International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 10 | Oct 2020 www.irjet.net p-ISSN: 2395-0072

Object Detection Algorithms in Autonomous Navigation Systems

Neha Chauhan

Abstract: Data is the new oil in current technological society. The impact of efficient data has changed benchmarks of performance in terms of speed and accuracy. The enhancement is visualizable because the processing of data is performed by two buzzwords in industry called Computer Vision (CV) and Artificial Intelligence (AI). Two technologies have empowered major tasks such as object detection and tracking for traffic vigilance systems. As the features in image increases demand for efficient algorithm to excavate hidden features increases. Convolution Neural Network (CNN) model is designed for urban vehicle dataset for single object detection and YOLOv3 for multiple object detection on KITTI and COCO dataset. Model performance is analyzed, evaluated and tabulated using performance metrics such as True Positive (TP), True Negative (TN), False Positive (FP), False Negative (FN), Accuracy, Precision, confusion matrix and mean Average Precession (mAP). Objects are tracked across the frames using YOLOv3 and Simple Online Real Time Tracking (SORT) on traffic surveillance video. This paper upholds the uniqueness of the state of the art networks like DarkNet. The efficient detection and tracking on urban vehicle dataset is witnessed. The algorithms give real-time, accurate, precise identifications suitable for real time traffic applications.

We build a real-time multiple object tracker (MOT) for autonomous navigation using deep convolutional neural networks. To achieve this, we combine state-of-the-art object detection framework, Faster R-CNN architecture. We freeze the pretrained weights for the detection network and train the tracking network on the MOT dataset. We show that such end-to-end modular approach for MOT performance is at par with the available computer vision techniques. We also use our model on real-world scenarios to show the generality of our model.

1. Introduction

Over the past years domains like image analysis and video analysis has gained a wide scope of applications. CV and AI are two main technologies dominating technical society. Technologies try to depict the biology of human. Human vision is the sense through which a perception of outer 3D world is perceived. Human Intelligence is trained over years to distinguish and process scene captured by eyes. These intuitions acts as a crux to budding new technologies. Rich resource is now accelerating researchers to excavate more details form the images. These developments are due to state of the-art methods like CNN. Applications from Google, Facebook, Microsoft, and Snap chat are all results of tremendous improvement in Computer vision and Deep learning. During time, the vision-based technology has transformed from just a sensing modality to intelligent computing systems which can understand the real world. Computer vision applications like vehicle navigation, surveillance and autonomous robot navigation find Object detection and tracking as important challenges. For tracking vehicles and other real word objects, video surveillance is a dynamic environment. In this paper, efficient algorithm is designed for object detection and tracking for video Surveillance in complex environment. Object detection and tracking goes hand in hand for computer vision applications. Object detection is identifying object or locating the instance of interest in-group of suspected frames. Object tracking is identifying trajectory or path; object takes in the concurrent frames. Image obtained from dataset is, collection of frames. Basic block diagram of object detection and tracking. Data set is divided into two parts. 80 % of images in dataset are used for training and 20 % for testing. Image is considered to find objects in it by using algorithms CNN and YOLOv3. A bounding box is formed across object with Intersection over union (IoU) > 0.5. Detected bounding box is sent as references for neural networks aiding them to perform Tracking. Bounded box is tracked in concurrent frames using Multi Object Tracking (MOT). Importance of this research work is used to estimate traffic density in traffic junctions, in autonomous vehicles to detect various kinds of objects with varying illumination, smart city development and intelligent transport systems [18].

2. Review of Literature

2.1 Object Detection

Despite its long history of development since 90's [29, 35], object detection has experienced major breakthrough since 2012 after AlexNet [24] was pop-ularized. Several pivotal works, such as Overfeat [30], SPPNet [18], Fast R-CNN [15], more recently Faster R-CNN [28] and Single Shot Detector (SSD) [26] have advanced the object detection approaches in terms of both speed and accuracy. In the following sections, the latter two works are in-troduced due to the fact that their designs well preserve the advantages but also compensate the shortcomings of the previous object detectors.

2.1.1 Faster R-CNN

Faster R-CNN is an object detector comprising of an object proposal gener-ator and a detection network serving as classifiers classifying the generated object proposals as shown in Figure 2.1. Unlike Fast R-CNN [15] relying on an external object proposal generator, e.g. Selective Search [34], Faster R-CNN introduces Region Proposal Network (RPN) which learns to generate the object proposals during the network training phase. The major contribu-tion of this architecture is that it shares the convolutional features not only among the object proposals (as Fast R-CNN does) but also among the object proposals and detection networks, contributing to less wasted computation and faster inference and, in addition, higher mean Average Precision (mAP) than Fast R-CNN on PASCAL VOC 2007 and 2012 benchmark datasets. In Section 2.1.1.1 and 2.1.1.2 we introduce the RPN architecture and the loss functions deviced to train RPN, respectively.

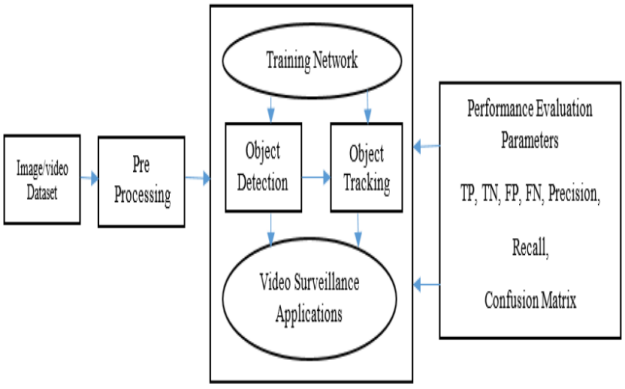


Fig 2.1 Faster R-CNN Network Structure

2.2 Online Single Object Tracking

On-line single object tracking addresses the problem in which given the cur-rent frame and the initial state of an object, the tracker predicts the object's state in the next frame. The object's state in the context of this thesis project is the bounding box that well wraps around the object. Tracking-by-detection is an approach that started gaining its popularity since 2006. The works that applauds tracking-by-detection methodology include on-line boosting trackers [5, 16, 17], tracking-learning-detection (TLD) tracker [22], and later on the correlation filtering tracker, such as MOSSE tracker [9], DSST [12], kernelized correlation filtering tracker (KCF) [19], etc.

Due to their computational efficiency and effectiveness on modeling an object's appearance, correlation filtering based trackers have become the state-of-the-art where their results are always ranked top on different benchmark datasets. Hence, in the later sections, we introduce Minimum Output Sum of Squared Error (MOSSE) tracker, which is one of the very first approaches that apply correlation filters for the tracking problem, and Discriminative Scale Space Tracker (DSST), which had won the Multiple Object Tracking'14 Challenge and still is able to operate in real-time on CPU.

2.3 Online Multiple Object Tracker

A multiple object tracker (MOT) typically has to handle a number of difficulties. Firstly, the identity switches, i.e. the situation in which the tracker mistakes another target for the one it is supposed to track, may happen due to the presence of other objects similar in their appearances. This situation happens frequently when the test videos are taken from the street view and other public spaces where the pedestrians wearing clothes with similar colors and styles and/or are highly occluded by each other. Secondly, because of the constraint of online methodology which cannot peek into future frames, the *tracklet*, i.e. the tracking trajectory on the same target, may be fragmented due to some tracking errors such as identity switches. Thirdly, if an object has been occluded for quite some time and re-appears, the tracker should be able to recognize and start tracking it again. To address these difficulties, Nicolai Wojke et al. [36] proposed an online multiple object tracking framework which incorporates an object detector, Kalman filter as the base tracker, and a data association method which is based on the features learned from a deep neural net to associate the results from detectors and trackers.

Many approaches like feature extraction based on color and gradients fail to give spatial positioning in the image. The challenges are overcome by employing Analysis of principal components by PCANet [4] pipeline of image undistortion, image registration, classification and detections based on coordinates and velocities. Approach uses detectors like FAST, FREAK descriptors and followed by classification of Squeeze Net [5]. The workflow of candidate target generation,

extracting features from candidate targets, the ground truth boxes around objects assist in tracking. The objects are classified using VGGNet [6]. CNN was designed to classify images, was repurposed to perform the object detection. The approach treats object detection as a relapse for object class to bounding objects detected. Series of gradual improvements has been witnessed from RCNN, Fast RCNN and faster RCNN then finally to YOLO. Instead of assessing image repetitively as in CNN, image is scanned once for all, thereby increasing the processing of frames per second (fps). YOLO is trained based on loss occurred unlike the traditional Classification approach [7]. Paper describes about video analytics part for road traffic. One of main application area apart from vehicle detection and tracking is vehicle counting. One of the novel algorithm called Single Shot Detector (SSD) is employed. Algorithm handles features like Binary large objects. It gives better results in applications like classification of objects. Object tracking employs concepts like background subtraction and virtual coil method. In terms of precision SSD outperforms YOLO versions. Swiftness and precision are always tradeoffs while selecting the right algorithm for object detection with the speed of 58fps performance metric for accuracy exceeds 85% [8], paper explains about upgradation to YOLO was made in the paper. Gradual updating has been witnessed throughout series of YOLO versions namely YOLOv1, YOLOv2, YOLOv3. YOLOv3 is state of the art technology. Upgradation such as thinner bounding boxes without affecting adjacent pixels. YOLOv3's implementation on COCO dataset shows mAP as good as SSD. YOLOv3 gives three times faster results. YOLOv3 promises in detecting smaller objects [9]. With increase in vehicle density in urban region, Single object tracking will no longer cater for the need. Multi object tracking is achieved by employing kernelized correlation filter (KCF). Many KCF are run in parallel. KCF is best suited when images have occlusions. KCF when combined with background subtraction yield reliable results on the urban traffic [10] [12] [14].

3. Methodology

In this chapter, we describe the proposed framework for multiple object track-ing in detail. Our primary object of interest in this work is *pedestrian*, and we do not impose any assumptions on the target object's size, aspect ratio, or appearance. Thereby, it is possible that the proposed framework can be extended to different object classes. In general, we follow the framework pro-posed in [36] but with few major adaptions. First, we replace the Kalman tracker in [36] with DSST tracker [12]. The rationale behind this is that when updating the Kalman state, one has to provide the measurement made in the current frame, which is, in their case, the measurement from object detector. Without the measurement from the object detector, the Kalman tracker would update the state merely with the pre-modeled linear motion [8]. As in our case, we do not assume the availability of the object detector in every frame, hence updating the Kalman tracker with only motion pre-diction may result in unsatisfatory result. Second, we do not train an object detector specifically as in [36], but we employ the object detector from [3] trained on MS COCO dataset [25] where it provides multiple detectors of different base networks (i.e. MobileNet V1 [20], InceptionV2 [33], RFCN [11], Faster RCNN [28]) that tradeoff the speed and accuracy [21]. Third, in order to monitor if a tracker starts to drift or has drifted, we measure the similarity between the patches of the tracked target in any two consecutive frames. Fourth, to enable the tracker recover from tracking failure, we employ a simple person re-identification method that is as well based on the same deep features. In the pursuit of a more efficient implementation, the similarity is measured based on the deep features extracted from the network that has been served as the base network in the object detector in use. These modifications enable the proposed framework to detect and track the object.

Table 3.1: Base network structure of InceptionV2 [2]. The classification layers have been removed as we adopt the network as a generic feature ex-traction, hence the classification layers are not needed. Please note that the input size and the structure are different from what is described in [33]. Our implementation follows the one provided in [2]. Figure 3.3a to 3.3j can be seen on page 38 to 39.

FasterRCNN(
(Transform): Generalized RCNN Transform(
Normalize (mean= [0.485, 0.456, 0.406], std= [0.229, 0.224, 0.225])
Resize (min_size= (800,), max_size=1333, mode='bilinear')
)
(backbone): Backbone With FPN(
(body): Intermediate Layer Getter(

(conv1): Conv2d(3, 64, kernel_size=(7, 7), stride=(2, 2), padding=(3, 3), bias=False)
(bn1): FrozenBatchNorm2d(64)
(relu): ReLU(inplace=True)
(maxpool): MaxPool2d(kernel_size=3, stride=2, padding=1, dilation=1, ceil_mode=False)
(layer1): Sequential(
(0): Bottleneck(
(conv1): Conv2d(64, 64, kernel_size=(1, 1), stride=(1, 1), bias=False)
(bn1): Frozen Batch Norm2d(64)
(conv2): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1), bias=False)
(bn2): Frozen Batch Norm2d(64)
(conv3): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
(bn3): Frozen Batch Norm2d(256)
(relu): ReLU(inplace=True)
(downsample): Sequential(
(0): Conv2d(64, 256, kernel_size=(1, 1), stride=(1, 1), bias=False)
(1): Frozen Batch Norm2d(256)
)
)

3.1 Update of Trackers

Three major steps are involved in updating the trackers, (1) update the DSST tracker, (2) update the auxiliary templates of the targets to track, and (3) update the auxiliary templates with adaptive learning rate. As described in 2.2.2.1, DSST seeks the location with maximal response as the final prediction of the target location. The maximal response (or regression score) can, in the one hand, be interpreted as how confident the tracker is, but in the other hand, the scores are positive numbers which are not strictly bounded within a range, e.g. [0, 100] or [0, 1]. This makes the response difficult to be interpreted and served as a reliable measurement.

Figure 3.4 The target's auxiliary template may be updated with wrong image content when a tracker starts to drift since much background information is included within the tracker's bounding box.

Figure 3.5 When a tracker starts drifting away from the target, it is likely to be stuck at a background object but still recognizes it as the target to track so the auxiliary template is kept updating with some learning rate. Under that case, $sim(F^t_i, F^t_i)$ will retain high value and keep increasing until it saturates to a rather high value.

4. Experiments

4.1 Evaluation Dataset and Protocols

MOT Challenge 2017 (MOT'17) offers 14 video sequences evenly divided into seven training and seven testing sequences summarized in Table 4.1. The target class of the evaluation focus is *pedestrian*. Particularly, in MOT'17 challenge the pedestrians who are static (e.g. sitting or standing without moving), behind the glasses, in the reflection, or in the vehicles are omitted and not considered in the evaluation. Thus, constantly moving pedestrians are the only left. The videos are taken in unconstrained public spaces (e.g. open streets, shopping malls, squares, etc.) that are usually crowded. Some cameras are installed in driving vehicles, some are carried by a walking per-son, and some are stationary. Challenges including wide variety of sizes, orientations, walking speeds, and heavy occlusions make the dataset realistic and to highly correspond to the real-world applications.

Table 4.1: Summary of MOT'17 training and test sets.

training set	*test set*	*frame rate (fps)*	*camera*
MOT17-02	*MOT17-01*	*30*	*static*
MOT17-04 7	*MOT17-03*	*30*	*static*
MOT17-05	*MOT17-06*	*14*	*dynamic*
MOT17-09	*MOT17-07*	*30*	*dynamic*
MOT17-10	*MOT17-08*	*30*	*static*
MOT17-11	*MOT17-12*	*30*	*dynamic*
MOT17-13	*MOT17-14*	*25*	*dynamic*

Noted in [27], it is difficult to quantify a MOT tracker's performance or capture the charateristics of the tracker with a single metric. Among all the existing metrics that are designed for assessing MOT systems, CLEAR metrics [32] and the metrics introduced in [37] have been the most widely used. Please note that even these metrics are the most trendy in the recent MOT works and treated as the standard measures, but the research of standarizing the metrics for MOT problem is still ongoing [27]. In MOT'17, those metrics are used altogether to assess the overall performance while the trackers can be ranked by their average ranking calculated from the ranks with respect to each individual metric. In the following, we walk through the formal definition of every metric included in MOT'17.

True Positive (TP), False Positive (FP), False Negative (FN): These are the most common metrics quantifying the hypotheses made by the tracker. TP measures whether the hypotheses are matched to the annotations while FP measures if they are false alarms. FN measures the misses of the hypotheses with respect to the annotations. Either metric is counted when the IoU is less than 0.5 as suggested in [27].

Precision (Precision), Recall (Recall): Precision is defined in (4.1), reflecting how relevant the predicted bounding boxes are to the ground-truth bounding boxes. Recall is defined (4.2):

$$\text{Precision} = \frac{TP}{TP+FP} \quad (4.1)$$

$$\text{Recall} = \frac{TP}{\text{number of ground-truth bounding boxes}} \quad (4.2)$$

Identity Switch (IDs): IDs counts the mismatching error which happens when an annotated target x is matched to a track y in frame t−1 but matched to another track z, z = y in frame t. Note that IDs alone may not inform the tracker's overall performance as it usually correlates with the number of annotated tracks. Hence, one can instead look at the ratio of IDs to the recall when needed. Note that throughout the experiments, we still report the raw IDs as Recall is also reported.

Fragmentation (FM): A fragmentation is counted when a track is inter-rupted for some frames and recovered either with or without ID switches.

Multiple Object Tracking Accuracy (MOTA): MOTA considers three sources of metrics to assess the overall accuracy of a MOT tracker across the frames. More formally, it is defined as

$$\text{MOTA} = 1 - \frac{\sum_t (FN_t + FP_t + IDs_t)}{\sum_t GT_t}, \quad (4.3)$$

where the subscript t denotes the frame index and GT_t denotes the number of objects in frame t. Note that it is possible that MOTA value is below zero while its maximum is 1.

Multiple Object Tracking Precision (MOTP): MOTP measures in aver-age across all frames how well do the tracker's outputs overlap with the annotations. More formally, it is defined as

$$\text{MOTP} = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}, \quad (4.4)$$

where c_t is the number of annotations in frame t and $d_{t,i}$ is the IoU value of the target i and its assigned annotation. In short, MOTP is used to measure the localization accuracy of a system where the detecter and the tracker work collaboratively with each other.

Number of Ground-Truth Tracks (GT), Mostly Tracked (MT), Partly Tracked (PT), Mostly Lost (ML): A track is said to be *mostly tracked* if over 80% of its annotations along the track are matched correctly to the tracker's outputs, while it is said to be *mostly lost* if under 20% of its annotations along the track is matched correctly, otherwise it is classified as *partly lost*. We define MT, PT, and ML as the percentage of each quantity (i.e. the numbers of mostly tracked, partly tracked, and mostly lost) to the number of grount-truth tracks, GT.

Tracker Ranking (TR): TR does not reflect the overall performance of a MOT tracker, but provides a relative figure that ranks the trackers by com-paring the average ranking. The average ranking is calculated according to the rank made by each individual metric (IDSW, MOTA, MOTP, etc).

In the following section, we provide ablation studies on how do the different parameters in the proposed framework affect each metric.

4.1.1 Results on MOT'17 Training Set

Table 4.6 shows the average results on all training sequences. While the n = 1 case slightly outperforms the n = 3 case in FP, MT, PT, and ML, n = 3 shows stronger in MOTA, MOTP, FN, IDs, and FM. However, the performance discrepancy between the two cases is not that significant. Next, we break it down to show the performance on each sequence in Tables 4.7 and 4.8. Note that here we present the results in two categories of video sequences: (1) the sequences captured by static cameras, and (2) the sequences captured by moving cameras. The former category is shown in Table 4.7 and the latter is shown in Table 4.8. The reason for the arrangement is to study if the frameworks based on different parameter settings would favor the different dynamics in the videos. Table 4.7 shows that the n = 3 case outperforms the other one consistently in MOTA, MOTP, FN, IDs, FM, and Recall. On two out of three sequences, the n = 3 case as well out performs the in PT and ML. As for the sequences presented in Table 4.8, one can see that on *MOT17-05*, the n = 1 case performs better in most of the metrics, i.e. MOTA, FP, FN, IDs, FM, Recall, Precision, MT, PT, and ML. On other dynamic sequences, the two cases perform rather similarly to each other. Later on, we study the results on the MOT'17 test set to examine if the proposed framework behaves similarly according to the dynamics in the videos.

Table 4.6: Comparison of average performance of the n = 1 and n = 3 cases on seven MOT'17 training sequences. Parameter selection is done by the strategy introduced in Section 4.2. Bold figures indicate the winner cases.

	MOTA	MOTP	FP	FN	IDs	FM
n = 1	8.1	70.7	4809	98067	340	752
n = 3	8.3	70.9	5344	97403	226	619
	Recall	Precision	MT (%)	PT (%)	ML (%)	
n = 1	12.0	74.0	1.8	27.8	70.3	
n = 3	13.0	73.0	1.1	25.1	73.8	

Table 4.7: The proposed framework on MOT'17 (training) static sequences (i.e. the camera is not moving). The bold figures indicate better performance.

MOT17-02

	MOTA	MOTP	FP	FN	IDs	FM
n = 1	6.6	69.1	571	16734	46	92
n = 3	7.3	69.6	602	16591	26	62
	Recall	Precision	MT (%)	PT (%)	ML (%)	
n = 1	9.9	76.4	1.6	17.7	80.1	
n = 3	10.7	76.8	1.6	21.0	77.4	
MOT17-04						

	MOTA	MOTP	FP	FN	IDs	FM
n = 1	4.0	69.5	539	45087	39	116
n = 3	5.0	70.3	861	44288	18	78
	Recall	Precision	MT (%)	PT (%)	ML (%)	
n = 1	5.2	82.1	0.0	10.8	89.1	
n = 3	6.9	79.2	0.0	13.3	86.7	
MOT17-09						
	MOTA	MOTP	FP	FN	IDs	FM
n = 1	25.4	72.4	303	3622	49	87
n = 3	26.3	73.5	353	3537	33	64
	Recall	Precision	MT (%)	PT (%)	ML (%)	
n = 1	32.0	84.9	7.7	50.0	42.3	
n = 3	33.6	83.5	7.7	50.0	42.3	

4.1.2 Results on MOT'17 Test Set

The results on the MOT'17 test set are summarized in Tables 4.9 to 4.11. All in all, the n = 1 case slightly outperforms the n = 3 case in MOTA, MOTP, FP, Precision, MT, PT, and ML as shown in Table 4.9. Next, Table 4.10 shows the performance of the proposed framework on the static sequences in the MOT'17 test set. The n = 1 case shows stronger in MOTA in two out of three sequences, and consistently outperforms the n = 3 case in MOTP. The n = 3 case shows slightly better consistency in keeping the tracks not being fragmented, which reflects in higher MT on *MOT17-01*, higher PT on *MOT17-03*, and higher (MT + PT) combined on *MOT17-08*. As the performance on the dynamic sequences, the n = 1 case out performs another on every sequence in MOTA. This indicates the n = 1 setting is still favored if the dynamics in the videos is highly fluid.

Table 4.8: The proposed framework on MOT'17 (training) dynamic sequences (i.e. camera is moving). The bold figures indicate better performance.

MOT17-05

	MOTA	MOTP	FP	FN	IDs	FM
n = 1	30.9	71.6	743	3949	90	119
n = 3	21.4	72.3	751	4619	67	173
	Recall	Precision	MT (%)	PT (%)	ML (%)	
n = 1	42.9	80.0	5.3	51.1	43.6	
n = 3	33.2	75.4	0.6	40.6	58.7	
MOT17-10						
	MOTA	MOTP	FP	FN	IDs	FM
n = 1	0.7	66.2	1568	11123	61	159
n = 3	2.3	66.4	1609	10884	48	133

	Recall	Precision	MT (%)	PT (%)	ML (%)	
n = 1	13.4	52.3	0.0	26.3	73.7	
n = 3	15.2	54.9	0.0	24.6	75.4	
MOT17-11						
	MOTA	MOTP	FP	FN	IDs	FM
n = 1	24.6	75.1	371	6710	36	111
n = 3	25.3	74.2	406	6617	29	78
	Recall	Precision	MT (%)	PT (%)	ML (%)	
n = 1	28.9	88	0.0	30.7	69.3	
n = 3	29.9	87.4	1.3	29.3	69.3	
MOT17-13						
	MOTA	MOTP	FP	FN	IDs	FM
n = 1	0.6	65.6	714	10842	19	68
n = 3	0.1	65.8	762	10867	5	31
	Recall	Precision	MT (%)	PT (%)	ML (%)	
n = 1	6.9	52.8	0.0	11.8	88.2	
n = 3	6.7	50.4	0.9	9.1	90.0	

4.2 Discussion

In this section, we first discuss the experiments conducted in the ablation studies. The discussion continues with the comparison made between the proposed framework and other on-line trackers. Finally, we discuss the possible improvements based on the observations made in the experiments.

4.4.1 On Ablation Studies

In the studies, the detector threshold τ^{det} has shown to be most impactful to all of the metrics. Note that in these experiments we attempt to optimize MOTA score, and thus it is shown clearly (in Figures 4.1b and 4.1e) that their FP, FN, MT, and PT scores are not optimized along with MOTA. Hence, it is worth mentioning that if one would like to pursue tracking results which can cover as many as ground-truth tracks as possible, a lower detector threshold τ^{det} (than 0.6) should be applied for higher recall but in return of lower precision rate.

Secondly, we analyze the results obtained from the static video sequences in both training and test sets shown in Tables 4.7 and 4.10. It is observed that while only in few static sequences does the n = 1 case outscore the another case in MOTA, the n = 3 case achieves higher or equal MT and PT scores combined. On the one hand, the results indicate that the n = 3 case, which relies more on correlation filters that localize the targets, delivers more consistent and uninterrupted tracks. On the other hand, the n = 1 case, which fuses the information from trackers and detectors, does not offer significant improvements over these metrics. While it was our belief that the detections could calibrate the predicted target locations and prevent the tracker from drifting away, the association process could be itself noisy and instead provide interference to the trackers. This can happen if the cost matrix is not well-devised in the association process (refer to Section 3.2.3), thus providing noisy estimation of the association cost.

Thirdly, to analyze the results from dynamic video sequences, two observations are worth mentioning: (1) the n = 1 case draws significant improvement over the n = 3 case in almost all the metrics on the video *MOT17-05* in the training set and its counterpart test video *MOT17-06*, e.g. 31.5% and 30.74% boosts in MOTA, respectively. By watching the contents in *MOT17-05* and *MOT17-06*, one can observe that these are taken at 14 fps and possibly with a hand-held camera carried by a walking person. Large movement of the cameras is constantly presented and could create some difficulties for the trackers to track the target. (2) The n = 1 case draws significant improvement over the n = 3 case in MOTA, MT, PT, and ML on the test sequences. These oberservations develop the thought that if the contents in the sequences are changing rapidly due to high dynamics of the scene or low frame-rate production, triggering detectors more frequently is much demanding for consistent tracking results.

Fourthly, the n = 1 case performs worse in IDs and FM in almost every sequence than the other case. The cause of the unsatisfyingly many ID switches and fragmented tracks could be correlated with the point already discussed that the association may not be sufficiently reliable.

4.4.2 On Comparison with Other Online Trackers

Previously we have shown that the proposed framework is out-performed by SORT and Deep SORT by a large margin. For instance, the proposed framework incurs nearly three times FN and similar FP compared with that in Deep SORT. This indicates that the major difficulty is the pro-posed framework missing a huge number of detections, especially the misses on the pedestrians beyond some certain distances.

Figure 4.6 shows two cases where larger objects are always easier to detect than small and clutter ones. Hence, it could be the accuracy gap between the different detectors that leads to the performance gap between the proposed framework and Deep SORT (or SORT). As mentioned in [7, 36], a sophisticated object detector is much demanded in the tracking framework for high accuracy, e.g. simply replacing the base network used in Faster R-CNN from ZFNet [38] to VGG16 [31] can improve MOTA from 24 to 34.

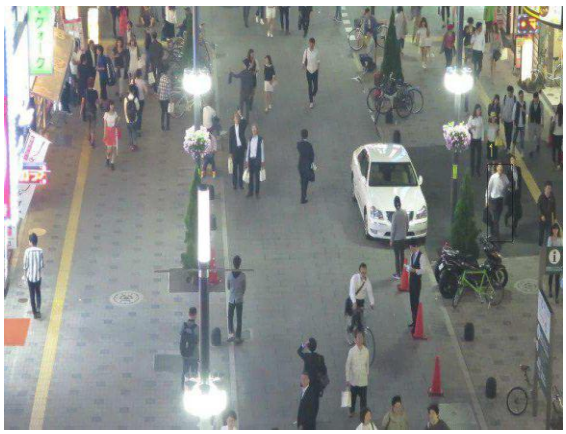
Despite the importance of selecting a highly accurate object detector, we emphasize that the aim of this work is to design a tracking framework being able to run at a reasonable speed without or with little modern GPU support. We cannot afford relying on the state-of-the-art object detector to boost the performance, instead we choose an object detector (introduced in Section 3.2.1) which runs at around 2.5fps on quad-core 2.6GHz CPU with no GPU support. In addition, we employ a more sophisticated single-target base tracker than the Kalman tracker used in SORT and Deep SORT, i.e. the correlation tracker that still ensures the real-time performance. By comparing n = 1 and n = 3 cases in the proposed framework shown in Table 4.12, the n = 3 case achieves comparable MOTP under similar FN with n = 1 case. This shows that the correlation tracker is able to localize the target accurately to some extent. However, the n = 3 case incurs higher FP which can be generated from the detector but can be also from the tracks that have been drifted to the wrong targets.

4.4.3 On Possible Improvements

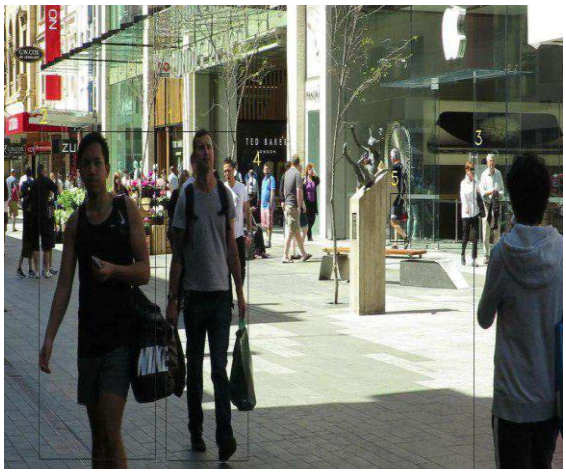
We provide a generic tracking framework in this work that does not require the availability of object detector in every frame. However, there is much room for improvement in the major constituents. We describe two possible improvements as follows, ordered by the suggested importance.

The pedestrian detector: It is of our knowledge that the detector's performance has the most impact on tracking performance. While we do not want to sacrifice the inference speed too much for the detection performance, alternatively one can train the same SSD network specifically on the large-scale pedestrian dataset (e.g. MARS [39] and KITTI [14]) initialized by the parameters in the SSD network used by this work. For improving the inference speed, one can remove all the neurons of classes in the classification layer except the *person* class. Thus, the computation will not be wasted on inferring other classes always ignored.

The deep features: Deep features incorporated in the framework plays an important role whenever data association takes place, e.g. data association for associating results from detector and trackers or for person re-identification. As shown in the experiment results triggering the detector more frequently does not show significant improvement over the metrics in general, despite that the detector we employ is of low recall rate, the association could possibly be improved by further reducing the false positive and false negative rates. For that, the discriminative power and descriptiveness



(a) A frame in MOT17-03. Only one detection is shown close to the right border.



(b) A frame in MOT17-08. Five detections are shown.

Figure 4.6: The top image shows when captured at far-range distance, the detector employed in the proposed framework often misses many small detections. The bottom image shows a relatively easier case for the detector as the pedestrians are close enough to the camera of the deep features should be improved. In addition, recall that we calculate the similarity between two auxiliary templates by averaging their channel-wise similarities where each template is of 24 × 24 × 64 dimensions. However, empirically we observe that the features within some of the channels are sparse, i.e. many features are of zero values. On the one hand, this could lead to the curse of dimensionality where the distance measure becomes less meaningful. On the other hand, it may also indicate that the features we ex-tract may not be sufficiently condensed and informative. Hence, to improve the deep features, it is suggested that one can train a network specifically on pedestrian data and use it for feature extraction [36]. What is more, to mitigate the curse of dimensionality, the features can be extracted from a fully-connected layer near the end of the network, thus the features would be of single channel and lower dimensionality (e.g. 128-D) [36].

5. Conclusions and Future Work

5.1 Conclusions

We presented in this thesis an on-line detect-and-track framework which aims to be operated in real-time without or with minimal GPU support. Unlike most of the multiple object tracking systems, the proposed framework does not assume the detector's availability in every frame as object detection is usually the largest computational burden in such systems. Overall, the pro-posed system works as follows. The tracker localizes the targets itself or with the information (e.g. estimated location of a target) periodically provided by the detector. During the course of tracking, a track of a target is constructed based on continually receiving above-the-threshold similarity measure between the target's auxiliary

templates in the two consecutive frames. In other words, a track is interrupted if the auxiliary templates are dissimilar to some extent, and a track is removed from the active tracks if there are too many interrupts. However, the removed tracks are moved to history tracks in which the tracks still have chances to be recovered in the future frames. The proposed system devises Single Shot Detector and correlation filter as the object detector and tracker, respectively. All relevant similarity measurements are based on the distance between the features in the Euclidean space extracted from the deep neural net.

We conducted experiments on the MOT'17 challenge dataset to demonstrate how the framework performs under full and partial availability of the detector, i.e. the detector is triggered in every frame versus in every three frames. The main findings in the experiments are: (1) in static sequences (where the camera is not moving) the case with partial availability of the detector achieves comparable or slightly better performance than the other case, however, (2) in dynamic sequences with a moving camera or when the dynamics in the video are high (i.e. when people are moving faster), the case with full availability of the detector tends to outperform that with partial availability of the detector, and (3) comparing the proposed framework with two other recently published on-line trackers, SORT and Deep SORT track-ers, the proposed framework is underperformed in the MOTA scores. These trackers, however, leverage a more sophisticated object detector which we cannot afford due to the excessive computational burden.

5.2 Future Work

Besides the improvements already suggested, it is also important to investigate how to share the features among the detector, tracker, and data association stages. Currently, in the proposed framework, the detector uses its own network model to do the inference while the tracker utilizes pixel values and a histogram of oriented gradients as the features. Sharing the features in similar tasks may bring several benefits, such as a higher level of generalization and less wasted computations, as suggested in [26, 28].

A recent publication proposed a two-way Siamese-like networks (i.e. two network streams fed with the frames at time t and (t + 1) as inputs respectively) to allow the system learn object representation and localization end-to-end [6]. Hence, it would be interesting to extend their work to one that learns object representation, detection, and localization given two consecutive frames. In addition, while the state-of-the-art object detectors pre-dominantly consider only spatial information from a single frame, in the context of object tracking, temporal information can be considered and possibly used to enhance the detection accuracy and consistency over frames if the detection and tracking are performed within a unified network. We leave the aforementioned as some thoughts for the future development of the project.

References

1. ABADI, M., AGARWAL, A., BARHAM, P., BREVDO, E., CHEN, Z., CITRO, C., CORRADO, G. S., DAVIS, A., DEAN, J., DEVIN, M., GHEMAWAT, S., GOODFELLOW, I., HARP, A., IRVING, G., ISARD, M., JIA, Y., JOZEFOWICZ, R., KAISER, L., KUDLUR, M., LEVEN- BERG, J., MANE´, D., MONGA, R., MOORE, S., MURRAY, D., OLAH, C., SCHUSTER, M., SHLENS, J., STEINER, B., SUTSKEVER, I., TAL-WAR, K., TUCKER, P., VANHOUCKE, V., VASUDEVAN, V., VIEGAS´, F., VINYALS, O., WARDEN, P., WATTENBERG, M., WICKE, M., YU, Y., AND ZHENG, X. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

2. BABENKO, B., YANG, M.-H., AND BELONGIE, S. Visual tracking with online multiple instance learning. In Computer Vision and Pat-tern Recognition, 2009. CVPR 2009. IEEE Conference on (2009), IEEE, pp. 983–990.

3. BERTINETTO, L., VALMADRE, J., HENRIQUES, J. F., VEDALDI, A., AND TORR, P. H. Fully-convolutional siamese networks for object tracking. In European Conference on Computer Vision (2016), Springer, pp. 850–865.

4. BEWLEY, A., GE, Z., OTT, L., RAMOS, F., AND UPCROFT, B. Simple online and real time tracking. In Image Processing (ICIP), 2016 IEEE International Conference on (2016), IEEE, pp. 3464–3468.

5. BISHOP, G., AND WELCH, G. An introduction to the kalman filter. Proc of SIGGRAPH, Course 8, 27599-23175 (2001), 41.

6. BOLME, D. S., BEVERIDGE, J. R., DRAPER, B. A., AND LUI, Y. M. Visual object tracking using adaptive correlation filters. In Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on (2010), IEEE, pp. 2544–2550.

7. BRUFF, D. The assignment problem and the hungarian method. Notes for Math 20 (2005), 29–47.

8. DAI, J., LI, Y., HE, K., AND SUN, J. R-fcn: Object detection via region-based fully convolutional networks. In Advances in neural infor-mation processing systems (2016), pp. 379–387.

9. DANELLJAN, M., HAGER¨, G., KHAN, F., AND FELSBERG, M. Accurate scale estimation for robust visual tracking. In British Machine Vision Conference, Nottingham, September 1-5, 2014 (2014), BMVA Press.

10. DANELLJAN, M., HAGER, G., SHAHBAZ KHAN, F., AND FELSBERG, M. Convolutional features for correlation filter based visual tracking. In Proceedings of the IEEE International Conference on Computer Vision Workshops (2015), pp. 58–66.

11. GEIGER, A., LENZ, P., STILLER, C., AND URTASUN, R. Vision meets robotics: The kitti dataset. International Journal of Robotics Research (IJRR) (2013).

12. GIRSHICK, R. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision (2015), pp. 1440–1448.

13. GRABNER, H., AND BISCHOF, H. On-line boosting and vision. In Computer Vision and Pattern Recognition, 2006 IEEE Computer Soci-ety Conference on (2006), vol. 1, IEEE, pp. 260–267.

14. GRABNER, H., LEISTNER, C., AND BISCHOF, H. Semi-supervised on-line boosting for robust tracking. Computer Vision–ECCV 2008 (2008), 234–247.

15. HE, K., ZHANG, X., REN, S., AND SUN, J. Spatial pyramid pool-ing in deep convolutional networks for visual recognition. In European Conference on Computer Vision (2014), Springer, pp. 346–361

16. HENRIQUES, J. F., CASEIRO, R., MARTINS, P., AND BATISTA, J. High-speed tracking with kernelized correlation filters. IEEE Transac-tions on Pattern Analysis and Machine Intelligence 37, 3 (2015), 583– 596.

17. HOWARD, A. G., ZHU, M., CHEN, B., KALENICHENKO, D., WANG, W., WEYAND, T., ANDREETTO, M., AND ADAM, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861 (2017).

18. HUANG, J., RATHOD, V., SUN, C., ZHU, M., KORATTIKARA, A., FATHI, A., FISCHER, I., WOJNA, Z., SONG, Y., GUADARRAMA, S., AND MURPHY, K. Speed/accuracy trade-offs for modern convolutional object detectors. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017).

19. KALAL, Z., MIKOLAJCZYK, K., AND MATAS, J. Tracking-learning-detection. IEEE transactions on pattern analysis and machine intelli-gence 34, 7 (2012), 1409–1422.

20. KING, D. E. Dlib-ml: A machine learning toolkit. Journal of Machine Learning Research 10 (2009), 1755–1758.

21. KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (2012), pp. 1097–1105.

22. LIN, T.-Y., MAIRE, M., BELONGIE, S., HAYS, J., PERONA, P., RAMANAN, D., DOLLAR´, P., AND ZITNICK, C. L. Microsoft coco Common objects in context. In European conference on computer vision (2014), Springer, pp. 740–755.

23. LIU, W., ANGUELOV, D., ERHAN, D., SZEGEDY, C., REED, S., FU, C.-Y., AND BERG, A. C. Ssd: Single shot multibox detector. In European conference on computer vision (2016), Springer, pp. 21–37.

24. MILAN, A., LEAL-TAIXE´, L., REID, I., ROTH, S., AND SCHINDLER, K. Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831 (2016).

25. REN, S., HE, K., GIRSHICK, R., AND SUN, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (2015), pp. 91–99.

26. ROWLEY, H. A., BALUJA, S., AND KANADE, T. Neural network-based face detection. IEEE Transactions on pattern analysis and machine intelligence 20, 1 (1998), 23–38.

27. SERMANET, P., EIGEN, D., ZHANG, X., MATHIEU, M., FERGUS, R., AND LECUN, Y. Overfeat: Integrated recognition, localization and detection using convolutional networks. arXiv preprint arXiv:1312.6229 (2013).

28. SIMONYAN, K., AND ZISSERMAN, A. Very deep convolutional net-works for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014).

29. STIEFELHAGEN, R., BERNARDIN, K., BOWERS, R., GAROFOLO, J., MOSTEFA, D., AND SOUNDARARAJAN, P. The clear 2006 evaluation. In International Evaluation Workshop on Classification of Events, Activities and Relationships (2006), Springer, pp. 1–44.

30. SZEGEDY, C., VANHOUCKE, V., IOFFE, S., SHLENS, J., AND WOJNA, Z. Rethinking the inception architecture for computer vision. CoRR abs/1512.00567 (2015).

31. UIJLINGS, J. R., VAN DE SANDE, K. E., GEVERS, T., AND SMEULDERS, A. W. Selective search for object recognition. International journal of computer vision 104, 2 (2013), 154–171.

32. VIOLA, P., AND JONES, M. J. Robust real-time face detection. Inter-national journal of computer vision 57, 2 (2004), 137–154.

33. WOJKE, N., BEWLEY, A., AND PAULUS, D. Simple online and real time tracking with a deep association metric. CoRR abs/1703.07402 (2017).

34. WU, B., AND NEVATIA, R. Tracking of multiple, partially occluded hu-mans based on static body part detection. In Computer Vision and Pat-tern Recognition, 2006 IEEE Computer Society Conference on (2006), vol. 1, IEEE, pp. 951–958.

35. ZEILER, M. D., AND FERGUS, R. Visualizing and understanding con-volutional networks. In European conference on computer vision (2014), Springer, pp. 818–833.

36. ZHENG, L., BIE, Z., SUN, Y., WANG, J., SU, C., WANG, S., AND TIAN, Q. Mars: A video benchmark for large-scale person re-identification. In European Conference on Computer Vision (2016), Springer, pp. 868–884.