in the coming years.

As a master by research student, I spent most of my time with Ph.D students and post doctoral researchers: Wei Yin, Bohan Zhuang, Zhi Tian, Tong He, Hao Chen, Hu Wang, Libo Sun, Yang Zhao, Yifan Liu, Yuliang Liu and Hui Li. I would like to thank for their help in both research and life.

Finally, I would like to thank my parents for their love and support, without whom I would never have enjoyed so many opportunities.

## Publications

The following peer-reviewed journal papers contain preliminary reports of the findings in this thesis (* indicates co-first author):

1. **Xinyu Wang**, Chunhua Shen, Hanxi Li and Shugong Xu, "Human Detection Aided by Deeply Learned Semantic Masks," Accepted to *IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT)*, 2019. DOI: 10.1109/TCSVT.2019.2924912 (Presented in Chapter 2.)
2. **Xinyu Wang**, Chunhua Shen, "Detecting small humans and vehicles in fixed camera angle videos," *under review*. (Presented in Chapter 3.)
3. Hanxi Li*, **Xinyu Wang***, Fumin Shen, Yi Li, Fatih Porikli and Mingwen Wang, "Real-time Deep Tracking via Corrective Domain Adaptation," Accepted to *IEEE Transactions on Circuits and Systems for Video Technology (T-CSVT)*, 2019. DOI: 10.1109/TCSVT.2019.2923639 (Presented in Chapter 4.)

# List of Figures

CHAPTER 1

# Introduction

---

In recent decades, as machines become increasingly powerful, machine intelligence has achieved great success in many real-world applications, such as face recognition system, machine translator, self-driving vehicle, safety monitoring and AlphaGo, all of these applications are making the artificial intelligence indispensable to our daily life. Benefiting from the development of mobile device, social media and high speed cellular network, there is an ever-increasing number of image data in the world, which makes it less and less possible for human-beings to manage all this data manually. Therefore, designing computer systems to automatically process and understand the large amount of data becomes a natural idea. However, it is commonly admitted that computers are accomplished in the tasks which can be defined by formal and mathematical rules, like calculating, storing and searching. But it is challenge for machines to solve the problems which are intuitive and abstract, such as recognising images. This is caused by the so-called semantic gap between human and machine, *i.e.*, image files are stored in the formulation of low-level pixel data on machines, but high-level semantic information is required for image analysis. Computer vision attempts to narrow this gap and teach the machines to understand pictures.

## 1.1 Object Detection

Object detection is a fundamental problem in computer vision, and it has been a cornerstone for many real-world applications. There are mainly two steps in a modern object detection system, localising a set of object candidates and classifying these targets into a certain category. In past decade, as the surge of deep learning, Convolutional Neural Networks

(CNN) [10, 11, 12, 13] have become the de-facto standard for solving this task, and a large number of CNN-based detectors have been proposed [1, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23]. Moreover, modern object detectors can be categorised into two types: one-stage approaches and two-stage approaches.

**Two-stage Detectors**, such as R-CNN [14], Fast R-CNN [15], Faster R-CNN [16] and Mask R-CNN [1], divide object detection task into two stages: extracting Region of Interest (RoIs) and classifying RoIs into foreground/background.

In [24], the authors proposed a Selective Search method to generate a set of candidate proposals which contain objects of all categories while filtering out most negative locations at the first stage, and then use SIFT [25] descriptors as feature representations to train SVM classifiers which classify the proposals into different categories at the second stage. R-CNN [14] simply replace the SIFT descriptors with convolutional features, which achieved significant improvements on detection accuracy. In more recent time, improved version of R-CNN have been proposed, such as Fast R-CNN [15], Faster R-CNN [16] and Mask R-CNN [1]. Fast R-CNN and Faster R-CNN revisited the feature extracting process in R-CNN, and proposed more efficient feature extracting strategies which allow the region proposal network to share the same backbone network with bounding-box regressors, such approaches significantly improved two-stage based detectors in both accuracy and speed.

However, it is admitted that, RPN-based detectors introduce excessively many hyper parameters, such as anchor sizes, anchor stride and anchor aspect ratios, which need to be carefully tuned in different datasets (especially for small targets) to achieve satisfied accuracy.

**One-stage Detectors**, such as YOLO [18, 20, 26], SSD [17, 19] and Retina Net [21], abandon the RoIs generation process in two-stage based detection framework, and directly regress and classify a set of pre-defined candidate anchor boxes.

YOLO [18] simply divides the input image into an $S \times S$ grid, and simultaneously predict bounding-boxes and categories in those boxes, which achieved very fast inference speed. Similarly, SSD [17] pre-defines a set of so-called “default boxes", and use deep features from different levels of convolutional layers to regress and classify these “default boxes".

Retina-Net [21] designed Focal Loss to solve the imbalance between positive and negative samples, but still relies heavily on the anchor boxes. These approaches [17, 18] have been tuned for very fast inference speed but their detection performance trails that of most two stage based detectors. Meanwhile, since the SSD regresses small targets on shallower convolutional layers, it has been blamed for worse accuracy on tiny objects. Also, as each grid cell in YOLO only predicts two boxes and can only have one class, it cannot perform well on small/crowded objects.

More recently, many anchor free based single-stage detectors have been proposed [23, 27, 28, 29, 30, 31]. CornerNet [27] detects an object bounding box as a pair of keypoints (top-left and bottom-right corners), however, complicated post-processing stage is required to group the pairs of corners belonging to the same instance. FCOS [23] formulates the object detection task into a per image pixel prediction fashion and achieves promising performance on public dataset. ExtremeNet [30] detects four extreme points (top-most, left-most, bottom-most, right-most) and center point of objects using a keypoint estimation network, then the five keypoints are grouped into a bounding box via geometrical rules.

## 1.2 Object Tracking

Online visual tracking aims to track the specific object labeled at the first frame of the video sequence, and is a popular research topic among the vision community.

In past decades, before the rise of deep learning, traditional tracking algorithms pay most attention to develop robust and powerful appearance model from the perspectives of hand-crafted features, model updating strategy, ensemble post-processor and observation model, some of them achieved great success in both accuracy and speed [8, 32, 33, 34, 35, 36, 37, 38, 39].

Over the last five years, convolutional neural networks [10, 11, 12, 13] achieved surprisingly success thanks to their ability in automatic feature extraction, experts no longer need to spend time on designing different manual-crafted features. According to some previous works, it has

been proved that the appearance model plays the most important role in a robust visual tracking system [40, 41]. Therefore, simply replace the hand-crafted features by deep convolutional features becomes a naive idea in some early deep learning based tracking algorithms. [42] is a well-known pioneering work that learns deep features for visual object tracking task. [43, 44] learn a deep model offline with a large number of images while updating it online for the current video sequence. [7, 45, 46, 47] extract hierarchical convolutional features from different level of deep neural network, then put the features into correlation filters to regress the respond map. These methods can be considered a combination between deep learning and the fast shallow tracker based on correlation filters. Recently, more and more state-of-the-art deep trackers adopt end-to-end training and testing fashion [5, 6, 9, 48, 49]. [9] proposed to pre-train deep CNNs in multi domains, with each domain corresponding to one training video sequence. The authors claim that there exist some common properties that are desirable for target representations in all domains such as illumination changes and motions. To extract these common features, the authors separate domain-independent information from domain-specific layers. The yielded tracker achieves excellent tracking performance while the tracking speed is only 1 fps. [5] learned a deep regressor that can predict the location of the current object based on its appearance in the last frame. The tracker obtains a much faster tracking speed (over 100 fps) comparing to conventional deep trackers. However, there is still a clear performance gap between [5] and the state-of-the-art deep trackers.

## 1.3 Overview of Contributions

Our work involves two important topics in the vision community, *i.e.*, object detection and tracking.

Here, we propose a simple yet effective approach to employ the segmentation masks as an external channel to provide extra semantic context for human detectors. Our experiment results show that the proposed method outperforms baseline detectors which use RGB channels alone. In other words, we aim to exploit the high-level semantic context provided by segmentation

masks, and use them to guide human detectors to learn much more discriminative features for detecting the humans from background. Our main contributes are as follows:

1. Firstly, we integrate extra segmentation features with RGB images for training the proposed human detector. We show that the extra features significantly improve detection performance on the COCO Persons dataset [4] and the CrowdHuman dataset [50].
2. To further explore the effectiveness of the external semantic context, we implement our proposed method with two popular detection frameworks, namely, Faster R-CNN and SSD, and train the detectors with segmentation masks of different levels of quality. Both of our binary models and scored models achieve significant improvements on the two datasets.
3. Moreover, we compare and analyze the learned discriminative features with the original features to gain insights on how the external semantic features improve the detection performance.
4. Finally, we propose two metrics termed NoZ and NoL to evaluate and compare the quality of learned deep features in a quantitative fashion, and find that more discriminative features can be learned by the proposed method compared to those existing ones.

Moreover, to evaluate the detectors ability in detecting small targets, we introduce a new dataset named SHV, which is designed for fixed angle video surveillance systems, two main categories of objects are annotated, *i.e.*, humans and vehicles. Compared to existing datasets, a large amount of tiny humans and vehicles are annotated. In addition, the average number of annotated objects is approximately $\sim$27 per frame in our proposed dataset, which has a much higher density than existing ones. Accordingly, the proposed dataset can be considered as a benchmark for evaluating the performance of detectors for tiny/crowded targets. In summary, contributions of our work on small object detection comprise the following:

1. We propose a new dataset for detecting vehicles and humans in fixed camera angle videos, most objects are of very small size. The training set includes 8881 image

frames while testing set includes 3600 image frames. The annotations of training set will be public, and an online benchmark will be setup.

2. We evaluate many state-of-the-art object detectors including both one-stage and two-stage based approaches on our proposed dataset to measure their ability in detecting tiny/crowded objects.
3. We propose a simple yet effective baseline network to exploit different motion patterns for convolutional neural networks, these motion patterns enable the network to notice tiny changes between adjacent video frames, which significantly improve the detection precision on small targets.

Finally, we propose a simple yet effective domain adaptation algorithm for visual object tracker. The equipped tracking algorithm, termed Corrective Domain Adaptation (CODA), transfers the features from the classification domain to the tracking domain, where the individual objects, rather than the image categories, are used as the learning samples. Furthermore, the adaptation is also naturally used for introducing the objectness concept into visual tracking. This removes a long-standing target ambiguity in visual tracking tasks and we illustrate the empirical superiority of the more well-defined task. The main contribution of this thesis includes:

1. We propose a simple yet effective domain adaptation method for visual tracking. The adaptation not only leads to a real-time tracking speed, but also remains a high tracking accuracy which is comparable to the state-of-the-art trackers.
2. For a certain type of tracking target, we propose to use the CNN branches, which are originally trained to adapt the deep feature to the visual tracking domain, to correct the initial tracking boxes. Within a sophisticated inference framework, the tracking accuracy boosts dramatically.
3. From another perspective, the success of the corrective adaptation empirically proofs that a more well-defined tracking target, rather than a simple bounding-box, could benefit the tracking process significantly. In other words,this work offers an alternative to addressing the long-standing “ill-posed” problem in visual tracking.

## 1.4 Outline

This thesis will process as follow:

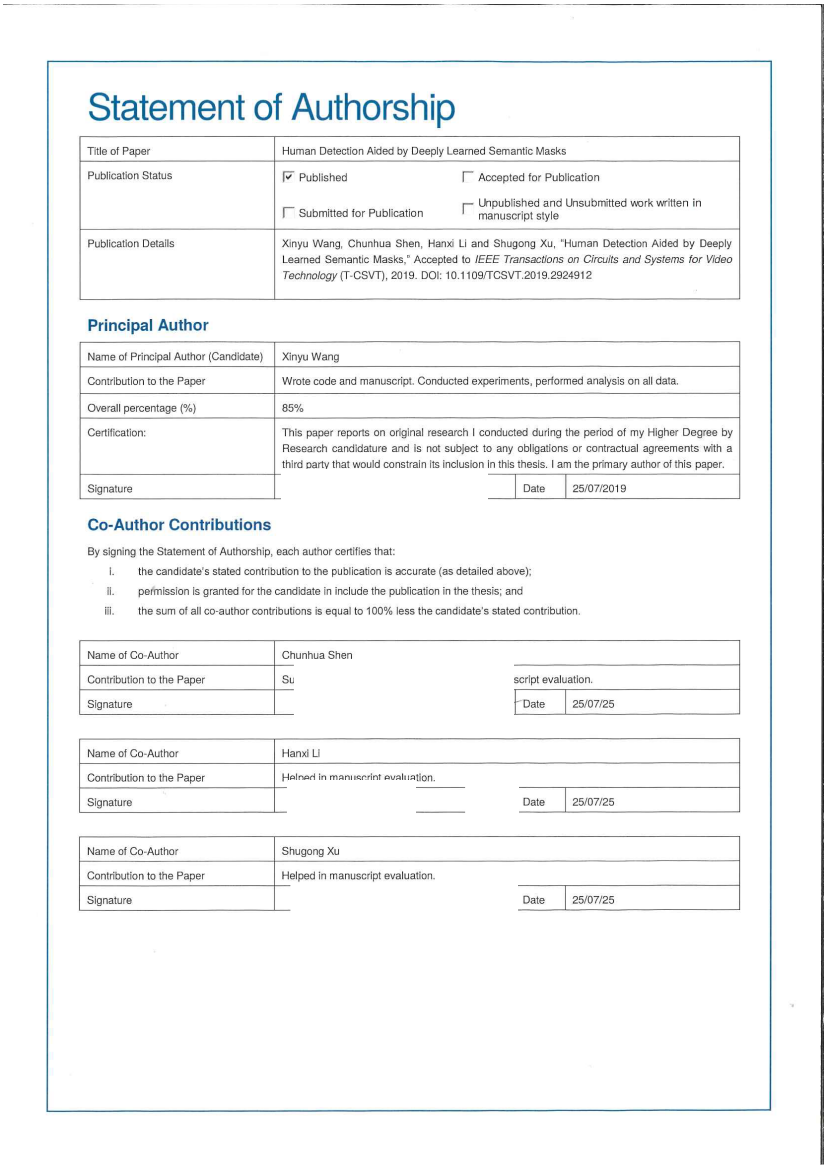
**Chapter 2: Human Detection Aided by Deeply Learned Semantic Masks.** In this chapter, we firstly review the literature combining extra features with the RGB images in computer vision tasks. Then, we will show that the use of semantic segmentation masks guide the CNN to learn more discriminative representations, and therefore significantly improve the detection performance. In the experiment, we will show the comparison of detection accuracy between our proposed methods and the state-of-the-arts on two widely used benchmarks.

**Chapter 3: Detecting Small Humans and Vehicles in Fixed Camera Angle Videos.** In this chapter, we review some widely-used detection benchmarks, and found that existing datasets only provide very weak annotations for small objects, which is therefore cannot be used for evaluating the detection performance on tiny targets. Hence, we propose a new dataset named SHV which includes more than 8000 images with manually annotated labels, and most of objects are of very small scale. Moreover, we also propose a baseline detector which exploits motion channel features for detecting small humans and vehicles in video sequences.

**Chapter 4: Real-time Deep Tracking via Corrective Domain Adaptation.** In this chapter, we will present a semi-deep learning based visual object tracker. Specifically, we first propose a domain adaptation approach for transferring the deep feature which is learned originally for image classification to the visual tracking domain. Then, we introduce objectness concept into visual tracking, which removes a long-standing target ambiguity in visual tracking tasks. In the experiment, we will show the comparison of tracking performance between our proposed tracker and the state-of-the-art methods on two widely used benchmarks under different evaluation metrics.

CHAPTER 2

# Human Detection Aided by Deeply Learned Semantic Masks



## Statement of Authorship

| | |
|---|---|
| Title of Paper | Human Detection Aided by Deeply Learned Semantic Masks |
| Publication Status | ☑ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style |
| Publication Details | Xinyu Wang, Chunhua Shen, Hanxi Li and Shugong Xu, "Human Detection Aided by Deeply Learned Semantic Masks," Accepted to *IEEE Transactions on Circuits and Systems for Video Technology* (T-CSVT), 2019. DOI: 10.1109/TCSVT.2019.2924912 |

### Principal Author

| | |
|---|---|
| Name of Principal Author (Candidate) | Xinyu Wang |
| Contribution to the Paper | Wrote code and manuscript. Conducted experiments, performed analysis on all data. |
| Overall percentage (%) | 85% |
| Certification: | This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper. |
| Signature | Date 25/07/2019 |

### Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);
ii. permission is granted for the candidate in include the publication in the thesis; and
iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

| | |
|---|---|
| Name of Co-Author | Chunhua Shen |
| Contribution to the Paper | Su script evaluation. |
| Signature | Date 25/07/25 |

| | |
|---|---|
| Name of Co-Author | Hanxi LI |
| Contribution to the Paper | Helped in manuscript evaluation. |
| Signature | Date 25/07/25 |

| | |
|---|---|
| Name of Co-Author | Shugong Xu |
| Contribution to the Paper | Helped in manuscript evaluation. |
| Signature | Date 25/07/25 |

FIGURE 2.1: Statement of Authorship for Paper "Human Detection Aided by Deeply Learned Semantic Masks"

## 2.1 Introduction

Detecting humans in images is a fundamental problem in computer vision, which aims to predict the bounding boxes of all the humans in an image. It has attracted a great deal of research interest in the computer vision community in recent years [51, 52, 53, 54, 55, 56]. Meanwhile, human detection has also been widely used in many real-world applications, such as video surveillance, robotics and self-driving vehicles.

During the last decade, benefiting from the power of deep Convolutional Neural Network (CNN), more and more CNN-based algorithms significantly outperformed those traditional methods on a wide variety of vision tasks [1, 13, 15, 57, 58]. The region-based convolutional neural network (R-CNN) [14] achieved remarkable performance for generic object detection, thus many R-CNN based human detectors have been proposed recently [54, 55, 59]. However, different from general object detection, image patches of humans are less distinguishable from the background, caused by the intra-class variation of humans in clothing, illumination and occlusion, which is shown to considerably affect the detection performance [54, 60] adversely. In other words, a human discriminator may need to rely more on semantic context information in order to achieve good performance.

In the literature combining extra features has been considered as an effective approach to boost RGB image features in that external semantic information can be introduced. Gupta *et al.* [61] implement an integrated system to exploit rich features from RGB-D images for object detection and segmentation. Spinello *et al.* [62] develop an Histogram of Oriented Depths (HOD) to enhance a detector. Mao *et al.* [54] aggregate six different types of extra features to improve pedestrian detection performance. Chen *et al.* [63] employ the segmentation mask to extract discriminative features for person search tasks. Song *et al.* [64] propose a mask-guided attention model for person re-identification. Wan *et al.* [65] deploy a min-entropy latent model which is trained with object confidence map to minimize the localization randomness. Compared to the original RGB images, these features such as edges, gradients, heat maps, dense depth maps, optical flow, object confidence map and segmentation mask can provide

an extra source of information, and guide the CNN to learn more powerful representations, which is the key to improve detection performance.

Prior to the recent work of CNN-based methods, most detectors are built on low-level appearance features (*e.g.*, edges and gradient) and carefully hand-crafted features (*e.g.*, HOG [51] and SIFT [66]).

These features are often not sufficiently strong to achieve satisfactory accuracies, especially for low-resolution images [54]. In addition, in many applications, detectors can benefit from depth information. However, to acquire depth typically needs a depth sensor such as laser or depth cameras, which is not always available. Recently, a few studies have revealed the power of segmentation masks [54, 63, 64, 67]. Image segmentation aims to output a segmentation mask which assigns semantic labels to every pixel in an image [58, 68, 69], such segmentation masks carry extremely rich semantic information, and it can be a powerful tool to boost human detectors. Inspired by the success of these works, here we propose a simple yet effective approach to employ the segmentation masks as an external channel to provide extra semantic context for human detectors. Our experiment results show that the proposed method outperforms baseline detectors which use RGB channels alone. In other words, we aim to exploit the high-level semantic context provided by segmentation masks, and use them to guide human detectors to learn much more discriminative features for detecting the humans from background.

## 2.2 Related Work

We review some works that are most relevant to ours.

**Multiple Features Integrating** Extra features, like gradient, hand-crafted features, depth and semantic segmentation masks have been used as a source to provide extra semantics to boost the performance of convolutional neural networks on a wide variety of vision problems, such as visual object tracking [44], person search [63], person re-identification [64] and pedestrian detection [2, 54, 61].

There are mainly two approaches (see Fig. 2.2) to aggregate the extra features with the original RGB features. The first and mostly used method is to employ external convolution layers to learn the extra features, and then concatenate these two feature maps at a later stage. The work of [54] proposed a HyperLearner which learns the representations of channel features from the extra context, and concatenate the extra feature maps with the features extracted by the VGG [57] backbone network.

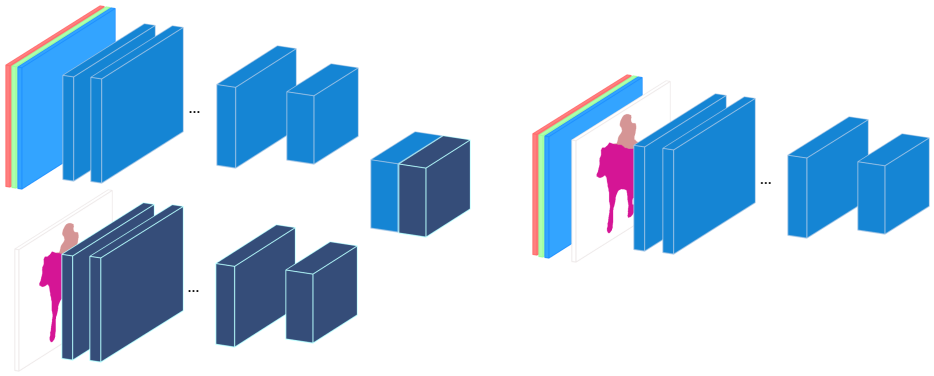


FIGURE 2.2: Two methods of integrating multiple information in a CNN. Left: Features are fused after several convolution operations. Convolutional kernels are learned separately. Right: Extra semantic context is directly concatenated to the original RGB channels. Note that the first approach may be viewed as a special case of the second approach. For the second approach, when the first a few convolutional layers employ group convolutions, it becomes the first case.

The authors of [63] and [64] employ the segmentation mask to separate the original RGB images into the foreground part and background part. Then several neural networks are trained to extract features from both the original RGB images and mask-guided cropped images. Finally, different feature maps are concatenated together before prediction. [70] proposes semantic segmentation infusion layer to encode semantic masks into shared feature maps. The work of [44] implements three sub-networks to learn discriminative feature representations from multiple cues which contain original RGB images and external gradient context, and all of the features extracted by these three sub-networks are fused to a feature vector classification. A drawback is that two or even three times of extra parameters have to be learned, introducing computation overhead. Besides, most of the above works design a novel neural network structure, which is more difficult to adapt to different detection frameworks, and it remains unclear if it is the optimal approach for original and extra features being learned separately

in different sub-networks. We conjecture that directly concatenating an extra channel of information with the original RGB channels and feeding them altogether into a CNN may be a simpler but effective approach in terms of exploiting extra information. Thus, only a few new parameters are learned to exploit the new semantic information.

### 2.2.1 Object Detection

**Faster R-CNN** Region-based Convolutional Neural Network (R-CNN) [14] has demonstrated the effectiveness of using region proposals with deep neural networks, and it achieves excellent performance for generic object detection. Many R-CNN based detectors have been proposed in recent years [1, 15, 16]. Faster R-CNN provides much faster and more flexible alternative to the original R-CNN, and becomes one of the most widely used detection frameworks in the vision community. Faster R-CNN consists of two stages. Firstly, a number of candidate object bounding boxes are generated by RPN. Secondly, the features of each RoI are extracted by the backbone network via RoIPool which was proposed in [15]. Then all of these features are fed into the regressors and classifiers for final prediction. For faster inference, features used by these two stages can be shared. In addition, we employ some recent technologies such as Batch Normalization (BN) [71] and Feature Pyramid Network (FPN) [72] to achieve better performance.

**Single Shot Multibox Detectors** Single shot detectors such as Single Shot Multi-Box Detector (SSD) [17] and YOLO [18] discard the region proposal module for simpler design, and allow single pipeline detection that directly predicts bounding boxes and category labels. In SSD, the output space of bounding boxes are discretized into a set of “Default Boxes" over different aspect ratios and scales for multiple convolutional layers, and each layer is enforced to focus on predicting objects of certain scale. Thus, for small and medium sized object prediction, SSD needs to use the features from shallow layers with small receptive fields, which may cause lower performance on small and medium objects due to the lack of semantic information. Therefore, feeding extra semantic context into these single-shot-based detectors might be useful to improve their performance.

### 2.2.2 Segmentation

**Semantic Segmentation** The purpose of semantic image segmentation is to predict a category label for every image pixel. Recently, convolutional networks are driving advances in semantic segmentation, and remarkable success has been achieved. Among these CNN-based methods, Fully Convolutional Network (FCN) [58] has become a popular choice. A FCN takes an input image of arbitrary size, and applies a series of convolutional layers. Then per-pixel likelihood score maps for all semantic categories are predicted by the network. Benefiting from the deep learning technologies, FCN provides an end-to-end solution for accurate semantic segmentation. DeepLab [68, 73], a FCN-based semantic segmentation method, achieves state-of-the-art performance in recent years. DeepLabv3+, the latest version of DeepLab framework, uses atrous convolution to control the resolution, and an encoder-decoder structure is deployed to further refine the segmentation results especially the pixels among object boundaries. Such a carefully-designed framework significantly improves the semantic segmentation performance.

**Instance Segmentation** Different from semantic segmentation, instance segmentation aims to identify individual instances of different semantic classes in an image. As the object appearance of the instances from the same category can be very similar, instance segmentation is therefore often regarded as a much more difficult problem compared to the traditional semantic segmentation task. Inspired by the semantic segmentation, FCN-based methods are also widely used and perform well in instance segmentation problem. For example, Li *et al.* [74] propose a Fully Convolutional Instance Segmentation (FCIS) framework. FCIS detects and segments the object instances jointly and simultaneously. He *et al.* [1] implement a Mask R-CNN framework, which combines R-CNN framework for bounding box detection and FCN framework for densely output tasks, and achieves state-of-the-art performance on multiple tasks including object detection, instance segmentation and human pose estimation.

## 2.3 Mask Guided Human Detection

### 2.3.1 Overview

In many recent works [1, 63, 64, 75, 76], it has been proved that results of image segmentation can be beneficial in terms of the performance of deep convolutional neural networks. However, the procedure of how to use the segmentation mask can be very different. In [1], a multiple-branch neural network is deployed to train the detection and segmentation tasks jointly, thus the shared backbone network can learn discriminative features from both tasks simultaneously. The experiment result shows that the object detector jointly trained with segmentation task can obtain slight improvement compared to the single task trained detector. In [75], the object detection task is formulated as a segmentation problem, then the initial object localization can be refined by the segmentation masks.

However, above methods may have a number of limitations. 1) Data Limitation: In [1], a joint training procedure is adopted to train multiple tasks, *e.g.* bounding box detection and instance segmentation. However, for most object detection datasets, pixel-wise annotations are not accessible. Thus such a training procedure can hardly be adapted to those datasets without ground-truth pixel labels. 2) Robustness: In [1], multiple task branches share the same backbone network to extract deep features. This strategy significantly improves the network efficiency. However, the features learned by different tasks might not be beneficial to the others' performance. For example, in [1], the detection performance dropped when the author jointly trained the detector with a human pose estimator.

In this paper, an image is first fed into the segmentation module, which outputs the *semantic segmentation mask*. For the segmentation module, we employ two off-the-shelf methods to generate segmentation masks DeepLabv3+ [68] for semantic segmentation mask and Mask R-CNN [1] for instance segmentation mask. Then both semantic segmentation mask and instance segmentation mask generated by the segmentation module are transferred into a binary segmentation mask.

*It is important to note that even the Mask R-CNN is able to generate instance masks, we only use the semantic masks* to verify the effectiveness of our approach.

Once the semantic segmentation masks have been generated, they are integrated with the original RGB images and fed into the detection module. In this paper, we implement our method on two popular generic object detection frameworks, *i.e.*, Faster R-CNN and SSD.

### 2.3.2 Segmentation Module

To explore the effectiveness of the input segmentation masks, we use multiple settings for both DeepLabv3+ and Mask R-CNN to generate segmentation masks of different quality.

For DeepLabv3+, we devise two types of segmentation masks termed *binary* semantic segmentation mask and *scored* semantic segmentation mask, which are denoted as $\mathcal{M}_b$ and $\mathcal{M}_s$ respectively. Both *binary* and *scored* semantic segmentation masks share the same backbone network, *i.e.*, Xception for feature extraction. However, $\mathcal{M}_b$ dropped the score information which $\mathcal{M}_s$ keeps.

The *binary* semantic segmentation mask $\mathcal{M}_b$ is defined as:

$$\mathcal{M}_b^j = f(\frac{e^{\mathcal{L}_j}}{\sum_{k=1}^{K} e^{\mathcal{L}_k}}) \qquad j = 1, ..., K \tag{2.1}$$

The *scored* semantic segmentation mask $\mathcal{M}_s$ is defined as:

$$\mathcal{M}_s^j = \frac{e^{\mathcal{L}_j}}{\sum_{k=1}^{K} e^{\mathcal{L}_k}} \qquad j = 1, ..., K \tag{2.2}$$

where $\mathcal{L}$ is the raw logits matrix generated by the DeepLabv3+ model. $\mathcal{M}^j$ is the $j^{th}$ element in the segmentation matrix, and $K$ is the number of matrix elements. We use $f(x)$ to transfer the raw segmentation mask to a single binary matrix, which is described in Eq. 2.4.

For Mask R-CNN, a few different backbone networks are employed to generate instance segmentation masks of different levels of quality. For example, we use ResNet-101 [13] and ResNext-152 [77] backbone networks here (more details can be found in Table 2.1). It is a

remarkable fact that the instance segmentation masks can be easily transferred into bounding boxes. If we keep the instance information, the human detectors would be easily to overfit the instance segmentation masks while training, which can led to a poor detection performance in testing without instance masks available. Therefore, instance segmentation masks are converted into a single binary segmentation mask $\mathcal{M}$. In other words, only semantic masks are used here during training and testing of our detector.

The definition of instance segmentation mask $\mathcal{M}$ writes:

$$\mathcal{M} = \sum_{i=1}^{n} f(m_i) \tag{2.3}$$

where $m_i$ is the segmentation mask for the $i^{th}$ instance in one image generated by Mask R-CNN. $n$ is the number of the instances in the image.

For fair comparison with the effectiveness of semantic segmentation masks, and reduce the computational complex, most of segmentation masks are transferred into a single binary segmentation mask for each image by $f(x)$, except the *scored* semantic mask, which keeps the score information. That is, each pixel indicates the probability of a certain category. The binary segmentation mask, which is similar to an attention mechanism, forces the human detectors to pay more attention to the regions which are highlighted by the semantic segmentation masks. Meanwhile, such binary segmentation masks can naturally separate one image into foreground part and background part, which can help the detectors to learn highly discriminative features for classifying the target objects and background.

The raw segmentation masks generated by the segmentation module are enforced into a binary mask by $f(x)$:

$$f(x) = \begin{cases} 0, & x_{ij} <= S_\tau \\ 1, & x_{ij} > S_\tau \end{cases} \tag{2.4}$$

where $S_\tau$ is a score threshold to filter prediction noises, and $x$ is the raw segmentation mask matrix. Each element $x_{ij}$ in the matrix indicates the probability of the 'Person' category.

In summary, the segmentation module is used to generate the raw segmentation masks, then both semantic segmentation mask and instance segmentation mask are transferred into a single binary segmentation mask which will be fed into the detection module.

### 2.3.3 Detection Module

To evaluate the generative ability of the proposed method, we implement our method on two widely-used detection frameworks, *i.e.* Faster R-CNN and SSD. During training, we replace the 3-channel RGB images to 4-channel RGBM images:

$$Input : \mathbb{R}^3_{RGB} \rightarrow \mathbb{R}^4_{RGBM} \tag{2.5}$$

where $\mathbb{R}^4_{RGBM}$ space is composed of one $\mathbb{R}^3_{RGB}$ space for original RGB channels and one segmentation mask space $\mathbb{R}^1_{\mathcal{M}}$ for extra semantic context.

In the Faster R-CNN framework, a set of rectangular object proposals are firstly generated by the RPN, which are Region of Interest (RoI). Then the features of RoI are extracted by the backbone network. Furthermore, the features are fed into a bounding box regressor and category classifier to predict the target localization and class label. These two tasks are trained jointly, thus the loss function of Faster R-CNN writes:

$$L_f = \frac{1}{N_{cls}} L_{conf} + \lambda \frac{1}{N_{reg}} L_{loc} \tag{2.6}$$

where $L_{conf}$ and $L_{loc}$ are log loss for binary classification and smooth $L_1$ loss for bounding box regression. $N_{cls}$ and $N_{reg}$ are the normalization parameters which are decided by mini-batch size and the number of proposals respectively. $\lambda$ is a term to balance the two losses.

In the SSD framework, a number of pre-defined 'Default Boxes' are generated for regressing the target bounding boxes. To accommodate the target objects in different scales and shapes, these generated 'Default Boxes' also vary in multiple aspect ratios and sizes. The classification

task and localization task are trained jointly, thus the loss function of SSD can be given as:

$$L_s = \frac{1}{N}(L_{conf} + \lambda L_{loc}) \tag{2.7}$$

where $L_{conf}$ and $L_{loc}$ are softmax loss for classification and smooth $L_1$ loss for localization respectively. $N$ is the number of positive default boxes that matched to the predicted boxes, and $\lambda$ is a constant weighting factor to keep a balance between these two losses.

In the detection module, RGB-M images are fed into the detectors, then object bounding boxes and category scores are predicted.

## 2.4 Experiments

In this section, firstly, we introduce the datasets and evaluation protocols that we use in this paper, followed by some implementation details. Then we show both quantitative results and qualitative results of our proposed method based on two popular detection frameworks, *i.e.* Faster R-CNN and SSD on both MS-COCO Persons and CrowdHuman datasets. Finally, the effectiveness of the segmentation masks are experimentally analyzed.

### 2.4.1 Datasets

It is notable that different from pedestrian detection tasks which mostly focus on outdoor scenes and whole body, and the target pedestrians usually have a fixed aspect ratio (*e.g* 0.41). Human detection task aims to predict the bounding boxes of all the people in an image, indoor or outdoor, partial or whole body. Thus, we evaluate the proposed method on two human detection benchmarks, *i.e.*, MS-COCO Persons [4] and the very recent CrowdHuman dataset [50].

The MS-COCO Persons dataset consists of 64k training images and 5k testing images. The CrowdHuman dataset consists of 15k training images and 4k testing images. In terms of density, on average there are ∼4.01 persons per image in COCO Persons dataset while ∼22.64 in CrowdHuman dataset. The annotations of CrowdHuman provide both visible part

bounding box and full part bounding box for the humans while the COCO Persons dataset only provides visible part. Thus we only use the visible bounding box for CrwodHuman while training. For segmentation, DeepLabv3+ [68] and Mask R-CNN [1] are employed to generate semantic segmentation masks. Both are trained on the MS-COCO dataset. It is noteworthy that the CrowdHuman dataset dose not provide segmentation annotations. Therefore the segmentation models trained on COCO dataset are directly used to generate segmentation masks for CrowdHuman experiments without any further fine-tuning.

### 2.4.2 Implementation Details

We evaluate our proposed method on both Faster R-CNN and SSD framework. Both Faster R-CNN detectors and SSD detectors use ResNet-50 [13] as the backbone network. For MS-COCO Persons, we initialize the models with ImageNet-pretrained model. It should be noted that our methods need to take a 4-channel RGB-M input which is incompatible with the original ImageNet-pretrained models. In [64], the authors had the same problem. They solved this problem by training from scratch. In this paper, we simply use a randomly initialized filter for the extra channel.

For Faster R-CNN based methods, we train the networks for 180k iterations on the MS-COCO Persons dataset, with the base learning rate set to 0.01 and decreased by a factor of 10 after 60k and 160k iterations. The Stochastic Gradient Descent (SGD) solver is adopted to optimize the network on 4 Nvidia K80 GPUs. A mini-batch involves 2 images per GPU. Weight decay and momentum are set to 0.0001 and 0.9 respectively. Then, for the CrowdHuman dataset, we simply fine-tune the MS-COCO Persons models for 80k iterations. The initial learning rate is set to 0.001 and decreased after 60k iterations. Other settings are identical with the MS-COCO Persons dataset.

For the SSD based detectors, we train the networks for 240k iterations on the MS-COCO Persons dataset, with the base learning rate set to 0.001 and decreased by a factor of 10 after 160k and 200k iterations. As the input image size is smaller than Faster R-CNN (input size

being $512 \times 512$ in this paper), the mini-batch of SSD detectors involves 8 images per GPU. Then, we also fine-tune the SSD MS-COCO Persons models on the CrowdHuman dataset.

| Name | Segmentation Module | Backbone | Binary Mask |
|---|---|---|---|
| Sem-X | DeepLabv3+ | Xception | N |
| Sem-X-B | DeepLabv3+ | Xception | Y |
| Ins-R101-B | Mask R-CNN | ResNet-101 | Y |
| Ins-R152-B | Mask R-CNN | ResNext-152 | Y |

| Detection Module | Backbone Network |
|---|---|
| SSD | ResNet-50 |
| Faster R-CNN | ResNet-50 |

TABLE 2.1: Experiment settings of the segmentation module and detection module.

### 2.4.3 Quantitative Results

For evaluation, we use the standard MS-COCO metrics including Average Precision (AP @ IoU=0.50:0.95), Average Recall (AR @ IoU=0.50:0.95) and $AP^S$, $AP^M$, $AP^L$, $AR^S$, $AR^M$, $AR^L$ which are AP and AR in different scales for comparison on both MS-COCO Persons dataset and CrowdHuman dataset.

Multiple settings of experiments are conducted (see Table 2.1). We use the ResNet-101 backbone network for Mask R-CNN to generate *lower quality* segmentation masks and Xception [78] backbone networks for DeepLabv3+ to generate *binary* segmentation masks (see Eq. 2.1 and Eq. 2.3). For *higher quality* instance models, we use ResNext-152 [77] as backbone network (see Eq. 2.3); For $scored$ semantic models, we keep the score information, $i.e.$ each pixel in the mask indicates the probability of 'Person' class (see Eq. 2.2). For the detection module, we use ResNet-50 as the backbone network for both Faster R-CNN and SSD detectors.

**MS-COCO Persons.** Table 2.2 and Table 2.3 show the performance of our method on MS-COCO Persons dataset by using Faster R-CNN framework and SSD framework respectively

For the Faster R-CNN framework, as shown in Table 2.2 we compare our method with several detectors. FCIS, MaskRCNN-Seg are jointly trained with instance segmentation tasks,

| Model Name | Input | Mask | AP | AR |
|---|---|---|---|---|
| DRFCN [79] | RGB | - | 0.475 | 0.536 |
| FCIS [80] | RGB | - | 0.510 | 0.564 |
| DeNet [81] | RGB | - | 0.519 | 0.628 |
| MaskRCNN [1] | RGB | - | 0.525 | - |
| MaskRCNN-Seg [1] | RGB | - | 0.536 | - |
| G-RMI[82] | RGB | - | 0.539 | 0.649 |
| Ours-FRCNN-Baseline | RGB | - | 0.534 | 0.622 |
| Sem-X-B | RGB-M | DeepLabv3+ | 0.548 | 0.626 |
| Sem-X | RGB-M | DeepLabv3+ | 0.552 | 0.629 |
| Ins-R101-B | RGB-M | Mask R-CNN | 0.547 | 0.624 |
| Ins-R152-B | RGB-M | Mask R-CNN | 0.567 | 0.643 |
| Ours-Upbound | RGB-M | Ground Truth | 0.756 | 0.792 |

TABLE 2.2: Faster R-CNN framework detection performance comparison with the baseline detector and our proposed mask-guided detectors on MS-COCO persons.

| Model Name | Input | Mask | AP | AR |
|---|---|---|---|---|
| Ours-SSD-baseline | RGB | - | 0.436 | 0.528 |
| Sem-X-B | RGB-M | DeepLabv3+ | 0.448 | 0.532 |
| Ins-R101-B | RGB-M | Mask R-CNN | 0.442 | 0.515 |
| Model Name | Input | Mask | $AP^s$ | $AP^m$ |
| R50-SSD | RGB | - | 0.147 | 0.533 |
| Sem-X-B | RGB-M | DeepLabv3+ | 0.178 | 0.550 |
| Ins-R101-B | RGB-M | Mask R-CNN | 0.210 | 0.544 |

TABLE 2.3: SSD framework detection performance comparison with the baseline detector and our proposed mask-guided detectors on MS-COCO persons.

others only train a bounding box detector. As shown in Table 2.2, our two *binary* models obtain around 2% improvement compared to the single task trained Mask R-CNN and 1% improvement compared to the jointly trained Mask R-CNN.

In addition, the *R152* instance model achieves a significant improvement compared to both Mask R-CNN and baseline detector, which are 4% and 3% respectively. Meanwhile, it is noteworthy that models trained with mask of higher quality perform better than the models trained with mask of lower quality.

We also apply our proposed method with the SSD framework. As the aim of this paper is not to achieve the highest performance on the dataset, we only compare the *binary* models for the SSD framework to save computational resources. As shown in Table 2.3, the proposed method also achieves better performance on MS-COCO Persons dataset. Moreover, as each convolution layer in SSD is enforced to focus on predicting objects of certain scale, thus the features employed to predict small objects are extracted from very shallow layers, which contain only a few semantic context.

Therefore, we further explore whether the extra semantic context can improve the SSD ability of detecting small humans. Table 2.3 shows the average precision $AP^s$ of small targets (area $< 32^2$) and $AP^m$ of medium targets ($32^2 <$ area $< 96^2$) on MS-COCO Persons. We can see that the segmentation mask can indeed considerably improve the SSD detection performance on small and medium objects.

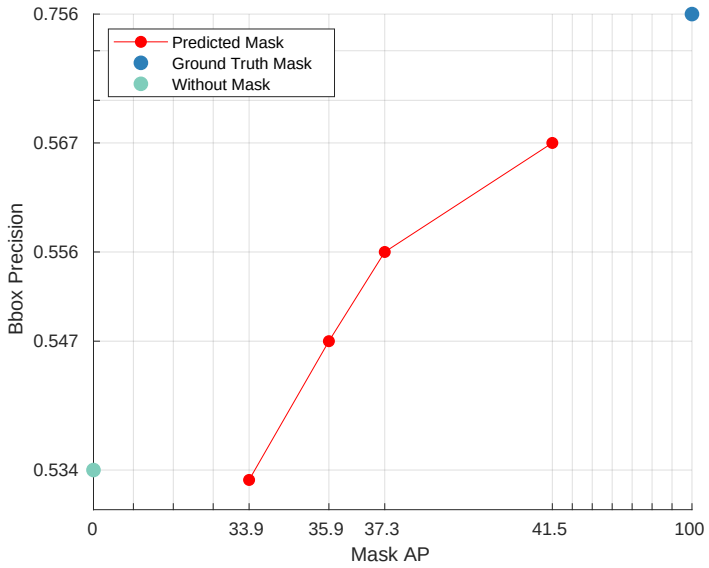



FIGURE 2.3: Effectiveness of segmentation masks with different AP. The green point and blue point indicate down-bound and up-bound respectively, which are models trained without mask and with ground truth mask. The red line shows the relationship between detector performance and quality of input masks.

In addition, to further explore the effectiveness of segmentation masks, we evaluated the models trained with the masks which are generated by different backbone networks (see Fig. 2.3), from left to right the backbone networks are ResNet-50, ResNet-101, ResNext-101,

ResNext-152. It is easy to find that the detector performance enjoys a significant improvement while we feed higher quality segmentation masks to the detectors. However, we also notice that there is still a gap between predicted segmentation mask and ground-truth mask.

Moreover, it is worth noting that the score thresh $S_\tau$ which is used in Eq. 2.4 can also affect the detection performance. We analyze the sensitivity of the threshold $S_\tau$ (see Table 2.4).

| $S_\tau$ | AP | $AP^S$ | $AP^M$ | $AP^L$ | AR | $AR^S$ | $AR^M$ | $AR^L$ |
|---|---|---|---|---|---|---|---|---|
| 0.0 | 0.559 | 0.377 | 0.628 | 0.735 | 0.646 | 0.497 | 0.704 | 0.801 |
| 0.1 | 0.562 | 0.383 | 0.629 | 0.736 | 0.646 | 0.498 | 0.704 | 0.802 |
| 0.2 | 0.565 | 0.388 | 0.634 | 0.737 | 0.647 | 0.498 | 0.706 | 0.803 |
| 0.3 | 0.566 | 0.391 | 0.635 | 0.736 | 0.646 | 0.495 | 0.706 | 0.801 |
| 0.4 | 0.567 | 0.392 | 0.635 | 0.736 | 0.645 | 0.493 | 0.706 | 0.801 |
| 0.5 | 0.567 | 0.392 | 0.637 | 0.738 | 0.643 | 0.487 | 0.706 | 0.801 |
| 0.6 | 0.566 | 0.392 | 0.636 | 0.738 | 0.640 | 0.482 | 0.704 | 0.800 |
| 0.7 | 0.563 | 0.386 | 0.634 | 0.737 | 0.633 | 0.470 | 0.700 | 0.798 |
| 0.8 | 0.559 | 0.378 | 0.635 | 0.737 | 0.624 | 0.450 | 0.698 | 0.799 |
| 0.9 | 0.546 | 0.352 | 0.629 | 0.736 | 0.603 | 0.409 | 0.686 | 0.797 |

TABLE 2.4: Analysis of the score threshold $S_\tau$ sensitivity

We can see that a lower $S_\tau$ may led to a high average recall while a higher $S_\tau$ can offer a better detection accuracy. Thus far, a suitable $S_\tau$ can filter the noise in the predictions and keep rich semantic context at the same time, which can be beneficial to the detection performance.

**CrowdHuman.** Table 2.5 and Table 2.6 show the evaluation results on CrowdHuman dataset. The segmentation masks are generated by the same model trained on the MS-COCO dataset without any further fine-tuning, because the CrowdHuman dataset does not provide segmentation annotations. The experiment settings are identical to MS-COCO Persons (see Table 2.1), both semantic and instance models obtain a improvement compared to the baseline SSD detector.

In addition, to verify that the extra semantic context can improve the SSD detectors performance on small and medium objects, the average precisions on small and medium targets are also evaluated, and we again see that the proposed method indeed achieves better performance on those small and medium objects.

| Model Name | Input | Mask | AP | AR |
|---|---|---|---|---|
| Ours-FRCNN-baseline | RGB | - | 0.384 | 0.465 |
| Sem-X-B | RGB-M | DeepLabv3+ | 0.395 | 0.473 |
| Ins-R101-B | RGB-M | Mask R-CNN | 0.393 | 0.474 |
| Ins-R152-B | RGB-M | Mask R-CNN | 0.425 | 0.504 |

TABLE 2.5: Faster R-CNN framework detection performance comparison with baseline detector and our proposed mask-guided detectors on CrowdHuman.

| Model Name | Input | Mask | AP | AR |
|---|---|---|---|---|
| Res-50-SSD | RGB | - | 0.311 | 0.393 |
| Sem-X-B | RGB-M | DeepLabv3+ | 0.324 | 0.415 |
| Ins-R101-B | RGB-M | Mask R-CNN | 0.318 | 0.411 |

TABLE 2.6: SSD framework detection performance comparison with baseline detector and our proposed mask-guided detectors on CrowdHuman.

Moreover, Table 2.7 shows comparison of computational cost on a single Nvidia GTX 1060 GPU between the proposed method and baseline detectors, batch size was set to 1 for both Faster R-CNN and SSD frameworks.

| Name | Segmentation | Detection |
|---|---|---|
| Faster R-CNN | - | 0.161s |
| Sem-X-B | 0.682s | 0.165s |
| Ins-R101-B | 0.163s | 0.165s |
| SSD | - | 0.006s |
| Sem-X-B | 0.682s | 0.006s |
| Ins-R101-B | 0.163s | 0.006s |

TABLE 2.7: Comparison of computational cost between baseline detectors and proposed method.

### 2.4.4 Qualitative Results

To gain insights on how the segmentation masks improve the performance of human detectors, we visualize the predicted bounding boxes and input segmentation masks on the MS-COCO compared with the baseline detectors.

We can see that our method gains a better performance in the heavy occlusion cases and smaller objects (see Fig. 2.4). This may be due to the fact that the segmentation masks provide

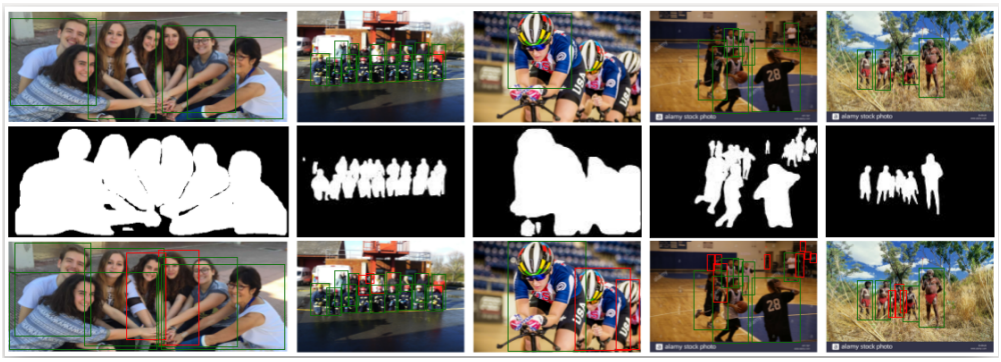
FIGURE 2.4: Visualization of robustness of the proposed method. Left column: when less accurate segmentation masks are fed into the proposed detector, satisfied results can still be predicted. Right column: the proposed detector can perform well when meet small and heavy occluded targets.

extra semantic context and play as an attention mechanism, which can help the detectors to focus more on the regions where potential object candidates may appear. Meanwhile, the proposed method also shows a high robustness to the poor segmentation masks. As image segmentation is a pixel-level vision task, thus the segmentation mask can be interfered when the targets are occluded by other objects. In this case, the segmentation mask can be cut into a number of irregular pieces. However, our proposed method can also handle those separated segmentation masks robustly.

## 2.5 Discussion

*Does the semantic segmentation mask really guide the CNN to learn better features?*

(a) COCO dataset



(b) CrowdHuman dataset

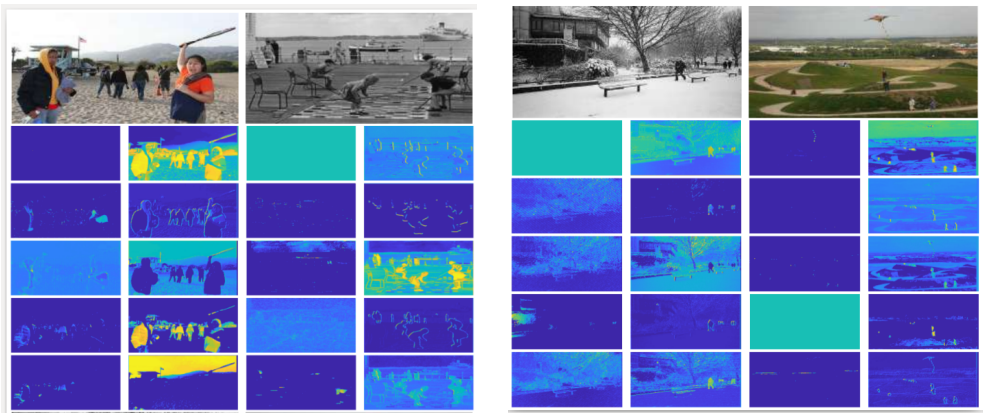
FIGURE 2.5: Visualization of detection results from the Mask R-CNN [1] and our proposed detector on MS-COCO Persons and CrowdHuman, both detectors use ResNet-50 as backbone network. The first row and third row are detection results from the Mask R-CNN [1] and our proposed detector respectively. The middle row shows the generated segmentation masks which were fed into the mask-guided detector. The results show that segmentation masks can play as an attention mechanism and help the detectors to notice small and heavy occluded persons. Green boxes indicate similar detection results; Red boxes indicate better detection results.

### 2.5.1 Intuitive and Qualitative Analysis

To further explore the effectiveness of the segmentation masks, we visualize and compare the features extracted by the baseline detector and our proposed detector (see Fig. 2.7). Two columns of features are showed under each image, features from the left column are extracted by the baseline detector while the others are extracted by the proposed method. We observe that the added segmentation masks can guide the convolutional neural network to learn more discriminative representations.


FIGURE 2.6: Deep features of different quality (better viewed in color), feature quality of a single channel featuremap is calculated by Eq. 2.10.

According to these featuremaps, we see that the features from both baseline detector and proposed detector share a considerable overlap with the input segmentation mask, even though the baseline detector is trained without any extra input. It is clear that these learned features carry extremely rich semantic context and can be highly useful for the detectors to discriminate the foreground objects from background. Furthermore, benefiting from the segmentation mask, the features extracted by the proposed method gain stronger response to both background and foreground. Thus we believe that the external input segmentation mask can guide the detectors to learn more discriminative features, which is the key to improve the detection performance.



(a) (b)

FIGURE 2.7: Comparison and visualization of learned features from Mask R-CNN and our proposed method (better viewed in color). As the features extracted from deeper layers are too abstract, we only visualize the features from very shallow layers here. There are two columns of features under each input image, the left column shows the features extracted by Mask R-CNN which is learned without mask guide, while the right column shows the features learned by the proposed mask-guided detector. Images are selected from the MS COCO val. 2017 dataset.

### 2.5.2 Objective and Quantitative Analysis

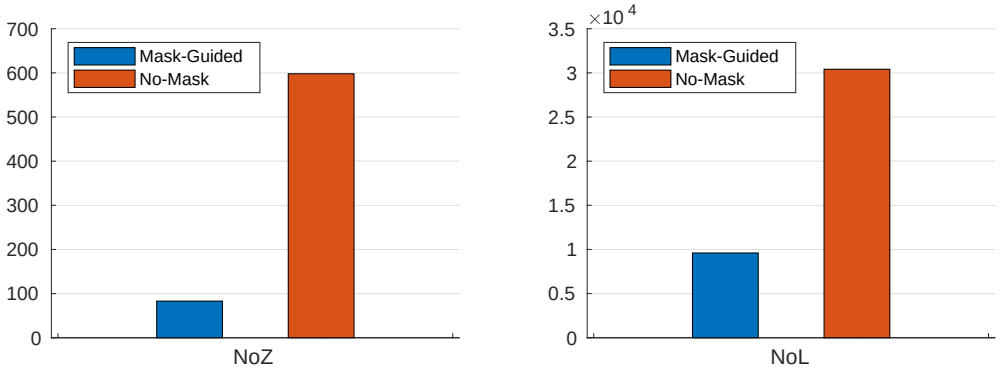



FIGURE 2.8: Comparison of feature quality between the baseline detector and our proposed method. NoZ and NoL are Number of Zero activation (see Eq. 2.8) and Number of Low quality featuremaps (see Eq. 2.9) respectively, the lower the better.

During the last decade, as the surge of deep learning, variety of deep neural network architectures are designed by experienced experts, such as AlexNet [10], VGGNet [57], GoogleNet [12] and ResNet [13], which are widely used in many different vision tasks, like image classification, object detection and visual object tracking. However, these networks are designed for different tasks or different datasets at the very beginning time. Most vision tasks pre-train these backbone networks on a very large image classification benchmark, such as the ImageNet dataset [83] then adapt the pre-trained model to the target domain by fine-tuning on new datasets. Although considerable performance improvements can be gain by this pipeline, researchers start to investigate new methods to improve performance and efficiency for deep neural networks, methods such as adversarial learning [84], deep neural architectures search [85], channel pruning [86] and knowledge distillation [87] become more and more popular among the computer vision community. Benefiting from these methods, deep neural networks can efficiently learn more discriminative deep features. Features of higher quality can usually significantly improve the final task performance for most computer vision task, such as object detection, semantic segmentation and object tracking.

Beyond intuition, we want to evaluate whether the proposed method can truly enhance the quality of extracted feature maps in a more objective and numerical fashion. Thus we introduce Average Percentage of Zeros (APoZ) here, which was firstly proposed in the deep

model compression task [88]. The APoZ is defined to measure the activation of neurons, neurons with higher values of APoZ are considered more redundant in the deep networks. Therefore, the features extracted by these neurons are more likely to have lower quality, and make less contribution to the final performance. Different from the original proposed APoZ, we define Number of Zero activation (NoZ) to evaluate feature maps here. Let $O_c^{(i)}$ denotes the output featuremap of *c*-th channel in *i*-th layer, then the NoZ can be denoted as $NoZ_c^{(i)}$:

$$NoZ_c^{(i)} = NoZ(O_C^{(i)}) = \sum_k^N \sum_j^M g(O_{c,j}^{(i)}(k) = 0) \tag{2.8}$$

where $g(\cdot) = 1$ if true, and $g(\cdot) = 0$ if false, $M$ and $N$ are dimension of output featuremaps and total number of validation examples respectively.

Similar to the NoZ (Eq. 2.8), we define Number of Low quality feature maps (NoL) to further evaluate the quality of learned deep features, which writes:

$$NoL_c^{(i)} = \sum_k^N \sum_j^M g(q(O_{c,j}^{(i)}(k), B_n) < \eta) \tag{2.9}$$

where $g(\cdot) = 1$ if true, and $g(\cdot) = 0$ if false, $M$ and $N$ are dimension of output feature maps and total number of validation examples respectively. $B_n$ is ground-truth bounding boxes of the $n$-th validation example, $\eta$ is quality constant which set to 0.05 in practice. $q(\cdot)$ is a quality function which used to evaluate the quality of a single channel feature map (see Eq. 2.10).

For high quality feature maps, it should be easy to discriminate foreground part from background part. Therefore, we define a quality function $q(\cdot)$ to measure the feature map quality for each channel. Let $X$ denotes the feature map waiting to be evaluated, and $B$ denotes ground-truth bounding boxes of the current validation example, then bounding box of each instance can be denoted as $b_n = [x_n, y_n, w_n, h_n]$. Further, foreground feature map for the $n$-th box can be denoted as $X(b_n)$, which is a $w_n \times h_n$ matrix. The average of overall background

feature map can be denoted as $\mu(\bar{X}(B))$, which is a scalar. Then the quality function writes:

$$q(X,B) = \frac{\sum_n^N (\frac{\sum_j^{h_n} \sum_i^{w_n} X^{i,j}(b_n)}{w_n \times h_n} - \mu(\bar{X}(B)))}{N} \tag{2.10}$$

We evaluate the quality of features extracted by both baseline detector and our proposed method on the COCO Person dataset. Fig. 2.8 shows the comparison of NoZ and NoL between baseline detector and our proposed method. As shown in Fig. 2.8, benefit from the guidance of segmentation mask, the proposed method significantly reduce the Number of Zero activation (NoZ) and the Number of Low quality features (NoL), which indicates that the proposed detector learns much more discriminative features. By adapting the proposed simple yet effective training procedure, almost 75% zero activation neurons learn new representations, and approximate 60% low quality features are improved to a higher quality.

## 2.6 Conclusion

In this paper, we have presented a simple method for improving human detectors with extra semantic features by aggregating the original RGB images with segmentation masks. We implement our method on two popular detection frameworks, Faster R-CNN and SSD, and evaluate the proposed method on two datasets *i.e.* MS-COCO Persons and CrowdHuman.

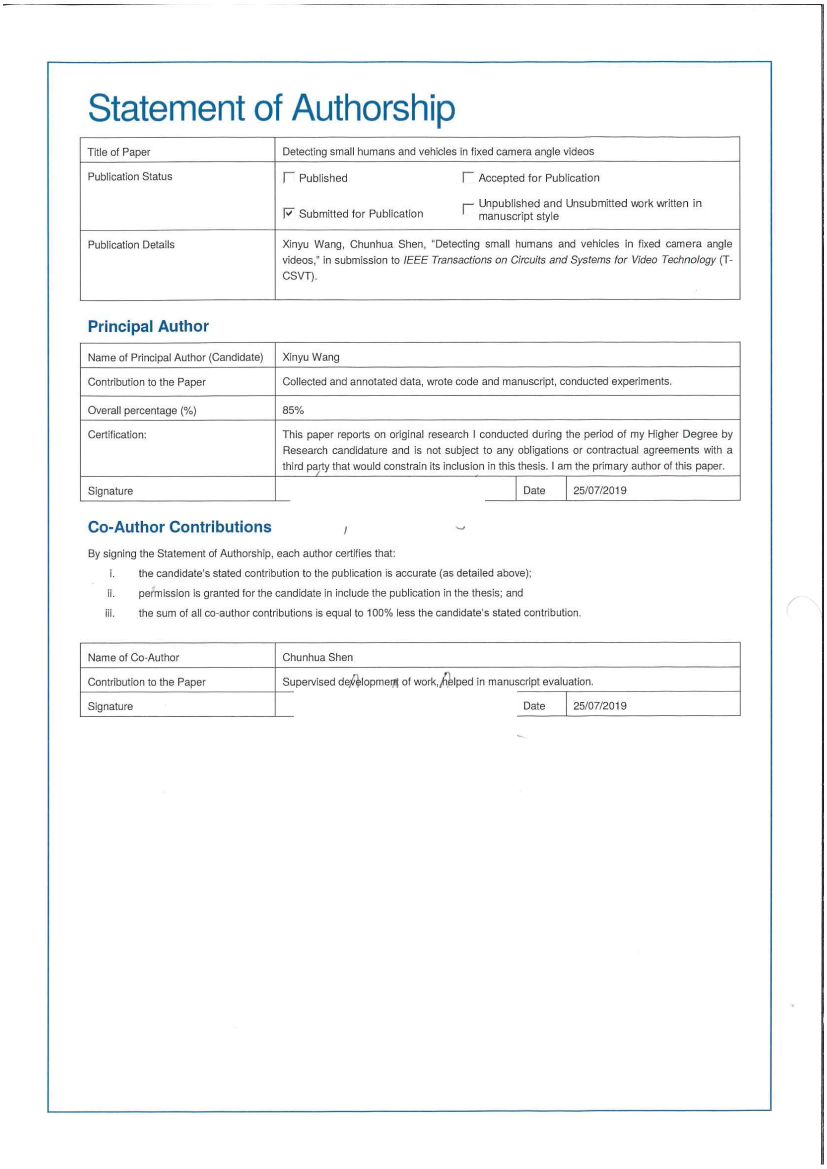
Our experiments show that the external segmentation masks can significantly improve the human detection performance on both detection frameworks. Moreover, we experimentally analyze the effectiveness of the segmentation masks generated by different methods and reveal the power of extra semantic context. In addition, to gain insights on how can the segmentation masks guide the convolutional neural network to learn more discriminative features, we visualize the learned features from both baseline detectors and proposed detectors. Meanwhile, we propose a quality function to measure the quality of learned deep features in a numerical way. Two metrics which termed NoZ and NoL, based on the proposed quality function, are employed to evaluate the feature quality. According to these metrics, we found that the mask-guided human detector learned more discriminative and higher quality features,

the number of zero activation and low quality feature maps significantly decreased compared to the baseline detector.

Admittedly, the segmentation masks can lift the human detection performance significantly. However, in this work, the segmentation module and detection module stand alone with each other, and the segmentation masks are generated in advance. One possible future direction can be to integrate these two procedure together and train the multiple tasks jointly. Besides, we have tested our method on human pose estimation tasks, and the proposed method also gains a slight improvement. Thus we believe that extra semantic context can also improve the performance of other vision tasks, meaning that our proposed mask-guided detector can be easily extended and adapted.

CHAPTER 3

# Detecting Small Humans and Vehicles in Fixed Camera Angle Videos



## Statement of Authorship

| | |
|---|---|
| Title of Paper | Detecting small humans and vehicles in fixed camera angle videos |
| Publication Status | ☐ Published ☐ Accepted for Publication<br>☑ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style |
| Publication Details | Xinyu Wang, Chunhua Shen, "Detecting small humans and vehicles in fixed camera angle videos," in submission to *IEEE Transactions on Circuits and Systems for Video Technology* (T-CSVT). |

### Principal Author

| | | | |
|---|---|---|---|
| Name of Principal Author (Candidate) | Xinyu Wang | | |
| Contribution to the Paper | Collected and annotated data, wrote code and manuscript, conducted experiments. | | |
| Overall percentage (%) | 85% | | |
| Certification: | This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper. | | |
| Signature | | Date | 25/07/2019 |

### Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);
ii. permission is granted for the candidate in include the publication in the thesis; and
iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

| | | | |
|---|---|---|---|
| Name of Co-Author | Chunhua Shen | | |
| Contribution to the Paper | Supervised development of work, helped in manuscript evaluation. | | |
| Signature | | Date | 25/07/2019 |

FIGURE 3.1: Statement of Authorship for Paper “Detecting Small Humans and Vehicles in Fixed Camera Angle Videos"

## 3.1 Introduction

Object detection is a fundamental problem in computer vision, and it has been a cornerstone for many real-world applications such as self-driving vehicles, advanced driver-assistance systems (ADAS), intelligent transportation systems and video surveillance. Benefiting from the powerful convolutional neural networks, modern object detectors such as Faster R-CNN [16], SSD [17] and YOLO [18] can easily predict accurate bounding boxes for most large objects. However, existing methods are less sensible to the targets which have a very small size in the input images. Nonetheless, in many real-world applications, for example, intelligent transportation video surveillance systems, it is necessary to detect far-away tiny objects, because these objects such as vehicles can move in a very fast speed and reach to the camera in a short time.

In recent decades, many object detection datasets have been introduced for both generic object detection task and specific object detection task, such as [2, 3, 4, 51, 89, 90, 91], which have enabled great progress in this area. However, as shown in Fig. 3.2, most of these existing datasets suffer from a number of drawbacks:

1 Lower density: although some datasets provide a large number of images or video frames, the average density of object is extremely low, which might induces an imbalance between positive and negative samples.

2 Poor annotations for tiny/crowed objects: as these datasets are not designed for tiny/crowded object detection task, they usually do not provide fine annotations for those smaller objects, and areas with group objects are simply labeled as ignore regions or even nothing.

3 Missing/Lower Quality annotations: As shown in Fig. 3.2, many object annotations are missed. Such poor annotations may introduce confusing samples and have a negative impact on the detector performance. Besides, in some datasets, for annotation convenience, aspect ratio of objects is fixed to a constant (*e.g* 0.41 for pedestrians in [2]), which cannot fit humans with different poses appropriately.

Hence, it is difficult to use these datasets to measure the ability of detectors in detecting tiny/crowded targets.

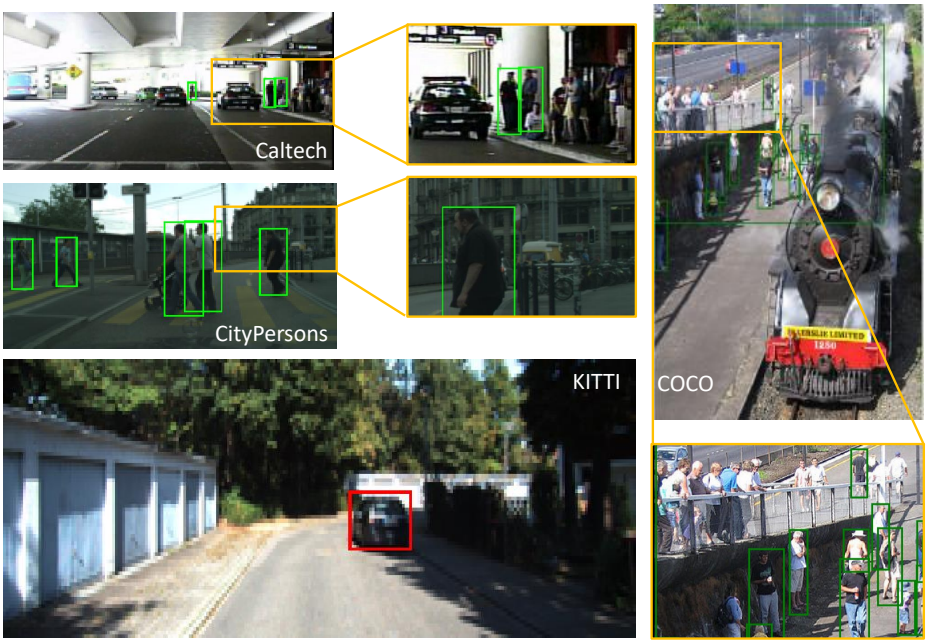



FIGURE 3.2: Visualized annotations of existing widely-used datasets (best viewed in color). Green and Red bounding boxes are manually labeled ground-truth annotations provided by the official dataset.

In this paper, we introduce a new dataset named SHV, which is designed for fixed angle video surveillance systems, two main categories of objects are annotated, *i.e.*, humans and vehicles. Compared to existing datasets, a large amount of tiny humans and vehicles are annotated. In addition, the average number of annotated objects is approximately $\sim$27 per frame in our proposed dataset, which has a much higher density than existing ones. Accordingly, the proposed dataset can be considered as a benchmark for evaluating the performance of detectors for tiny/crowded targets.

## 3.2 Related Work

### 3.2.1 CNN-based Object Detector

In recent years, as the prevalence of deep learning technologies, CNN-based object detectors have achieving impressive performance on both generic object detection and specific object detection tasks. CNN-based object detectors can be basically categorised into two types: one-stage approaches and two-stage approaches.

**Two-stage detectors**, such as [1, 15, 16], divide object detection task into two stages: extracting Region of Interest (RoIs) and classifying RoIs into foreground/background.

In [24], the authors proposed a Selective Search method to generate a set of candidate proposals which contain objects of all categories while filtering out most negative locations at the first stage, and then use SVM with SIFT [25] features to classify the proposals into different classes at the second stage. R-CNN [14] improved this approach by employing CNN as feature extractor to replace SIFT, which achieved a significant improvements on detection accuracy. In more recent, improved version of R-CNN, such as Fast R-CNN [15], Faster R-CNN [16] and Mask R-CNN [1] have been proposed. The core idea of these variants of R-CNN is to devise a Region Proposal Network (RPN), which integrated RoIs generation with feature extraction for the second stage into a single CNN. RPN significantly improved two-stage based detectors in both detection accuracy and speed.

However, it is admitted that, RPN-based detectors introduce excessively many hyper parameters, such as anchor sizes, anchor stride and anchor aspect ratios, which need to be carefully tuned in different datasets (especially for small targets) to achieve satisfied accuracy [2, 59].

**One-stage detectors**, such as [17, 18, 21], abandon the RoIs generation process in two-stage based detection framework, and directly regress and classify a set of pre-defined candidate anchor boxes.

YOLO [18] simply divides the input image into an $S \times S$ grid, and simultaneously predict bounding boxes and categories in those boxes, which achieved very fast inference speed.

SSD [17] pre-defines a set of “default boxes", and use deep features from different levels of convolutional layers to regress and classify these “default boxes". Retina-Net [21] designed Focal Loss to solve the imbalance between positive and negative samples, but still relies heavily on the anchor boxes. These approaches [17, 18] have been tuned for very fast inference speed but their detection performance trails that of most two-stage based detectors. Meanwhile, since the SSD regresses small targets on shallower convolutional layers, it has been blamed for worse accuracy on tiny objects. Also, as each grid cell in YOLO only predicts two boxes and can only have one class, it cannot perform well on small/crowded objects.

More recently, many anchor free based single-stage detectors have been proposed [23, 27, 28]. CornerNet [27] detects an object bounding box as a pair of keypoints (top-left and bottom-right corners), however, complicated post-processing stage is required to group the pairs of corners belonging to the same instance. FCOS [23] formulates the object detection task into a per image pixel prediction fashion and achieves promising performance on public dataset.

### 3.2.2 Dataset for Object Detection

TABLE 3.1: Comparison of widely used benchmarks

| Dataset | Type | Video | Year |
|---|---|---|---|
| Pascal VOC[89] | generic | N | 2010 |
| MS COCO[4] | generic | N | 2014 |
| INRIA[51] | pedestrian | N | 2005 |
| Caltech[91] | pedestrian | Y (Driver Viewed) | 2012 |
| CityPersons[2] | pedestrian | Y (Driver Viewed) | 2017 |
| KITTI[90] | human†/vehicle | Y (Driver Viewed) | 2012 |
| UA-DETRAC[3] | vehicle | Y (Fixed Camera) | 2015 |
| Ours | human†/vehicle | Y (Fixed Camera) | 2019 |

† Human includes pedestrian, cyclist, sitting person and etc.

As shown in Table 3.1, many detection benchmarks have been proposed in last decades, which include both generic and specific object detection tasks.

**Generic object detection** aims at predicting positions of object instances from a large number of pre-defined categories in natural images/videos. Pascal VOC [89] and MS COCO [4] are two of the most popular benchmarks that used among generic object detection community.

The Pascal VOC dataset has approximately 11k training images (VOC 2012) which contains more than 27k annotated objects from 20 categories, while the MS COCO dataset contains ∼120k training images (train/val 2017) with over 500k object instances from 80 categories. It is admitted that both datasets provide successful benchmarks for the community to evaluate generic object detectors. However, such generic object datasets suffer from a lower average object density, *i.e.*, ∼2.4 and ∼4.2 annotated objects per image for Pascal VOC and MS COCO dataset respectively. Meanwhile, as shown in Fig. 3.2, MS COCO only provides weak annotations for crowded objects, many human instances are not annotated in the example image. Therefore, generic object datasets can hardly be used as benchmarks for evaluating the ability of detectors in locating tiny/crowded targets.

**Specific object detection** focuses on locating bounding boxes for objects from one or two specific object categories, *e.g* pedestrian [92], vehicle [3], text [93] and etc. Compared to generic object detection tasks, specific object detection benchmarks usually contain more difficult cases, such as heavy occluded and/or crowded targets. Caltech [91] and CityPersons [2] are two widely used datasets for pedestrian detection task. However, as shown in Fig. 3.2, the annotation of Caltech Pedestrian dataset is very rough, many instances were not annotated correctly. CityPersons is a dataset built upon the Cityscapes semantic segmentation dataset [94], benefit from the high-quality of segmentation annotations, CityPersons provide much finer ground-truth labels than the Caltech Pedestrian dataset. It is noteworthy that all of the bounding boxes annotated for pedestrians are forced to a fixed aspect ratio (0.41), which the authors claim can provide good alignments. Due to this annotation policy, people with "unusual poses", cyclists and sitting persons are marked as ignore regions in CityPersons. UA-DETRAC [3] is a similar benchmark to the proposed dataset, which also provides fixed camera angle videos and correspond annotations. However, UA-DETRAC only annotated vehicle category, and a large number of areas where include crowded and smaller sized vehicles are annotated as ignore regions. Therefore, these existing specific object detection dataset cannot be treated as a benchmark to evaluate the ability of detectors in locating tiny/crowded targets.

## 3.3 Dataset

(a) CityPersons

(b) Proposed SHV dataset

FIGURE 3.3: Comparison of annotation protocols between CityPersons [2] and ours. (a) shows instances selected from CityPersons, green solid boxes are ground-truth provided by the official dataset, yellow dash boxes are annotations under our protocol. (b) shows pedestrians selected from proposed dataset, yellow solid boxes are ground-truth provided by our dataset, green dash boxes are labeled under a fixed aspect ratio fashion, which was employed by CityPersons.

In this section, we firstly introduce the annotation policy in our proposed dataset, and show annotations between different protocols for comparison. Then, we provide statistics of the proposed dataset and other widely-used benchmarks, includes the distribution of object size, aspect ratio and etc. Finally, we show the evaluation metrics.

### 3.3.1 Bounding Box Annotations

**Human Annotation:** As shown in Fig. 3.3, we use a different annotation protocol compared to the recent proposed pedestrian detection benchmark CityPersons [2]. In [2], the pedestrians are annotated by drawing a line from the top of the head to the middle of feet, and the bounding box is then generated using a fixed aspect ratio (0.41). By annotating instances in this fixed aspect ratio fashion, some issues may raised: 1) Persons with different poses have varying aspect ratio, simply fixing the aspect ratio can includes unnecessary background

in and/or exclude human parts from the bounding boxes. Also, sometimes poor alignments would occurred by this approach (see Fig. 3.3 for examples). 2) The fixed aspect ratio cannot be applied to “sitting persons", “cyclists" and etc., thus these persons are marked as ignore regions in [2], which is harmful to the dataset diversity and can induce negative impacts on detector performance. In consequence, we do not follow the fixed aspect ratio policy in our proposed dataset, all part of the instance object is included in a rectangle bounding box (see Fig. 3.3 for details).

**Vehicle Annotation:** As the vehicles are rigid objects, and the annotation protocols for such type of targets are almost identical among different datasets. Thus we do not give more details here, we refer readers to Fig. 3.8 for annotation examples.

**Ignore Region Annotation:** Apart from “real" positive training samples, it is common to see some areas are labeled as ignore regions in many datasets [2, 3, 4, 90], due to low resolution, fake objects in posters or too crowded to be annotated. However, due to the variety of different annotation protocols and annotators, some ignore regions are unnecessary in the existing datasets. CityPersons was built on the basis of semantic segmentation masks provided by Cityscapes, and inherited pixel-level definitions from the Cityscapes, which are however inappropriate for detection tasks. As shown in Fig. 3.4 (a), a large number of unreasonable ignore regions are marked in the CityPersons dataset, including traffic signs, traffic lights, hardly recognised people in the vehicles and etc. Also, in UA-DETRAC dataset, most vehicles of small or even median sizes are not labeled, areas include these instances are treated as ignore regions (see Fig. 3.4 (b)). Such annotation policy may raise some issues: 1) Unnecessary handle for unreasonable ignore regions has to be done. 2) Diversity of dataset would be harmed, and may further influence the detector performance. 3) Cannot be used for evaluating the CNN ability in detecting tiny/crowded targets. Therefore, in our proposed dataset, only a few instances with heavy occlusion, or fake objects in pictures would be labeled as ignore objects.

(a) CityPersons

(b) UA-DETRAC

(c) Proposed SHV dataset

FIGURE 3.4: Comparison of ignore regions among two widely used benchmarks and ours. Red boxes indicate ignore regions marked in (a) CityPersons [2], (b) UA-DETRAC [3] and (c) proposed SHV dataset. Many unreasonable ignore regions are included in CityPersons dataset while most vehicles of smaller size are directly treated as general backgrounds.

### 3.3.2 Statistics

**Volume:** As shown in Table 3.2, we compare the number of images and annotations between our proposed dataset and widely-used datasets. In a total of 12k video frames, we provide ∼170k person, ∼130k vehicles and ∼50k ignore region annotations. And we notice that the density of annotated objects in proposed dataset is much higher than that in all of four other widely-used datasets (see Table 3.3). Although we do not provide a validation set as CityPersons, the users can split the training set into two subsets by themselves.

**Object Size:** Since the purpose of the proposed dataset is to provide a benchmark for evaluating ability of detectors in recognising tiny/crowded objects, the proposed dataset is

TABLE 3.2: Comparison of dataset volume between proposed dataset and widely-used datasets

| Dataset | Training Set | | | | Validation Set | | | | Testing Set | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Image | Human | Vehicle | Ignore | Image | Human | Vehicle | Ignore | Image | Human | Vehicle | Ignore |
| Ours | 8,881 | 137,738 | 103,813 | 37,977 | - | - | - | - | 3,600 | 32,363 | 25,930 | 10,524 |
| Caltech_1x | 4,250 | 5,564 | N/A | 4,992 | - | - | - | - | 4,024 | 1,349 | N/A | 0 |
| CityPersons | 2,975 | 19,654 | N/A | 6,768 | 500 | 3,938 | N/A | 1,631 | 1,525 | 11,424 | N/A | 4,773 |
| KITTI | 7,481 | 6,336 | 4,519 | 11,295 | - | - | - | - | 7,518 | ? | ? | ? |
| UA-DETRAC | 83,791 | N/A | 577,899 | | - | - | - | - | 56,340 | N/A | 632,270 | |

consist of a large number of small objects. We follow the object size definition used in MS COCO [4], and further split the “small" objects into two subsets, *i.e.*, “tiny" and “small". The detailed division of object sizes are: *Large* (area $\in [96^2, Inf]$), *Medium* (area $\in [32^2, 96^2]$), *Small* (area $\in [16^2, 32^2]$) and *Tiny* (area $\in [0, 16^2]$). As shown in Fig. 3.5, only $\sim$1% humans and vehicles are of large size in our dataset, while more than 80% humans and 50% vehicles are of small and tiny size. Such distribution of object size allows the proposed dataset to become a benchmark which focuses on tiny human/vehicles detection task. Compared to our proposed dataset, [90] and [2] include $\sim$23% and $\sim$22% large humans in the training set, and less than $\sim$3% and $\sim$5% tiny objects respectively. For vehicles, [90] contains $\sim$30% of large samples while only less than $\sim$0.5% tiny instances are included. In addition, we show the distribution of bounding box aspect ratio for “human" category in Fig. 3.6. It can be seen, most boxes are forced to an aspect ratio of 0.41 in [2], and the authors also proposed a customised Faster R-CNN which equipped with specific designed RPN scales, ratios and strides based on the fixed aspect ratio, similar work was done in [59]. By tuning these hyper-parameters for specific objects at training stage can indeed improve the detection performance. However, such strategy is not only time consuming but also can hardly be adapted to another different dataset. Consequently, we do not use a strict aspect ratio to forcibly align the targets. But as shown in Fig. 3.6, we can see most annotated “human" boxes have an aspect ratio between [0.3, 0.5].

### 3.3.3 Benchmark

**Evaluation Metric:** We follow the same evaluation protocol as used for MS COCO [4], *i.e.*, AP and AR which are averaged over multiple Intersection over Union (IoU) values, and 10

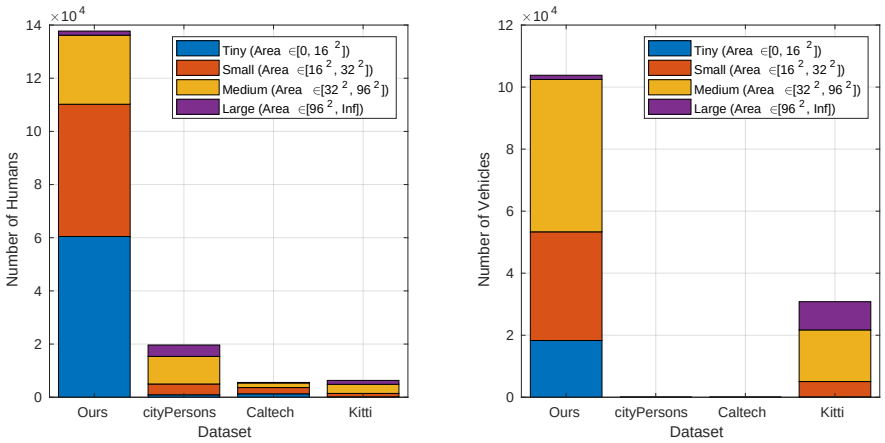



FIGURE 3.5: Comparison of object size distribution between the proposed dataset and three other widely-used datasets. We use the similar definition of object sizes introduced in MS COCO dataset [4], and the “small" object category is further split into “small" and “tiny" subsets.

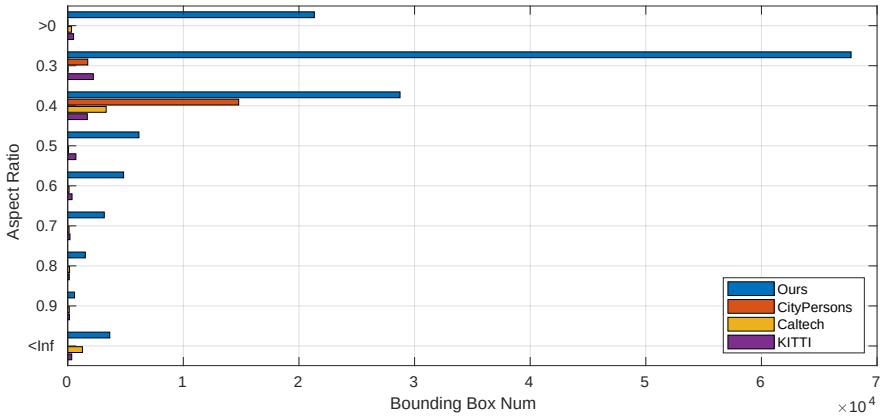



FIGURE 3.6: Comparison of “human" bounding box aspect ratio distribution between our proposed dataset and widely-used datasets (only positive samples are considered).

TABLE 3.3: Comparison of object density in training set

| Dataset | Human/img | Vehicle/img | Ignore/img |
|---|---|---|---|
| Caltech_1x | 1.31 | N/A | 1.17 |
| CityPersons | 6.61 | N/A | 2.27 |
| KITTI | 0.85 | 0.60 | 1.51 |
| UA-DETRAC | N/A | 6.90 | |
| Ours | **15.51** | **11.79** | 4.28 |

thresholds of $[.50 : .05 : .95]$ which is same as COCO are used for evaluation. Averaging over IoUs rewards detectors with better localisation. In addition, it is important to note that,

although fine grained class are provided (*e.g* car, truck, bus and etc.), the final evaluation only based on the super category, *i.e.*, *human* and *vehicle*, thus overall AP, $\text{AP}_{human}$ and $\text{AP}_{vehicle}$ will be reported.

With the publication of this paper, we will create a website for the SHV dataset, where train/test images and train annotations can be downloaded, and an online evaluation server will be installed for computing detection performance based on the test annotations.

## 3.4 Baseline Method and Experiments

In this section, we evaluate some popular object detectors on our proposed dataset. Moreover, we also introduce to exploit motion information between video frames via aggregate motion features with RGB channels, which can significantly improve the detection performance.

### 3.4.1 Motion exploiting channel

Employing extra feature channels such as depth map, optical flow, saliency map to boost the original RGB images have been a widely used approach to enhance object detectors. In [54], the authors introduce a “HyperLearner" to learn representations of channel features such as semantic segmentation mask, edge and heat-map channel for pedestrian detection, and found that integrating external features into the network can boost the detectors working on images of both low and high resolution, thus can improve the detection accuracy. In [62], the authors propose “Histogram of Oriented Depth" (HOD) which encodes the direction of depth changes to the original RGB images, with the extra depth channel, the detector can easily discriminate foreground objects from background. Also, many previous works have proved that video related features such as motions and optical flow could be helpful to improve the performance for video-based computer vision tasks [95, 96, 97]. In [95], “Optical Flow guided Feature" (OFF) is learned by introducing optical flow into the convolutional neural network, which the authors claim can enable neural network to distill temporal information. Unfortunately, generating extra features like segmentation mask, depth, saliency maps and optical flow is not only time-consuming but also cost remarkable computational resources, thus the trade-off

between accuracy and speed among these methods becomes a problem for most real-world applications.

It is interesting to note that, human beings can easily catch moving objects even though the targets are of very small sizes. Motivated by this phenomenon, we believe motions can be considered as reasonable features for object detection task in videos. Hence, one of the goals of this paper is to devise a simple yet effective way to exploit the motion information without introducing heavy computational cost. Inspired by [97], we propose to use additional motion channels to guide the convolutional neural networks to learn motion priors, this operation significantly enhance the detectors ability in recognising moving small objects. Suppose the gray-scale image for current video frame is $I_i$, step is $s$, then the base motion map $\mathcal{M}_{base}$ can be simply defined as:

$$\mathcal{M}_{base} = \frac{|I_i - I_{i-s}| + |I_i - I_{i+s}|}{2} \tag{3.1}$$

Simply adding base motion map into the RGB image may encounter with two problems: a) neural networks can be easily overfit to the motion channel b) still objects without motions have a high probability to be missed. Consequently, to avoid the above issues, shifted motion channel is generated as a support to the base motion map. Shifted motion between the $i^{th}$ and $j^{th}$ frames can be calculated by:

$$\mathcal{S}(I_i, I_j) = \frac{U(I_i, I_j) + D(I_i, I_j) + L(I_i, I_j) + R(I_i, I_j)}{4} \tag{3.2}$$

where $U$, $D$, $L$, $R$ are Motion Shifting Operations (MSO) that defined as followings:

$$MSO(I_i, Ij) \begin{cases} U(I_i, I_j) = |I_i - \uparrow (I_j, o)| \\ D(I_i, I_j) = |I_i - \downarrow (I_j, o)| \\ L(I_i, I_j) = |I_i - \leftarrow (I_j, o)| \\ R(I_i, I_j) = |I_i - \rightarrow (I_j, o)| \end{cases} \tag{3.3}$$

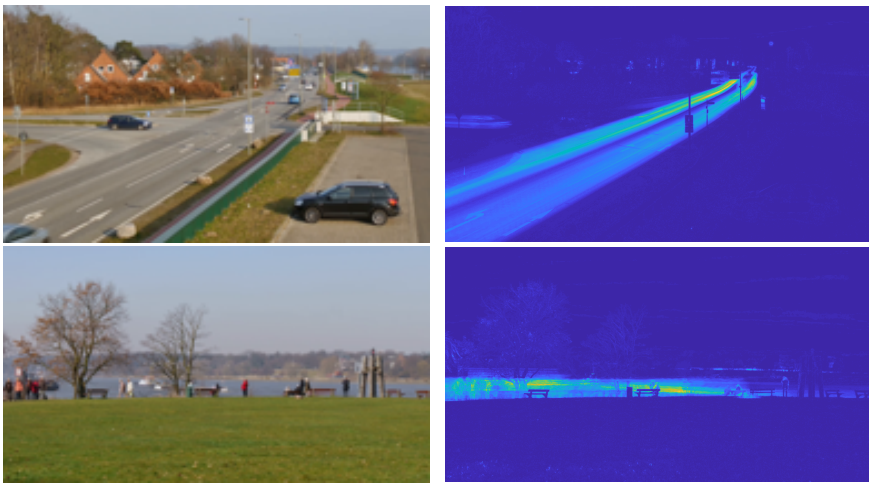


FIGURE 3.7: Motion priors among different videos (best viewed in color). Left: Original image selected from training set. Right: Heatmap of motions, which represents the probability of the area where the moving objects may appear.

where arrows $\{\uparrow, \downarrow, \leftarrow, \rightarrow\}$ are shift operators ($\uparrow (I_j, o)$ is image $I_j$ shifted up by $o$ pixel(s)), $o$ is shifting offset. Therefore, the shifted motion map $\mathcal{M}_{shift}$ can be obtained by:

$$\mathcal{M}_{shift} = \frac{S(I_i, I_{i-s}) + S(I_i, I_{i+s})}{2} \tag{3.4}$$

Once the motion channels are prepared, they would be concatenate with the original RGB channels, therefore the input data $\mathbb{I}^5$ is consist of five channels:

$$\mathbb{I}^5 = \{I_{RGB}, \mathcal{M}_{base}, \mathcal{M}_{shift}\} \tag{3.5}$$

The base and shifted motion patterns encode motion priors (see Fig. 3.7) into the input image, which can guide the neural network to pay attention to moving objects. Moreover, these motion priors allow the region proposal network (RPN) to ignore general background areas, such as sky and buildings in our dataset.

### 3.4.2 Pixel-wise information learning

In some previous works [1, 54], it has been proved that multi-task co-training can improve the detection performance. He *et al.* propose a framework termed Mask R-CNN in [1], which simultaneously trained object detector with instance segmenter. Compared to single task trained Faster R-CNN, Mask R-CNN achieves better performance by adopting the jointly training strategy. However, instance-level segmentation masks are not available for most object detection datasets, which therefore makes this strategy can hardly be implemented. Some previous works employ off-the-shelf methods to generate semantic/instance segmentation masks to assist high-level vision problems, such as person re-identification [98], person search [63] and detection [76]. Usually, the mask is used as external channel to guide the neural networks to learn better representations, which can thus be considered as a type of attention mechanism. Nonetheless, heavy computational cost has to be paid for generating masks at both training and inference stage. Therefore, directly using bounding box area as segmentation mask to supervise the external branch becomes an alternative. In [99], the authors proposed an attention mechanism termed ZIZOM which directly utilized the whole area of ground-truth bounding box as attention mask for pedestrian detection.

Inspired by these works, we use an external branch that shares the same backbone network with the detection branch to predict generated instance masks. It is important to note that, this external branch is only installed at training stage for guiding the backbone network to learn pixel-level representations, and will NOT be activated at testing stage. Therefore, no external input or extra computational cost will be requested at inference time.

For generating the attention masks, since we do not exactly ask the network to predict a highly accurate segmentation mask at testing stage, therefore, the motion masks are not requested to have very high quality. However, simply employing bounding box area as instance masks like [99] may introduce redundant background information. Hence, we propose a motion-based method to generate the masks. As introduced in eq. 3.1, base motion patterns are firstly extracted from the $i^{th}$ frame $I_i$ and its relative frame $I_{i\pm s}$, then instance-level motion is cropped by ground-truth bounding box. Furthermore, the instance motion map is transferred

into binary maps while noises are filtered simultaneously. Finally, convex hull is calculated based on a set of representative points selected from the edge of the rough map. After refining, reasonable attention masks can be obtained without bells and whistles. There is no doubt that better masks can be obtained by human annotations, which is however out of the scope of this paper. The purpose of the proposed method is to provide a simple baseline approach for the SHV dataset, and explore the possible direction for small object detection task.

However, two problems should not be ignored: a) motion patterns can hardly be obtained from still objects such as standing persons and parked vehicles; b) motion masks of poor quality could be harmful to the detection performance. Therefore, we use a combination of both motion-based and boundingbox-based masks to train the external head, *i.e.*, boundingbox-based masks would replace the motion-based mask when reasonable motions cannot be accessed. Such strategy can significantly improve the robustness of the proposed detector, especially for still objects.

### 3.4.3 Quantitative results

**Comparison with state-of-the-art detectors on SHV:** To understand the difficulties of tiny/crowded object detection on SHV, and evaluate the ability of state-of-the-art object detectors in recognising small targets, we compare the performance of several widely-used detectors on our dataset. As shown in Table 3.4, both one-stage and two-stage methods including YOLO, SSD, RetinaNet and Faster R-CNN are trained and evaluated on the proposed dataset[1], since the proportion of “large" objects is less than 1% in our dataset, we only show AP for “tiny", “small" and “median" targets (but the overall AP is calculated based on all of the positive samples). Limited by the approach proposed by the original paper and code implementations, we can not ensure all of the training hyper-parameters keep identical among different frameworks, such as input image size, backbone networks and etc. All of the detectors were trained under the widely-used settings, and details of the training settings can be found in Table 3.4.

[1]We use the MXNet GluonCV toolkit [100] for training YOLO and SSD models, while Faster R-CNN, RetinaNet models are trained based on the Facebook Mask R-CNN benchmark [101].

We also noticed that simply employing deeper backbone networks such as ResNet-101 and ResNeXt-101 [77] as feature extractor can only provide very limited improvements for the detection performance on the proposed dataset. As shown in Table 3.4, the ResNet-101 backbone only gain only 0.3% overall AP improvements compared to ResNet-50 model, while ResNext-101 achieved approximately 1% further improvements. Moreover, the RetinaNet with ResNet-101 even perform worse than the ResNet-50 model. We suppose it is because that the receptive fields are extremely large in the very deep layers, and features extracted by these layers are more suitable for localising larger targets but not tiny ones. Hence, we did not equip deeper backbone for our detector but increase the input image size to obtain high resolution features. As shown in Table 3.4, simply up-sample the input image significantly improve detection accuracy on tiny and small targets.

Table 3.5 shows category related detection performance where $AP_h$ and $AP_v$ indicate performance for human and vehicle respectively. Since sample size for some fine grained categories such as truck and bus is extremely small, we only evaluate super categories (human and vehicle) at inference time. Consequently, we treat the dataset as a 3-class (include background) detection problem in practical. As shown in the results, one-stage methods obtained worse performance due to lower input resolution, but it is interesting to note that all of the single-stage methods perform better for vehicles rather than humans while two-stage methods go to an opposite case.

TABLE 3.4: Quantitative Results

| Detector | Backbone | Traininput | Testinput | AP | $AP_t$ | $AP_s$ | $AP_m$ | AR | $AR_t$ | $AR_s$ | $AR_m$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Two-stage:** | | | | | | | | | | | |
| Faster R-CNN-R50 | ResNet-50 | ∼1333×800 | ∼1333×800 | 0.291 | 0.125 | 0.303 | 0.500 | 0.372 | 0.185 | 0.387 | 0.593 |
| Faster R-CNN-R101 | ResNet-101 | ∼1333×800 | ∼1333×800 | 0.294 | 0.136 | 0.304 | 0.483 | 0.378 | 0.205 | 0.391 | 0.567 |
| Faster R-CNN-X101 | ResNeXt-101 | ∼1333×800 | ∼1333×800 | **0.305** | 0.142 | 0.311 | 0.519 | **0.380** | 0.182 | 0.393 | 0.599 |
| **One-stage:** | | | | | | | | | | | |
| YOLOv3-320 | DarkNet-53 | 320×320 | 320×320 | 0.063 | 0.002 | 0.055 | 0.248 | 0.120 | 0.011 | 0.089 | 0.319 |
| YOLOv3-416 | DarkNet-53 | 320×320 | 320×320 | 0.097 | 0.011 | 0.089 | 0.244 | 0.142 | 0.029 | 0.118 | 0.323 |
| YOLOv3-608 | DarkNet-53 | 608×608 | 608×608 | 0.162 | 0.048 | 0.169 | 0.322 | 0.236 | 0.093 | 0.232 | 0.423 |
| SSD-V16 | VGG-16 | 512×512 | 512×512 | 0.072 | 0.002 | 0.037 | 0.246 | 0.155 | 0.028 | 0.129 | 0.363 |
| SSD-R50 | ResNet-50 | 512×512 | 512×512 | 0.063 | 0.003 | 0.019 | 0.264 | 0.125 | 0.011 | 0.079 | 0.346 |
| RetinaNet-R50 | ResNet-50 | ∼1333×800 | ∼1333×800 | **0.192** | 0.023 | 0.214 | 0.404 | **0.306** | 0.074 | 0.340 | 0.536 |
| RetinaNet-R101 | ResNet-101 | ∼1333×800 | ∼1333×800 | 0.185 | 0.021 | 0.191 | 0.388 | 0.286 | 0.062 | 0.308 | 0.513 |
| **Ours:** | | | | | | | | | | | |
| Faster R-CNN-ours-1x | ResNet-50 | ∼1333×800 | ∼1333×800 | 0.313 | 0.123 | 0.341 | 0.505 | 0.388 | 0.186 | 0.416 | 0.588 |
| Faster R-CNN-ours-1.25x | ResNet-50 | ∼1666×1000 | ∼1666×1000 | 0.320 | 0.154 | 0.328 | 0.518 | 0.391 | 0.212 | 0.397 | 0.602 |
| Faster R-CNN-ours-1.5x | ResNet-50 | ∼1999×1200 | ∼1999×1200 | **0.352** | 0.205 | 0.358 | 0.533 | **0.423** | 0.285 | 0.429 | 0.605 |

TABLE 3.5: Category detection performance

| Detectors | $AP_h(\%)$ | $AP_v(\%)$ |
|---|---|---|
| Faster R-CNN-R50 | 28.7 | 28.4 |
| Faster R-CNN-R101 | 31.1 | 27.7 |
| Faster R-CNN-X101 | **32.4** | **28.7** |
| YOLOv3-320 | 5.4 | 7.2 |
| YOLOv3-416 | 8.8 | 10.5 |
| YOLOv3-608 | 13.5 | 18.8 |
| SSD-V16 | 4.0 | 10.4 |
| SSD-R50 | 5.3 | 7.3 |
| RetinaNet-R50 | **16.8** | **21.6** |
| RetinaNet-R101 | 16.7 | 20.3 |
| Faster R-CNN-R50-ours-1x | 32.0 | 30.7 |
| Faster R-CNN-R101-ours-1.25x | 32.4 | 31.6 |
| Faster R-CNN-X101-ours-1.5x | 34.8 | 35.5 |

**Ablation study:** To further evaluate the effectiveness of motion channels, ablation study was conducted as shown in Table 3.6. With base motion feature, the overall AP obtained a limited improvement from 0.291 to 0.294, however, AR dropped from 0.372 to 0.365. We conjecture this is because that although motion channels enhance the network to focus on moving objects, still targets like standing persons and parked vehicles are missed. By adding shifted motion channels, the AP was significantly improved to 0.313 while the AR was also increased.

TABLE 3.6: Ablation study on the proposed dataset

| | RGB | $\mathcal{M}_{base}$ | $\mathcal{M}_{shift}$ | AP | AR |
|---|---|---|---|---|---|
| Plain | ✓ | | | 0.291 | 0.372 |
| w/ Base Motion | ✓ | ✓ | | 0.294 | 0.365 |
| w/ Shifted Motion | ✓ | ✓ | ✓ | 0.313 | 0.388 |

### 3.4.4 Qualitative results

To further explore the effectiveness of the proposed method, visualisation of the detection results are shown in Fig. 3.9. It is obvious that both baseline method and our proposed method can accurately localise those objects which are of large sizes. However, our methods achieve higher AP for tiny objects.

### 3.4.5 Computational cost analysis

As shown in Table 3.7, we compare the inference time between baseline detector and ours [2]. $T_m$ and $T_r$ are average model inference time and total running time per image on our proposed dataset respectively. Different from some previous works [54, 62, 76, 98], which introduced extra parameters to learn the external features, and led to a computational overhead. Our proposed approach does not improve the computational cost heavily while the external motion channel features can also be obtained efficiently.

TABLE 3.7: Computational Cost

| Methods | $T_m$ | $T_r$ |
|---|---|---|
| Faster R-CNN-R50-baseline | 0.15s | 0.19s |
| Faster R-CNN-R50-ours-1x | 0.18s | 0.29s |
| Faster R-CNN-R50-ours-1.25x | 0.23s | 0.36s |
| Faster R-CNN-R50-ours-1.5x | 0.28s | 0.40s |

## 3.5 Conclusion

In this paper, a new dataset namely SHV was introduced, objects of two super categories, *i.e.*, “human" and “vehicles" are annotated. Compared to the widely-used datasets, SHV provides much more “small" and “tiny" object instances, while only less than 1% “large" objects are included. Besides, the average density of objects is also higher than most other datasets. Therefore, based on these features, SHV can be treated as a benchmark for evaluating the ability of detectors in recognising tiny/crowded targets. Moreover, as the images in the dataset are continuous frames from fixed angle camera records, it can also be used for motion analysis and video detection task.

Except the dataset, we also proposed a baseline method for detecting small humans and vehicles in fixed camera angle videos. External channel features include base motion and shifted motion maps were employed to boost the original RGB input, which can be simply considered as a type of attention mechanism that guide the neural networks to pay more attention to moving objects. Moreover, We jointly trained the Faster R-CNN with an external

[2]Both methods are tested with Nvidia Tesla M40 GPU and Intel Xeon E5-2667 CPU.

branch, which is supervised by generated masks. Pixel-wise information is incorporated into the backbone network, which can guide the convolutional neural networks to learn more discriminative representations.

Some other popular object detectors were also evaluated on the proposed dataset, including SSD, YOLO, RetinaNet and etc., and we found that one-stage methods can achieve comparable accuracy on “medium" and “large" objects. However, performance gap between one-stage methods and two-stage methods on “small" and “tiny" targets still exist.

FIGURE 3.8: Preview of the SHV dataset, the green and blue bounding boxes represent the annotated objects of “human" and “vehicle" super category respectively (ignore regions are not shown here). It should be note that, the images included in the dataset have a 960×540 resolution, here we have resized the images to a very low resolution for preview.

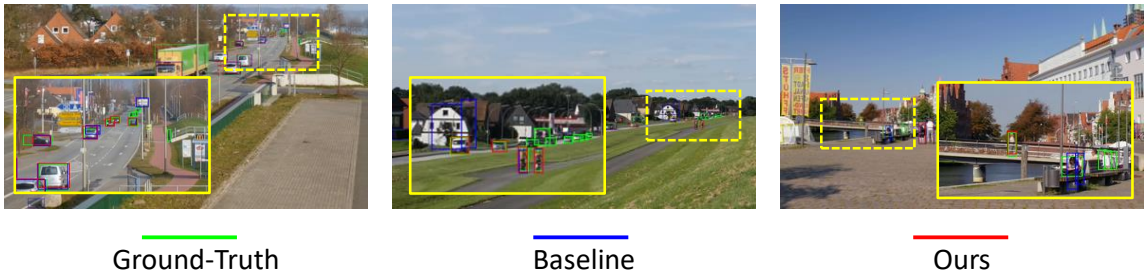
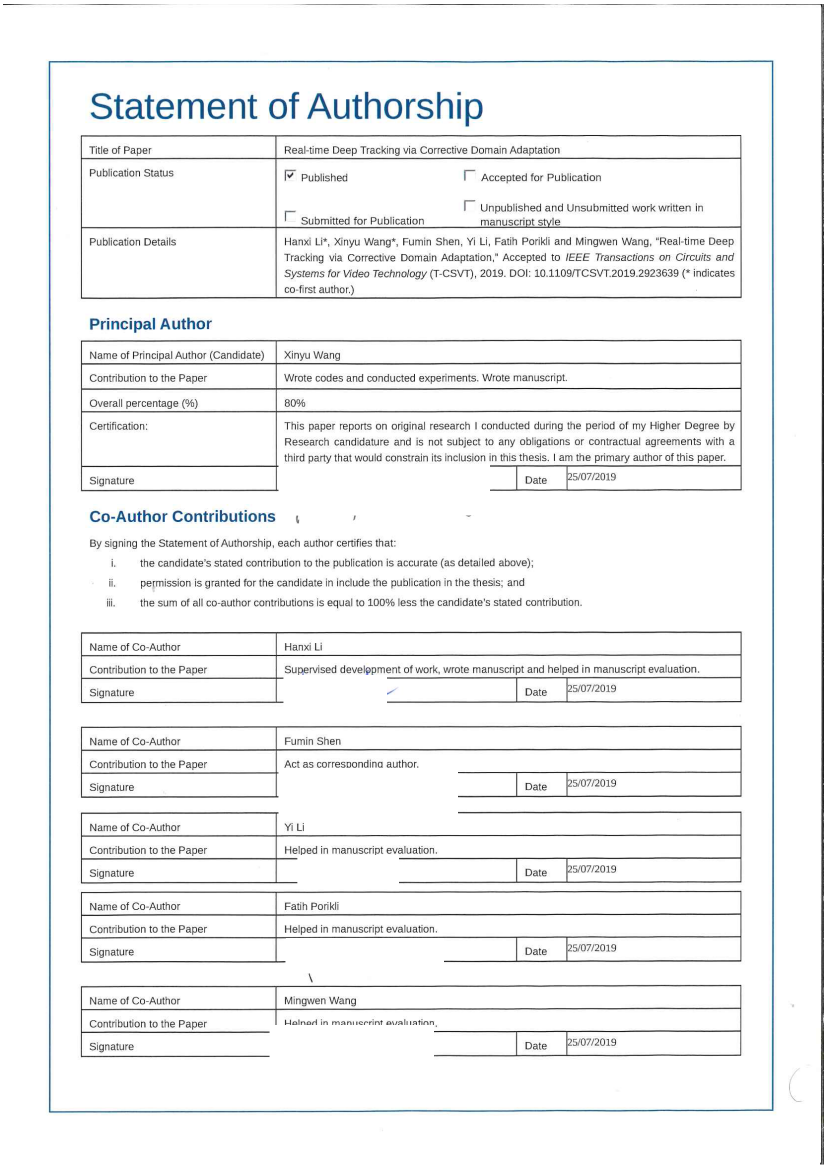



FIGURE 3.9: Qualitative results on the proposed SHV dataset (best viewed in color), visualisation threshold was set to 0.75. Bounding boxes showed in green, blue and red represent ground-truth annotation, detection results of baseline Faster R-CNN-R50 and the proposed Faster R-CNN-R-50-ours respectively.

CHAPTER 4

# Real-time Deep Tracking via Corrective Domain Adaptation

---



## Statement of Authorship

| | |
|---|---|
| Title of Paper | Real-time Deep Tracking via Corrective Domain Adaptation |
| Publication Status | ☑ Published ☐ Accepted for Publication ☐ Submitted for Publication ☐ Unpublished and Unsubmitted work written in manuscript style |
| Publication Details | Hanxi Li*, Xinyu Wang*, Fumin Shen, Yi Li, Fatih Porikli and Mingwen Wang, "Real-time Deep Tracking via Corrective Domain Adaptation," Accepted to *IEEE Transactions on Circuits and Systems for Video Technology* (T-CSVT), 2019. DOI: 10.1109/TCSVT.2019.2923639 (* indicates co-first author.) |

### Principal Author

| | |
|---|---|
| Name of Principal Author (Candidate) | Xinyu Wang |
| Contribution to the Paper | Wrote codes and conducted experiments. Wrote manuscript. |
| Overall percentage (%) | 80% |
| Certification: | This paper reports on original research I conducted during the period of my Higher Degree by Research candidature and is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in this thesis. I am the primary author of this paper. |
| Signature | Date 25/07/2019 |

### Co-Author Contributions

By signing the Statement of Authorship, each author certifies that:

i. the candidate's stated contribution to the publication is accurate (as detailed above);
ii. permission is granted for the candidate in include the publication in the thesis; and
iii. the sum of all co-author contributions is equal to 100% less the candidate's stated contribution.

| | |
|---|---|
| Name of Co-Author | Hanxi Li |
| Contribution to the Paper | Supervised development of work, wrote manuscript and helped in manuscript evaluation. |
| Signature | Date 25/07/2019 |

| | |
|---|---|
| Name of Co-Author | Fumin Shen |
| Contribution to the Paper | Act as corresponding author. |
| Signature | Date 25/07/2019 |

| | |
|---|---|
| Name of Co-Author | Yi Li |
| Contribution to the Paper | Helped in manuscript evaluation. |
| Signature | Date 25/07/2019 |

| | |
|---|---|
| Name of Co-Author | Fatih Porikli |
| Contribution to the Paper | Helped in manuscript evaluation. |
| Signature | Date 25/07/2019 |

| | |
|---|---|
| Name of Co-Author | Mingwen Wang |
| Contribution to the Paper | Helped in manuscript evaluation. |
| Signature | Date 25/07/2019 |

FIGURE 4.1: Statement of Authorship for Paper "Real-time Deep Tracking via Corrective Domain Adaptation"

## 4.1 Introduction

Visual tracking is one of the fundamental computer vision tasks. During the last decade, as the surge of deep learning, more and more tracking algorithms benefit from deep neural networks, e.g. Convolutional Neural Networks [42, 44] and Recurrent Neural Networks [102, 103]. Despite the well-admitted success, a dilemma still existing in the community is that, deep learning increases the tracking accuracy, while at the cost of high computational complexity. As a result, most well-performing deep trackers usually suffer from low efficiency [7, 9]. In recent years, some real-time deep trackers were proposed [5, 6, 45]. They achieved very fast tracking speed, but can not beat the shallow methods in some important evaluations, as we illustrate later.

In this paper, a simple yet effective domain adaptation algorithm is proposed. The equipped tracking algorithm, termed Corrective Domain Adaptation (CODA), transfers the features from the classification domain to the tracking domain, where the individual objects, rather than the image categories, are used as the learning samples. The advantage of the proposed domain adaptation is three-fold. Firstly, the shallow visual tracker, *e.g.*, the KCF algorithm employed in this work, can extract more informative deep features from the transferred feature space. Secondly, the adaptation could be also viewed as a dimension-reduction process that removes the redundant information for tracking, and more importantly, reduces the channel number of the deep feature significantly. This leads to a remarkable increase on tracking speed. Last but not least, the adaptation introduces small auxiliary CNN "branches" that could seamlessly correct the predictions of the shallow visual trackers. Inspired by the successful adoption of objectness in visual tracking [43, 104], we exploit the category information of the tracking target in CODA, in a relatively natural way. For a certain object category, the CNN "branches" are fine-tuned to correct the tracking boxes, and thus higher tracking accuracies are obtained.

The experiments show that the proposed CODA algorithm runs in around 35 FPS while achieves comparable tracking accuracy to the state-of-the-art trackers. Furthermore, given the

category information of the tracking target, the corrective CNN branches lead to a significant boost in tracking accuracy while keep the tracking speed nearly unchanged.

## 4.2 Related Work

### 4.2.1 Deep Trackers

Similar to other fields of computer vision, in recent years, more and more state-of-the-art visual trackers are deep-learning based. [42] is a well-known pioneering work that learns deep features for visual tracking. The DeepTrack method [43, 44] learns a deep model from scratch at the first frame and then updates it online. [105, 106] adopt the similar learning strategy, *i.e.*, learning the deep model offline with a large number of images while updating it online for the current video sequence. [107] achieves real-time speed via replacing the slow model update with a fast inference process.

The HCF tracker [7] extracts hierarchical convolutional features from the VGG-19 network [57], then put the features into correlation filters to regress the respond map. It can be considered as a combination between deep learning and the fast shallow tracker based on correlation filters. It achieves high tracking accuracy while the speed is around 10 fps.

Hyeonseob Nam *et al.* proposed to pre-train deep CNNs in multi domains, with each domain corresponding to one training video sequence [9]. The authors claim that there exist some common properties that are desirable for target representations in all domains such as illumination changes. To extract these common features, the authors separate domain-independent information from domain-specific layers. The yielded tracker, termed MD-net, achieves excellent tracking performance while the tracking speed is only 1 fps.

### 4.2.2 Real-time Deep Trackers

In recent years, some real-time deep trackers have also been proposed. [107] propose to infer the target location based on the deep features extracted from a fixed CNN model. Without

updating the CNN model, it achieves real-time speed. In [5], David Held *et al.* learn a deep regressor that can predict the location of the current object based on its appearance in the last frame. The tracker obtains a much faster tracking speed (over 100 fps) comparing to conventional deep trackers. Similarly, in [6] a fully-convolutional siamese network is learned to match the object template in the current frame. It also achieves real-time speed. Even though these real-time deep trackers also illustrate high tracking accuracy, there is still a clear performance gap between them and the state-of-the-art deep trackers. [108] discusses how different regularization terms of correlation filters essentially influence the tracking performance. The yielded variations of KCF tracker achieve higher tracking accuracy than the ordinary KCF, at the cost of speed reduction (from the speed over 100 fps of the original KCF to around 37 fps).

### 4.2.3 Deep tracking with objectness

Nearly all the deep trackers exploit the information of generic or specific object categories to achieve higher tracking accuracies. Most of the state-of-the-art deep trackers involve the objectness implicitly via pre-training the network off-line on the dataset with object categories and bounding-boxes. [9, 42, 106, 107, 109]. [104] designs a heuristic object proposal algorithm for eliminating the non-object tracking candidates and thus the tracker can hardly lose the target due to a misleading background patch. While most of the methods mainly focus on the generic objectness, [43] pay more attention to the specific object categories. By pre-training the CNN model with the object samples from a certain category, *e.g.*, human faces, the DeepTrack algorithm performs more robust for the specific object type.

## 4.3 Proposed Domain Adaptation

### 4.3.1 Network Structure

The proposed work is developed based on the HCF [7] tracking algorithm which is one of pioneering work in semi-deep trackers. In HCF, deep features are firstly extracted from

multiple layers of the VGG-19 network [57], and a set of KCF [8] trackers are carried out on those features, respectively. The final tracking prediction is obtained in a weighted voting manner. Following the setting in [7], we also extract the deep features from $conv3\_5$, $conv4\_5$ and $conv5\_5$ network layers of the VGG-19 model. However, the VGG-19 network is pre-trained using the ILSVRC dataset [110] for image classification, where the learning algorithm usually focus on the object categories. This is different from visual tracking tasks, where the individual objects are distinguished from other ones (even those from the same category) and the background. Intuitively, it is better to transfer the classification features into the visual tracking domain.

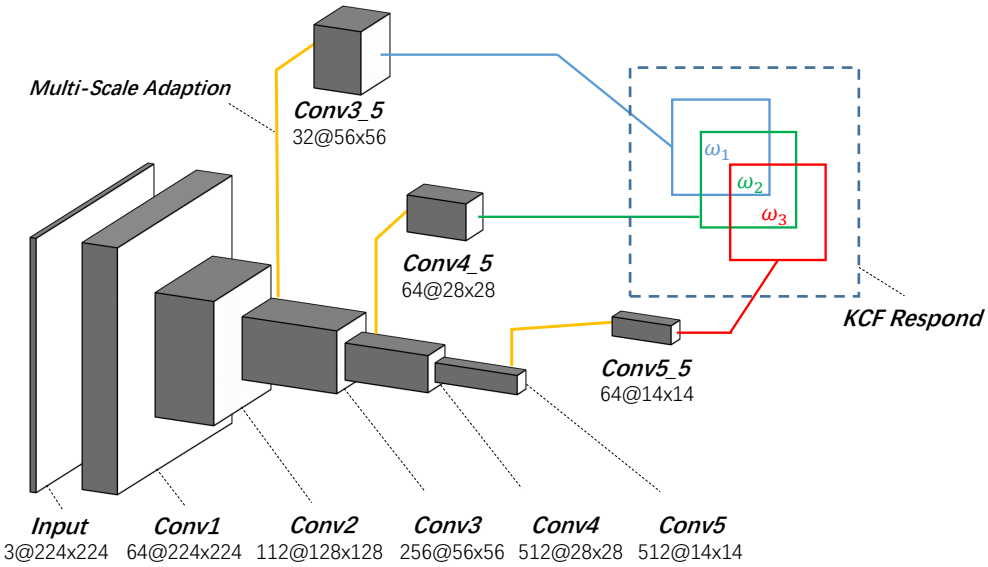



FIGURE 4.2: The network structure of the proposed CODA tracker. Three layers, namely, $conv3\_5$, $conv4\_5$ and $conv5\_5$ are selected as feature source. The domain adaption (as shown in yellow lines) reduces the channel number by $8$ times and keeps feature map size unchanged. Better viewed in color.

In this work, we propose to perform the domain adaptation in a simple way. A "tracking branch" is "grafted" onto each feature layer, as shown in Fig. 4.7. The tracking branch is actually a convolution layer which reduces the channel number by $8$ times and keeps feature map size unchanged. The convolution layer is then learned via minimizing the loss function tailored for tracking, as introduced below.

### 4.3.2 Learn the Domain Adaptation

The parameters in the aforementioned tracking branch is learned following a similar manner as Single Shot MultiBox Detector (SSD), a state-of-the-art detection algorithm [17]. When training, the original layers of VGG-19 (*i.e.* those ones before $conv\_5$ are fixed and each "tracking branch" is trained independently) The flowchart of the learning procedure for one tracking branch (based on $conv3\_4$) is illustrated in upper row of Figure 4.3, comparing with the learning strategy of MD-net [9] (the bottom row). To obtain a completed training circle, the adapted feature in $conv3\_5$ is used to regress the objects' locations and their objectness scores (shown in the dashed block). Please note that the deep learning stage in this work is purely offline and the additional part in the dashed block will be abandoned for generic object tracking. For specific categories, we propose to utilize the "tracking branches" for correcting the initial tracking boxes.

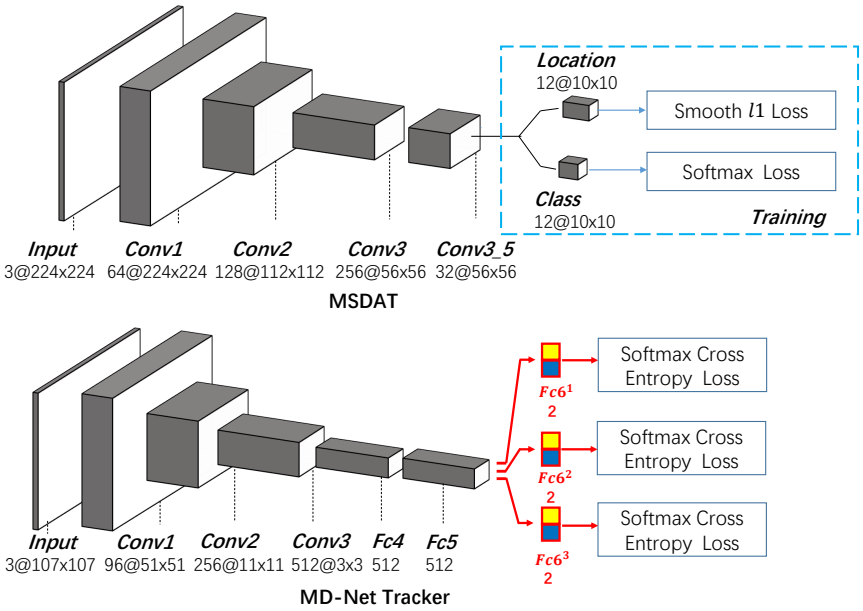



FIGURE 4.3: The flow-charts of the training process of CODA and MD-net. Note that the network parts inside the dashed blocks are only used for training and will be abandoned before tracking. Better viewed in color.

In SSD, a number of "default boxes" are generated for regressing the object rectangles. Furthermore, to accommodate the objects in different scales and shapes, the default boxes

also vary in size and aspect ratios. Let $m_{i,j} \in \{1, 0\}$ be an indicator for matching the $i$-th default box to the $j$-th ground truth box. The loss function of SSD writes:

$$L(m, c, l, g) = \frac{1}{N} \left( L_{conf}(m, c) + \alpha L_{loc}(m, l, g) \right) \tag{4.1}$$

where $c$ is the category of the default box, $l$ is the predicted bounding-box while $g$ is the ground-truth of the object box, if applicable. For the $j$-th default box and the $i$-th ground-truth, the location loss $L_{loc}^{i,j}$ is calculated as:

$$L_{loc}^{i,j}(l, g) = \sum_{u \in \{x,y,w,h\}} m_{i,j} \cdot \text{smooth}_{L_1}(l_i^u - \hat{g}_j^u) \tag{4.2}$$

where $\hat{g}^u, u \in \{x, y, w, h\}$ is one of the geometry parameter of normalized ground-truth box.

However, the task of visual tracking differs from detection significantly. We thus tailor the loss function for the KCF algorithm, where both the object size and the KCF window size are fixed. Recall that, the KCF window plays a similar role as default boxes in SSD [8], we then only need to generate one type of default boxes and the location loss $L_{loc}^{i,j}(l, g)$ is simplified as:

$$L_{loc}^{i,j}(l, g) = \sum_{u \in \{x,y\}} m_{i,j} \cdot \text{smooth}_{L_1}(l_i^u - g_j^u) \tag{4.3}$$

In other words, only the displacement $\{x, y\}$ is taken into consideration and there is no need for ground-truth box normalization.

Note that the concept of domain adaptation in this work is different from that defined in MD-net [9], where different video sequences are treated as different domains and thus multiple fully-connected layers are learned to handle them (see Figure 4.3). This is mainly because in MD-net samples the training instances in a sliding-window manner. An object labeled negative in one domain could be selected as a positive sample in another domain. Given the training video number is $C$ and the dimension of the last convolution layer is $d_c$, the MD-net learns $C$ independent $d_c \times 2$ fully-connected alternatively using $C$ soft-max losses, *i.e.*,

$$\mathcal{M}_{fc}^{i} : \mathbb{R}^{d_c} \rightarrow \mathbb{R}^2, \forall i = 1, 2, \ldots, C \tag{4.4}$$

where $\mathcal{M}_{fc}^{i}, \forall i \in \{1, 2, \ldots, C\}$ denotes the $C$ fully-connected layers that transferring the common visual domain to the individual object domain, as shown in Figure 4.3.

Differing from the MD-net, the domain in this work refers to a general visual tracking domain, or more specifically, the KCF domain. It is designed to mimic the KCF input in visual tracking (see Figure 4.3). In this domain, different tracking targets are treated as one category, *i.e.*, objects. When training, the object's location and confidence (with respect to the objectness) are regressed to minimize the smoothed $l_1$ loss. Mathematically, we learn a single mapping function $\mathcal{M}_{conv}(\cdot)$ as:

$$\mathcal{M}_{conv} : \mathbb{R}^{d_c} \rightarrow \mathbb{R}^4 \tag{4.5}$$

where the $\mathbb{R}^4$ space is composed of one $\mathbb{R}^2$ space for displacement $\{x, y\}$ and one label space $\mathbb{R}^2$.

Compared with Equation 4.4, the training complexity in Equation 4.5 decreases and the corresponding convergence becomes more stable. Our experiment proves the validity of the proposed domain adaptation approach.

### 4.3.3 Multi-scale Domain Adaptation

As introduced above, the domain adaption in our CODA method is essentially a convolution layer. To design the layer, an immediate question is how to select a proper size for the filters. According to Figure 4.7, the feature maps from different layers vary in size significantly. It is hard to find a optimal filer size for all the feature layers. Inspired by the success of Inception network [12], we propose to simultaneously learn the adaptation filters in different scales. The response maps with different filter sizes are then concatenated accordingly, as shown in Figure 4.4. In this way, the input of the KCF tracker involves the deep features from different scales.

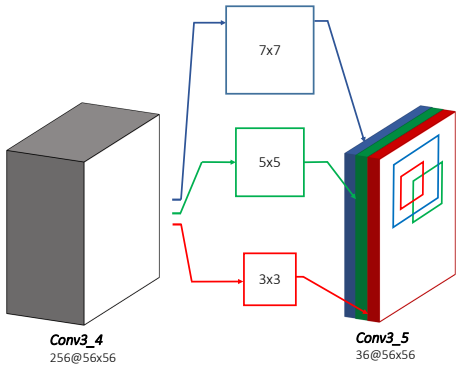



FIGURE 4.4: Learn the adaptation layer using three different types of filters

In practice, we use $3 \times 3$ and $5 \times 5$ filters for all the three feature layers. Given the original channel number is $K$, each type of filter generate $\frac{K}{16}$ channels and thus the channel reduction ratio is still $8:1$. With the channel reduction, the tracking speed increased significantly. It is easy to see that the speed of KCF tracker drops dramatically as the channel number increase. In this work, after the adaptation, the channel number is shrunk by $8$ times which accelerates the tracker by $2$ to $2.5$ times.

## 4.4 Tracking with Objectness

### 4.4.1 A Long-standing Ambiguity in Visual Tracking

Despite the widespread real-world usages, visual tracking is still criticized as less well-posed compared with other tasks with clearly-defined targets, such as object detection and semantic segmentation. In visual tracking, the only reliable target information is given at the first frame while the information could be ambiguous or misleading in many circumstances. For example, in Figure 4.5, a car is to be tracked in the sequence. From the viewing angle at the first frame, only the car back can be observed so it is defined as the "target" by the blue bounding box. Nonetheless, this simple target definition usually leads to an ambiguity: when the target pose changes significantly, it is hard to evaluate tracking results. In specific, as shown in Figure 4.5, either the yellow box or the blue box can be considered as a "perfect" tracking, depending on what exactly the tracking target is, the car back or the whole car.

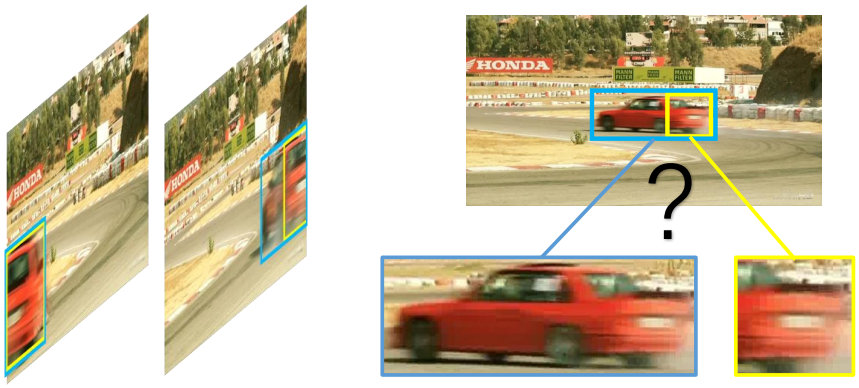



FIGURE 4.5: The commonly existing ambiguity in visual tracking. From left to right, the car back is labeled as the tracking target at the first frame, as the viewing angle changes, the car back and the visible part of the car become more and more different. Finally, when the pose changes significantly, as shown in the right column, it is hard to judge which bounding box (among blue and yellow ones) is the better tracking result.

Unfortunately, a clearly-defined tracking target is usually absent in visual tracking due to the very limited information, namely, a bounding box, given at the first frame. In this work, we try to address the ill-posed problem via imposing the object category in visual tracking tasks. In other words, the tracker tracks the object given the target's bounding box at the first frame as well as the category of the target. This assumption is similar to the original DeepTrack algorithm [43] while we exploit the object information in a easier yet more effective way.

### 4.4.2 Corrective Domain Adaptation

Given the specific target category, we naturally use the proposed learning strategy proposed in Section 4.3.2 to learn a set of CNN "branches" on the samples from this category and then use the "branches" for correcting the prediction of the deep tracker. The high-level concept of the "tracking-detection-fusion" is illustrated in Figure 4.2

From Figure 4.2 one can see that the CNN model is essentially the same to that in Figure 4.7 expect that the auxiliary CNN branches are used for regressing the object bounding box. Note that all the regression branches are not computationally complex compared with the whole network, the extra computation burden is not heavy.

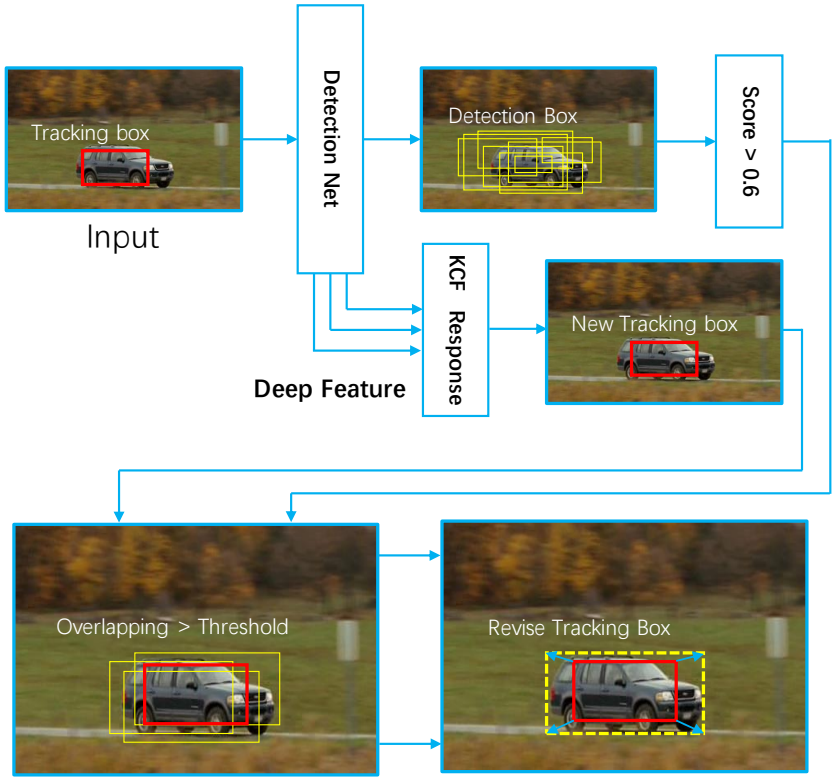



FIGURE 4.6: The flowchart of the detection-guided tracking process. Top: the tracking box (shown in red) is obtained following the same strategy as HCF. Meanwhile, some detection bounding boxes are also generated by SSD. Bottom: after removing the unqualified detection bounding boxes, the average scale and aspect ratio of the detection results are used to correct the current tracking box. Better view in color.

### 4.4.3 A Simple Yet Effective Guidance from Detector

Given the tracking bounding-box and detection bounding-boxes, CODA merges the results in a simple yet effective way. Figure 4.6 demonstrates the merging process. Specifically, let us assume the tracking bounding-box (red bounding-box obtained in the same way as the ordinary HCF tracker) is represented as a 4-D vector $\mathbf{B}_t = [x_t, y_t, w_t, h_t] \in \mathbb{R}^{4\times 1}$ where $x_t$, $y_t$, $w_t$ and $h_t$ are the $x$-axis coordinate of the box center, the $y$-axis coordinate of the box center, the width and the height of the tracking box, respectively. The SSD detector generates multiple detection bonding-boxes stored in the set $\mathbb{B}_d = \{\mathbf{B}_d^1, \mathbf{B}_d^2, \ldots \mathbf{B}_d^N\}$ with the SSD scores $\{s_d^1, s_d^2, \ldots, s_d^N\}$. As shown in Figure 4.6, we firstly remove some unqualified

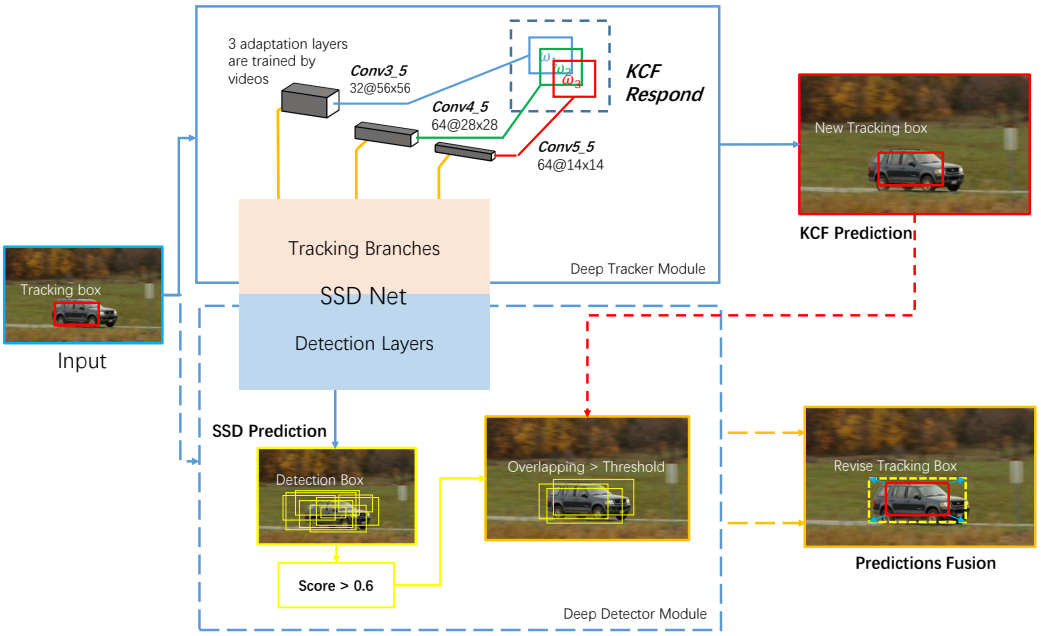



FIGURE 4.7: For a specific target object. CODA extracts features from $conv3\_3$, $conv4\_3$ and $conv5\_3$ for KCF tracking and extracts features from other 6 layers for SSD regressing the object bounding-box. The predictions of the KCFs and the detection regressors are then merged for more robust tracking results.

detection boxes that are far away from the tracking box or with low scores. Normally, the qualified detection box set is selected as:

$$\mathbb{B}'_d = \{\forall \mathbf{B}^i_d \mid \text{IoU}(\mathbf{B}^i_d, \mathbf{B}_t) > 0.5 \;\&\; s^i_d > 0.6\} \tag{4.6}$$

where the function $\text{IoU}(\mathbf{B}_1, \mathbf{B}_2)$ stands for the "Intersection over Union" of two bounding boxes $\mathbf{B}_1$ and $\mathbf{B}_2$, which is used for evaluate their overlapping state.

We use $a_t = \sqrt{w_t \cdot h_t}$ and $r_t = w_t/h_t$ to represent the scale and aspect ratio of the tracking box. Suppose the number of qualified detection boxes is $N_q$, we calculate the average scale and aspect ratio for the qualified detection boxes as:

$$\bar{a}_d = \frac{1}{N_q} \sum_{\mathbf{B}_d^i \in \mathbb{B}'_d} a_d^i \tag{4.7}$$

$$\bar{r}_d = \frac{1}{N_q} \sum_{\mathbf{B}_d^i \in \mathbb{B}'_d} r_d^i \tag{4.8}$$

Then the scale and aspect ratio of the final prediction, *i.e.*, $a_t^*$ and $r_t^*$ are given by:

$$a_t^* = \left(1 - \frac{1}{1+\exp(-\lambda(s_d^* - s_0))}\right) \cdot a_t + \frac{1}{1+\exp(-\lambda(s_d^* - s_0))} \cdot \bar{a}_d \tag{4.9}$$

$$r_t^* = \left(1 - \frac{1}{1+\exp(-\lambda(s_d^* - s_0))}\right) \cdot r_t + \frac{1}{1+\exp(-\lambda(s_d^* - s_0))} \cdot \bar{r}_d \tag{4.10}$$

where $s_d^* = \max([s_d^1, s_d^2, \ldots, s_d^{N_q}])$, *i.e.*, the max scores over the qualified detection boxes. The hyper-parameters $\lambda$ and $s_0$ are set to $10$ and $0.6$ in practice.

Finally, the predicted bounding-box of CODA writes:

$$\mathbf{B}_t^* = \left[x_t, y_t, \frac{w_t \cdot a_t^*}{a_t}, \frac{w_t \cdot a_t^*}{a_t \cdot r_t^*}\right]. \tag{4.11}$$

From Equation 4.11 and Figure 4.6 one can see the original HCF tracking box is corrected by the detection boxes. We found the correction is usually beneficial thanks to the more clear definition of the target category and the well-learned detector. To make the corrective adaptation more clear for readers, we summarize the whole process in Algorithm 1.

## 4.5 Experiment

### 4.5.1 Experiment Overview

In this section, we evaluate the proposed CODA tracker in two scenarios. First, the CODA for generic objects in which the corrective CNN branches are abandoned. And second, the CODA for specific target categories. The experiment is conducted on several well-adopted

**Algorithm 1** Corrective Domain Adaptation Tracker (CODA) Algorithm

**Input:** Pre-trained CNN network $N$, video sequence $S$, init bbox $p$

$p_i = p$
$ovp = \varnothing$
**for each** $i \in [1, f]$ **do**
  $I_i = next(S)$; ▷ Get current frame
  **if** $i < I_{warm}$ **or** $sum(ovp) > t$ **then**
    $feat_i, boxes_i, scores_i = forward(I_i, N)$ ▷ Feed-forward
    $boxes_i = filtering(boxes_i, scores_i, \theta)$ ▷ Filter boxes
  **else**
    $feat_i = forward(I_i, N^{'})$ ▷ Forward without fully connected layer
  **end if**
  **if** $i$ = 1 **then**
    $M_{kcf} = init(feat_i)$ ▷ Init KCF model
  **else**
    $p_i = predict(M_{kcf}, feat_i)$ ▷ Get predicted box
    **if** $boxes_i$ **then**
      $ovp_i = overlapping(boxes_i, p_i)$
      **if** $ovp_i > threshold$ **then**
        $p_i = merging(p_i, boxes_i)$ ▷ Predictions Fusion
      **end if**
    **end if**
  **end if**
  **if** $i > 1$ **and** $|p_i - p_{i-1}| > \tau$ **then** ▷ Lazy update strategy
    $M_{kcf} = update(M_{kcf}, feat_i)$
  **end if**
**end for each**

datasets and compared with some state-of-the-art trackers, especially the recently proposed real-time deep trackers.

The proposed CODA tracker is based on a VGG-19 network [57] which is initialized using the ILSVRC classification dataset, and then trained 3 domain adaptation layers which transfer the deep features from classification domain to tracking domain. All the experiment is implemented in MATLAB with matcaffe [111] deep learning interface, on a computer equipped with a Intel i7 4770K CPU, a NVIDIA GTX1070 graphic card and 32G RAM.

### 4.5.2 Experiment on Generic Objects

In this subsection, we report the tracking performances on generic objects of the proposed tracker and some state-of-the-art approaches. As this work focus on real-time or semi-real-time trackers, we compare our algorithm with HCF [7], GOTURN [5], KCF tracker [8], TGPR [39], Struck [37], MIL [32], TLD [34], SCM [112], MD-net [9] and SiameseFC [6]. As explained above, for generic objects, the corrective CNN branches are abandoned and only the KCF tracking results are used.

**OTB50.** The Object Tracking Benchmark 50 (OTB-50) [113] consists 50 video sequences and involves 51 tracking tasks. It is one of the most popular tracking benchmarks since the year 2013, The evaluation is based on two metrics: center location error and bounding box overlap ratio. The one-pass evaluation (OPE) is employed to compare our algorithm with the HCF [7], GOTURN [5], the Siamese tracker [6] and the afore mentioned shallow trackers.

The 3 domain adaptation layers of CODA are trained on 58 video sequences that collected from VOT2013 [114], VOT2014 [115] and VOT2015 [116], excluding the ones also include in OTB50. The result curves are shown in Figure 4.8

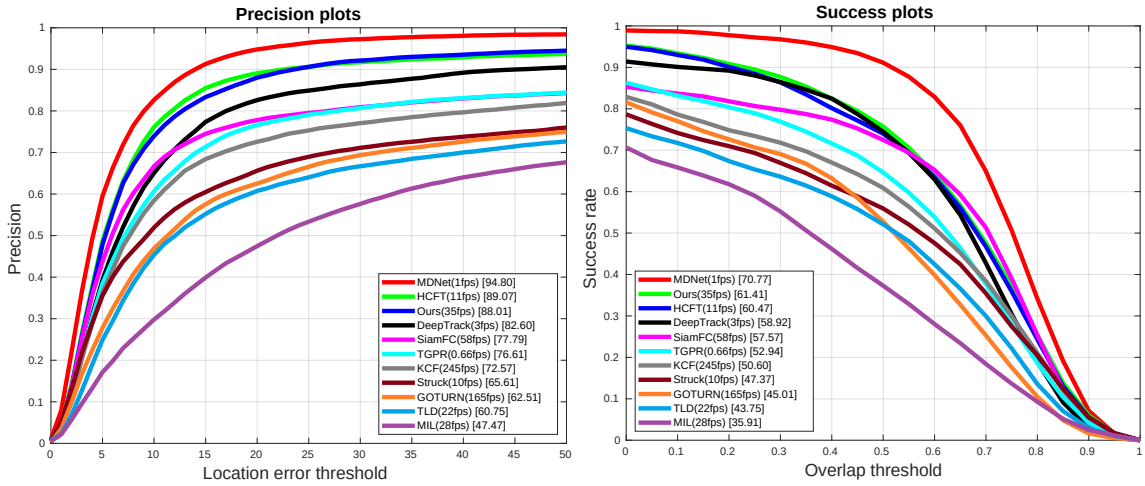



FIGURE 4.8: The location error plots and the overlapping accuracy plots of the involving trackers, tested on the OTB-50 dataset.

**OTB100.** The Object Tracking Benchmark 100 [117] is the extension of OTB-50 and contains 100 video sequences. We test our method under the same experiment protocol as OTB-50 and

comparing with all the aforementioned trackers. The training set of the CODA learning keeps the same to the experiment on OTB-50. The test results as well as the tracking speeds (in fps) are reported in Table 4.1

As can be seen in the table, the proposed CODA algorithm keeps its superiority over all the other real-time trackers and keeps the similar accuracy to HCF. The best-performing MD-net still enjoys a remarkable performance gap over all the other trackers while runs in around 1 fps. To further illustrate the comparison between the CODA tracker and other real-time trackers, Figure 4.9 shows the tracking results of the comparing real-time trackers on some key frames of 6 representative OTB-100 video sequences. As a reference, the HCF results are also depicted.

| Sequence | Ours | HCF | MD-Net | SiamFC | GOTURN | KCF | Struck | MIL | SCM | TLD |
|---|---|---|---|---|---|---|---|---|---|---|
| DP rate(%) | 83.0 | 83.7 | 90.9 | 75.2 | 56.39 | 69.2 | 63.5 | 43.9 | 57.2 | 59.2 |
| OS(AUC) | 0.567 | 0.562 | 0.678 | 0.561 | 0.424 | 0.475 | 0.459 | 0.331 | 0.445 | 0.424 |
| Speed(FPS) | 34.8 | 11.0 | 1 | 58 | 165 | 243 | 9.84 | 28.0 | 0.37 | 23.3 |

TABLE 4.1: Tracking accuracies and speeds (in fps) of the compared trackers on OTB-100

### 4.5.3 Experiment on Humans

We perform the same experiment on the pedestrian category to evaluate the proposed corrective framework with non-rigid objects. We select all the 37 pedestrian video sequences from OTB-100 as the test set and show the tracking performances of comparing trackers in Figure 4.10.

## 4.6 Conclusion

In this work, we propose a simple yet effective algorithm to transferring the features in the classification domain to the visual tracking domain. The yielded visual tracker, termed CODA, is real-time and achieves the comparable tracking accuracies to the state-of-the-art deep trackers. For a specific target category, CODA guides the visual tracking by the detection results. As the deep tracker and the deep detector share most part of the deep network, no much extra computation is required. Meanwhile, we can see a dramatic performance


FIGURE 4.9: Tracking results comparison on some key frames of 9 representative OTB-100 video sequences. The comparing methods include the proposed CODA tracker (green), GOTURN [5] (blue), Siamese tracker [6] (dashed yellow), HCF tracker [7] (dashed green) and the KCF algorithm [8] (dashed light blue). The red bounding boxes are the ground-truth locations of the tracking targets. Better view in color.

improvement in CODA, over its prototype, the HCF tracker. This improvement implies the absence of the target category could lead to poor tracking performance while to address this ambiguity in a sophisticated way could yield much better deep trackers.

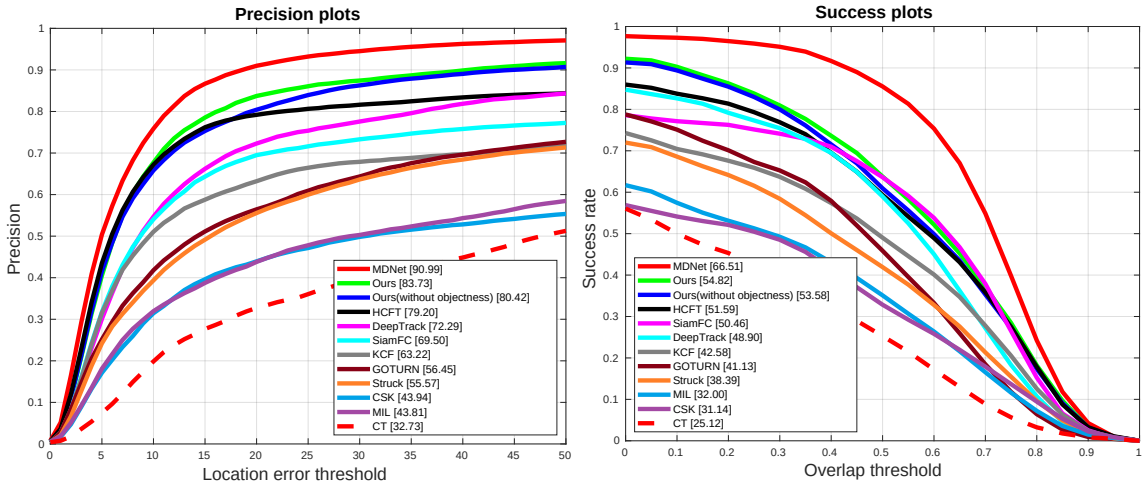



FIGURE 4.10: The location error plots and the overlapping accuracy plots tested on the "pedestrian subset" of OTB-100. The comparing methods including MD-Net [9], HCF [7], the Siamese Tracker [6], GOTURN [5], CODA (this paper) and the shallow trackers.

Admittedly, updating the neural network online can lift the tracking accuracy significantly [9, 44]. However, the existing online updating scheme results in significant speed reduction. One possible future direction could be to simultaneously update the KCF model and a certain part of the neural network (*e.g.* the last convolution layer). In this way, one could strike the balance between accuracy and efficiency and thus better tracker could be obtained.

Another possible direction is to involve more than one object categories in the corrective CNN branches for a certain type of tracking scenario. For instance, one can take pedestrian, car, bicycle and motorbike into consideration for road scene tracking. This could lead to even higher tracking robustness than the one-category CODA proposed in this paper.

CHAPTER 5

# Conclusion

This thesis has presented a study of two fundamental research topics in the computer vision community, *i.e.*, human detection and tracking. For computers, detection aims to localise a set of object candidates and classify them into a certain category while tracking aims to predict the target position based on the ground-truth information from the first frame in a video sequence.

For detection part, we have presented a simple method for improving human detectors with extra semantic features by aggregating the original RGB images with segmentation masks. We implement our method on two popular detection frameworks, *i.e.*, Faster R-CNN and SSD, and evaluate the proposed method on two datasets *i.e.*, MS-COCO Persons and Crowd-Human. Furthermore, we also introduce a new dataset for detecting small humans and vehicles in fixed angle camera videos namely SHV. At the same time, a baseline detector which exploits the motion channel features is proposed. We have empirically shown that the fusion of extra features is able to achieve more accurate and robust results of object detection. For future work, one possible direction is to employ neural network architecture search technology to search for backbone network automatically for the object detection task.

For tracking part, we propose a simple yet effective algorithm to transfer the features in the classification domain into the visual tracking domain. Moreover, we introduce the objectness into visual object tracking. For a specific target category, the visual tracker is guided by the detection results. A possible future direction is to involve more than one object categories in the corrective CNN branches for a certain type of tracking scenario. For instance, one can take pedestrian, car, bicycle and motorbike into consideration for road scene tracking. This

could lead to even higher tracking robustness than the one category method proposed in this paper.

# Bibliography


[1] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2017, pp. 2961–2969.

[2] S. Zhang, R. Benenson, and B. Schiele, "Citypersons: A diverse dataset for pedestrian detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2017, pp. 4457–4465.

[3] L. Wen, D. Du, Z. Cai, Z. Lei, M.-C. Chang, H. Qi, J. Lim, M.-H. Yang, and S. Lyu, "Ua-detrac: A new benchmark and protocol for multi-object detection and tracking," *arXiv preprint arXiv:1511.04136*, 2015.

[4] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2014, pp. 740–755.

[5] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," *Proc. Eur. Conf. Comp. Vis.*, 2016.

[6] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. Torr, "Fully-convolutional siamese networks for object tracking," in *Proc. Eur. Conf. Comp. Vis.*, 2016, pp. 850–865.

[7] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Hierarchical convolutional features for visual tracking," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3074–3082.

[8] J. F. Henriques, R. Caseiro, P. Martins, and J. Batista, "High-speed tracking with kernelized correlation filters," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 583–596, 2014.

[9] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016, pp. 4293–4302.

[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances in Neural Inf. Process. Syst.*, 2012,

pp. 1097–1105.

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Proc. Int. Conf. Learn. Representations*, 2015.

[12] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2015, pp. 1–9.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016, pp. 770–778.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2014, pp. 580–587.

[15] R. Girshick, "Fast r-cnn," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015, pp. 1440–1448.

[16] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Proc. Advances in Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2016, pp. 21–37.

[18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016, pp. 779–788.

[19] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "DSSD: Deconvolutional single shot detector," in *arXiv preprint arXiv:1701.06659*, 2016.

[20] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017, pp. 7263–7271.

[21] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2017, pp. 2980–2988.

[22] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018.

[23] Z. Tian, C. Shen, H. Chen, and T. He, "FCOS: Fully convolutional one-stage object detection," *arXiv preprint arXiv:1904.01355*, 2019.

[24] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vision*, vol. 104, no. 2, pp. 154–171, 2013.

[25] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features." in *Proc. IEEE Int. Conf. Comp. Vis.*, 1999, pp. 1150–1157.

[26] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[27] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. Eur. Conf. Comp. Vis.*, 2018, pp. 734–750.

[28] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," *arXiv preprint arXiv:1904.08189*, 2019.

[29] T. Kong, F. Sun, H. Liu, Y. Jiang, and J. Shi, "Foveabox: Beyond anchor-based object detector," *arXiv preprint arXiv:1904.03797*, 2019.

[30] X. Zhou, J. Zhuo, and P. Krähenbühl, "Bottom-up object detection by grouping extreme and center points," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2019.

[31] C. Zhu, Y. He, and M. Savvides, "Feature selective anchor-free module for single-shot object detection," *arXiv preprint arXiv:1903.00621*, 2019.

[32] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1619–1632, 2010.

[33] D. A. Ross, J. Lim, R.-S. Lin, and M.-H. Yang, "Incremental learning for robust visual tracking," *Int. J. Comput. Vision*, vol. 77, no. 1-3, pp. 125–141, 2008.

[34] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1409–1422, 2011.

[35] H. Li, C. Shen, and Q. Shi, "Real-time visual tracking using compressive sensing," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2011, pp. 1305–1312.

[36] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparse collaborative appearance model," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2356–2368, 2014.

[37] S. Hare, S. Golodetz, A. Saffari, V. Vineet, M.-M. Cheng, S. L. Hicks, and P. H. Torr, "Struck: Structured output tracking with kernels," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 10, pp. 2096–2109, 2015.

[38] J. Zhang, S. Ma, and S. Sclaroff, "Meem: robust tracking via multiple experts using entropy minimization," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2014, pp. 188–203.

[39] J. Gao, H. Ling, W. Hu, and J. Xing, "Transfer learning based visual tracking with gaussian processes regression," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2014, pp. 188–203.

[40] N. Wang, J. Shi, D.-Y. Yeung, and J. Jia, "Understanding and diagnosing visual tracking systems," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015, pp. 3101–3109.

[41] X. Li, W. Hu, C. Shen, Z. Zhang, A. Dick, and A. V. D. Hengel, "A survey of appearance models in visual object tracking," *ACM Trans. on Intell. Sys. and Tech.*, vol. 4, no. 4, p. 58, 2013.

[42] N. Wang and D.-Y. Yeung, "Learning a deep compact image representation for visual tracking," in *Proc. Advances in Neural Inf. Process. Syst.*, 2013, pp. 809–817.

[43] H. Li, Y. Li, F. Porikli, *et al.*, "Deeptrack: Learning discriminative feature representations by convolutional neural networks for visual tracking." in *Proc. British Machine Vis. Conf.*, vol. 1, no. 2, 2014, p. 3.

[44] H. Li, Y. Li, and F. Porikli, "Deeptrack: Learning discriminative feature representations online for robust visual tracking," *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1834–1848, 2015.

[45] X. Wang, H. Li, Y. Li, F. Shen, and F. Porikli, "Robust and real-time deep tracking via multi-scale domain adaptation," in *Proc. IEEE Int. Conf. Mutimedia & Expo*. IEEE, 2017, pp. 1338–1343.

[46] C. Ma, J.-B. Huang, X. Yang, and M.-H. Yang, "Robust visual tracking via hierarchical convolutional features," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2018.

[47] H. Li, X. Wang, F. Shen, Y. Li, F. Porikli, and M. Wang, "Real-time deep tracking via corrective domain adaptation," *IEEE Trans. Circuits Syst. Video Technol.*, 2019.

[48] B. Li, J. Yan, W. Wu, Z. Zhu, and X. Hu, "High performance visual tracking with siamese region proposal network," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018.

[49] Z. Zhu, Q. Wang, L. Bo, W. Wu, J. Yan, and W. Hu, "Distractor-aware siamese networks for visual object tracking," in *Proc. Eur. Conf. Comp. Vis.*, 2018.

[50] S. Shao, Z. Zhao, B. Li, T. Xiao, G. Yu, X. Zhang, and J. Sun, "Crowdhuman: A benchmark for detecting human in a crowd," *arXiv preprint arXiv:1805.00123*, 2018.

[51] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2005, pp. 886–893.

[52] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," in *Proc. IEEE Int. Conf. Comp. Vis.* IEEE, 2003, pp. 734–741.

[53] Q. Hu, P. Wang, C. Shen, A. van den Hengel, and F. Porikli, "Pushing the limits of deep cnns for pedestrian detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1358–1368, 2018.

[54] J. Mao, T. Xiao, Y. Jiang, and Z. Cao, "What can help pedestrian detection?" in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2017, pp. 6034–6043.

[55] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, "Repulsion loss: Detecting pedestrians in a crowd," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2018.

[56] W. Ouyang, H. Zhou, H. Li, Q. Li, J. Yan, and X. Wang, "Jointly learning deep features, deformable parts, occlusion and classification for pedestrian detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1874–1887, 2018.

[57] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[58] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2015, pp. 3431–3440.

[59] L. Zhang, L. Lin, X. Liang, and K. He, "Is faster r-cnn doing well for pedestrian detection?" in *Proc. Eur. Conf. Comp. Vis.* Springer, 2016, pp. 443–457.

[60] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast r-cnn for pedestrian detection," *IEEE Trans. Multimedia*, vol. 20, no. 4, pp. 985–996, 2018.

[61] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, "Learning rich features from rgb-d images for object detection and segmentation," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2014, pp. 345–360.

[62] L. Spinello and K. O. Arras, "People detection in rgb-d data," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots & Systems*. IEEE, 2011, pp. 3838–3843.

[63] D. Chen, S. Zhang, W. Ouyang, J. Yang, and Y. Tai, "Person search via a mask-guided two-stream cnn model," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2018, pp. 764–781.

[64] C. Song, Y. Huang, W. Ouyang, and L. Wang, "Mask-guided contrastive attention model for person re-identification," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2018, pp. 1179–1188.

[65] F. Wan, P. Wei, J. Jiao, Z. Han, and Q. Ye, "Min-entropy latent model for weakly supervised object detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2018, pp. 1297–1306.

[66] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[67] X. Du, M. El-Khamy, V. I. Morariu, J. Lee, and L. Davis, "Fused deep neural networks for efficient pedestrian detection," *arXiv preprint arXiv:1805.08688*, 2018.

[68] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2018, pp. 833–851.

[69] G. Lin, C. Shen, A. Van Den Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2016, pp. 3194–3203.

[70] G. Brazil, X. Yin, and X. Liu, "Illuminating pedestrians via simultaneous detection & segmentation," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2017, pp. 4950–4959.

[71] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[72] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2017, pp. 936–944.

[73] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp.

834–848, 2018.

[74] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2017, pp. 4438–4446.

[75] Y. Li, L. Liu, C. Shen, and A. van den Hengel, "Image co-localization by mimicking a good detector's confidence score distribution," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2016, pp. 19–34.

[76] X. Wang, C. Shen, H. Li, and S. Xu, "Human detection aided by deeply learned semantic masks," *IEEE Trans. Circuits Syst. Video Technol.*, 2019.

[77] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2017, pp. 5987–5995.

[78] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2017, pp. 1800–1807.

[79] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017, pp. 764–773.

[80] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017, pp. 2359–2367.

[81] L. Tychsen-Smith and L. Petersson, "Denet: Scalable real-time object detection with directed sparse sampling," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2017, pp. 428–436.

[82] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, *et al.*, "Speed/accuracy trade-offs for modern convolutional object detectors," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2017, pp. 7310–7311.

[83] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2009.

[84] J. Donahue, P. Krähenbühl, and T. Darrell, "Adversarial feature learning," *Proc. Int. Conf. Learn. Representations*, 2016.

[85] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," *J. Mach. Learn. Res.*, 2013.

[86] Y. He, X. Zhang, and J. Sun, "Channel pruning for accelerating very deep neural networks," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2017, pp. 1389–1397.

[87] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. Advances in Neural Inf. Process. Syst.*, 2017, pp. 742–751.

[88] H. Hu, R. Peng, Y.-W. Tai, and C.-K. Tang, "Network trimming: A data-driven neuron pruning approach towards efficient deep architectures," *arXiv preprint arXiv:1607.03250*, 2016.

[89] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[90] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* IEEE, 2012, pp. 3354–3361.

[91] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 743–761, 2012.

[92] X. Wang, T. Xiao, Y. Jiang, S. Shao, J. Sun, and C. Shen, "Repulsion loss: Detecting pedestrians in a crowd," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018, pp. 7774–7783.

[93] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *Proc. Eur. Conf. Comp. Vis.* Springer, 2016, pp. 56–72.

[94] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016.

[95] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, "Optical flow guided feature: A fast and robust motion representation for video action recognition," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2018, pp. 1390–1399.

[96] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, "Flow-guided feature aggregation for video object detection," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2017, pp. 408–417.

[97] P. Viola, M. J. Jones, and D. Snow, "Detecting pedestrians using patterns of motion and appearance," *Int. J. Comput. Vision*, vol. 63, no. 2, pp. 153–161, 2005.

[98] L. Qi, J. Huo, L. Wang, Y. Shi, and Y. Gao, "Maskreid: A mask based deep ranking neural network for person re-identification," *arXiv preprint arXiv:1804.03864*, 2018.

[99] C. Lin, J. Lu, G. Wang, and J. Zhou, "Graininess-aware deep feature learning for pedestrian detection," in *Proc. Eur. Conf. Comp. Vis.*, 2018, pp. 732–747.

[100] Z. Zhang, T. He, H. Zhang, Z. Zhang, J. Xie, and M. Li, "Bag of freebies for training object detection neural networks," *arXiv preprint arXiv:1902.04103*, 2019.

[101] F. Massa and R. Girshick, "maskrcnn-benchmark: Fast, modular reference implementation of Instance Segmentation and Object Detection algorithms in PyTorch," https://github.com/facebookresearch/maskrcnn-benchmark, 2018.

[102] A. Milan, S. H. Rezatofighi, A. Dick, K. Schindler, and I. Reid, "Online multi-target tracking using recurrent neural networks," *arXiv preprint arXiv:1604.03635*, 2016.

[103] G. Ning, Z. Zhang, C. Huang, Z. He, X. Ren, and H. Wang, "Spatially supervised recurrent convolutional neural networks for visual object tracking," *arXiv preprint arXiv:1607.05781*, 2016.

[104] G. Zhu, F. Porikli, and H. Li, "Beyond local search: Tracking objects everywhere with instance-specific proposals," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2016, pp. 943–951.

[105] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung, "Transferring rich feature hierarchies for robust visual tracking," *arXiv preprint arXiv:1501.04587*, 2015.

[106] S. Hong, T. You, S. Kwak, and B. Han, "Online tracking by learning discriminative saliency map with convolutional neural network," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 597–606.

[107] K. Zhang, Q. Liu, Y. Wu, and M. H. Yang, "Robust visual tracking via convolutional networks without training." *IEEE Trans. Image Process.*, vol. 25, no. 4, pp. 1779–1792, 2015.

[108] Y. Sui, Z. Zhang, G. Wang, Y. Tang, and L. Zhang, "Real-time visual tracking: Promoting the robustness of correlation filter learning," in *Proc. Eur. Conf. Comp. Vis.*, 2016, pp. 662–678.

[109] X. Wang, H. Li, Y. Li, F. Porikli, and M. Wang, "Deep tracking with objectness," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 660–664.

[110] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[111] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[112] W. Zhong, H. Lu, and M.-H. Yang, "Robust object tracking via sparsity-based collaborative model," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2012, pp. 1838–1845.

[113] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, 2013, pp. 2411–2418.

[114] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, F. Porikli, L. Cehovin, G. Nebehay, G. Fernandez, T. Vojir, A. Gatt, *et al.*, "The visual object tracking vot2013 challenge results," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2013, pp. 98–111.

[115] M. Kristan, R. Pflugfelder, A. Leonardis, J. Matas, L. Čehovin, G. Nebehay, T. Vojíř, G. Fernández, A. Lukežič, A. Dimitriev, A. Petrosino, A. Saffari, B. Li, B. Han, C. Heng, C. Garcia, D. Pangeršič, G. Häger, F. S. Khan, F. Oven, H. Possegger, H. Bischof, H. Nam, J. Zhu, J. Li, J. Y. Choi, J.-W. Choi, J. F. Henriques, J. van de Weijer, J. Batista, K. Lebeda, K. Öfjäll, K. M. Yi, L. Qin, L. Wen, M. E. Maresca, M. Danelljan, M. Felsberg, M.-M. Cheng, P. Torr, Q. Huang, R. Bowden, S. Hare, S. Y. Lim, S. Hong, S. Liao, S. Hadfield, S. Z. Li, S. Duffner, S. Golodetz, T. Mauthner, V. Vineet, W. Lin, Y. Li, Y. Qi, Z. Lei, and Z. H. Niu, "The visual object tracking vot2014 challenge results," in *Proc. Eur. Conf. Comp. Vis.*, L. Agapito, M. M. Bronstein, and C. Rother, Eds. Springer, 2015, pp. 191–217.

[116] M. Kristan, J. Matas, A. Leonardis, M. Felsberg, L. Cehovin, G. Fernandez, T. Vojir, G. Hager, G. Nebehay, and R. Pflugfelder, "The visual object tracking vot2015 challenge results," in *Proc. IEEE Int. Conf. Comp. Vis.*, 2015, pp. 1–23.

[117] Y. Wu, J. Lim, and M.-H. Yang, "Object tracking benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1834–1848, 2015.