CenterTrack with Crop-based Detection

Shepard Xia
CS 3860 Final Report
Institute for Software Integrated Systems
Vanderbilt University
May 07, 2022

I. INTRODUCTION

Multiple Object Tracking (MOT) refers to the problem of identifying and localizing multiple objects in a video and representing them as a set of trajectories. MOT has always been a topic of interest in computer vision, as it has a wide range of applications, ranging from self-driving cars to monitoring systems. MOT presents us with a set of non-trivial problems like dealing with occlusion and matching objects across frames, all the while the performance also depends on the detector. As good as a one-size-fits-all tracker sounds, it would inevitably come with shortcomings, either sacrifice of speed, or compromise in accuracy. On the other hand, by placing certain constraints on the context, we could develop trackers specific to scenarios that perform especially well, as the constraints could be leveraged to eliminate certain concerns that, if dealt with, would otherwise drag down the performance.

During my research this semester in the I-24 Mobility Technology Interstate Observation Network (MOTION) lab at Institute for Software Integrated Systems, I worked on developing such a tracker that leverages the tracking scenario to boost performance of the tracker. I-24 MOTION lab is dedicated to a project called Smart Corridors that in the near future will use over 300 ultra-HD cameras to capture the behavior of vehicles on a 6 miles segment of Interstate 24 that cuts right through Nashville. The data collected will be used to provide a testbed for understanding how all kinds of vehicles on highways interact with each other and the infrastructure, thus allowing better congestion management.

A state of the art tracker that would work well in this particular context is crucial to the success of the project. My work this semester was to borrow Crop-KIOU's idea[1] of crop-based detection and apply it on baseline trackers to see if the method would further improve their performance.

II. EXISTING MODELS

There is not a definite way to determine the performance of MOT models quantitatively, while the comparisons are dependent on two main choices of testing: dataset that the model is benchmarked on and the metrics looked at. The dataset used refers to the video feeds that the tracker ran on, and the choice of which is important because different datasets take interest in different contexts and targets of tracking, with MOTChallenge being the most popular for pedestrians in indoor and outdoor contexts, while there are also novel datasets like SeaDronesSee that helps develop systems for Search and Rescue missions at sea[2]. For tracking vehicles on roads, UA-DETRAC dataset is among the most commonly used as it features videos captured in different weather conditions, camera angles and vehicle density. UA-DETRAC encourages detection based tracking, which refers to separating the task of tracking objects into two parts, as first detection is performed and then a tracker is used to associate objects across the frames. Because the performance of the MOT models is mainly based on the detector used, next I will review some top-performing MOT models[3] on UA-DETRAC in terms of Multiple Objects Tracking Accuracy (MOTA) with a focus on the detector used.

Faster R-CNN. As the name suggests, Faster R-CNN is a modified version of Fast R-CNN and R-CNN, as it introduces a Region Proposal Network (RPN)[4] that produces primitive proposals for regions that may contain objects from input feature maps. RPN eliminates the need for the sliding window approach in its predecessors, a process that examines a high density of the input to produce proposals. As a result Faster R-CNN sees a significant speedup from its predecessors while scoring higher in mAP[5], with fps ranging from 5 to 17 fps depending on the network model used.

CompACT. Spanning from the idea of cascaded detector which also utilizes sliding windows, complexity aware cascade training (CompACT)[6] is a boosting algorithm that seeks to optimize accuracy and complexity by pushing features of higher complexity to later stages. CompACT reaches a state of the art performance on both vehicle and pedestrian detection, as the consideration of complexity allows it to run well under image of different orders of magnitude.

CenterTrack. The sliding window technique is replaced entirely in the case of CenterTrack, as its detector, CenterNet, detects objects as heatmap peak points and regresses to form the best bounding box[7]. As a result, CenterTrack is able to produce more accurate bounding boxes at 22 to 28 fps[8].

II. Crop CenterTrack

A. Motivation

The choice of CenterTrack as the baseline model is not arbitrary, as CenterTrack only involves one stage of feature extraction and yields accurate bounding boxes based on the heatmap instead of sliding windows. The simple structure of CenterTrack makes it easily extendable with crop-based extension.

The height of I-24 traffic monitoring cameras dissipates the concern of occlusion, as with cameras placed high up on poles would provide a near bird's-eye view, where one vehicle blocking the view of another is unlikely. The context would also allow us to safely assume that the objects will maintain a smooth and continuous motion inside the frame until they exit the view at the horizon and not disappear from the frame suddenly.

B. General Idea

Almost all MOT models perform detection over the entire input image and hence fail to incorporate tracking information of previous frames. Crop-based detection instead uses the bounding box location of the track objects in *a priori* frame to predict the region the object is likely to be in in the next frame, and only performs detection on these smaller individual regions, ignoring the most part of the background.

By specifically looking at the likely regions of the tracked objects, the detection step takes less time as only crops of the image are passed into the network. The amount of speedup depends on the number of objects in *a priori* frame, while more objects would lead to more total area of crops that need to be passed.

C. Implementation

The code implementation mainly involves developing a second mode of detection and tracking step: the crop mode. While the baseline mode, or full frame mode, happens every *k* frames, the rest of the frames are run on crop frames. Testing results for different choices of *k* will be discussed later in this paper.

During crop mode frames at j th frame, with a given bounding box, $box_{i,j-1}$ of a tracked object in *a priori* frame, the prediction is made by taking the center of $box_{i,j-1}$, $(x_{i,j-1}, y_{i,j-1})$, and the maximum of the box height and width $d_{i,j-1}$. The predicted region of the object is calculated as:

$$scale_{i,j-1} = d_{i,j-1} * 1.5 \tag{1}$$

$$pred_{i,j-1} = (x_{i,j-1} - scale_{i,j-1}, y_{i,j-1} - scale_{i,j-1}, x_{i,j-1} + scale_{i,j-1}, y_{i,j-1} + scale_{i,j-1}) \tag{2}$$

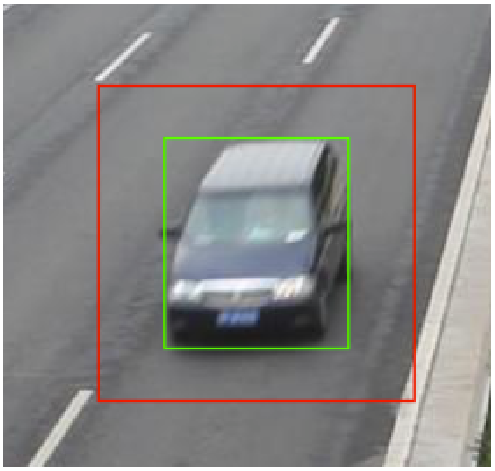


Figure 1: green box is the bounding box for the car in current frame, and red box will be the predicted region of the car in the next frame, if it were to be a crop frame.

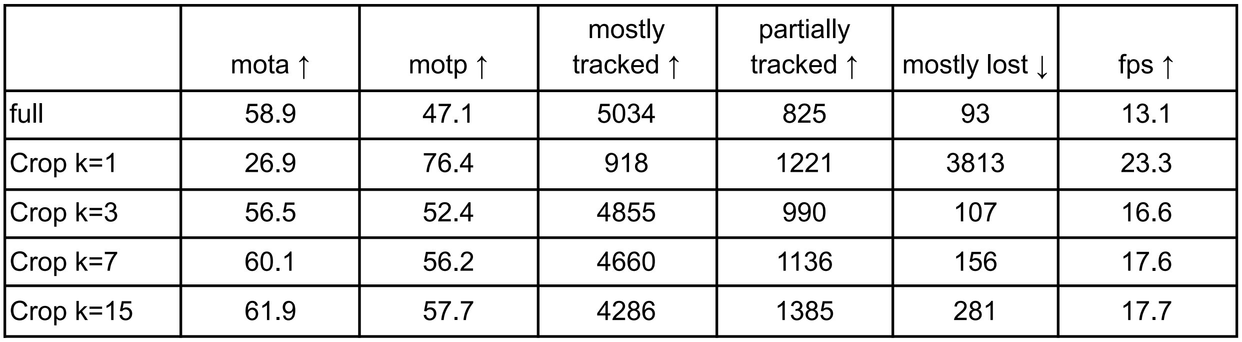
Then the predicted regions are cropped from the image and passed into the network instead of the entire image. The detector scores the proposals based on a combination of their confidence levels and intersection over union (IoU), and picks the top scoring proposal of the region as the detected object if the score passes a threshold value 0.3. The choice of the specific value for the weight is thanks to Gloudemans' testing in his extension on KIOU[1]. In the below equation, $prop_{i,j}$ refers to the j th proposal in the i th bounding box, with box_i being the bounding box for the i th tracked object.

$$score(prop_{i,j}, box_i) = 0.8 * conf(prop_{i,j}) + 0.2 * IoU(prop_{i,j}, box_i) \tag{3}$$

In the tracking step, the tracker associates each top scoring proposal with the object from *a priori* frame if its score is at least 0.3. If there are no proposals scoring higher than 0.3 for an object in *a priori* frame, then it is marked as not detected.

III. Testing Results

The objective of the testing is to find out whether the extension improves the performance of the baseline Crop-CenterTrack, and in what ways, if any. Testing was conducted focusing on Crop-CenterTrack's performances with different choices of *k*. In short, with a given *k*, full frame mode will take place every k frames, while the rest of the frames will be crop frame. Tracking starts on a full frame detection. The benchmark was performed with coco tracking model pre-trained to epoch 70 on the entire UA-DETRAC dataset. The UA-DETRAC datasets include videos of 540 x 940 in size and all the crops are resized to 64 x 64.



	mota ↑	motp ↑	mostly tracked ↑	partially tracked ↑	mostly lost ↓	fps ↑
full	58.9	47.1	5034	825	93	13.1
Crop k=1	26.9	76.4	918	1221	3813	23.3
Crop k=3	56.5	52.4	4855	990	107	16.6
Crop k=7	60.1	56.2	4660	1136	156	17.6
Crop k=15	61.9	57.7	4286	1385	281	17.7

Figure 2: test on GTX 1080 with results averaged over three runs

Multiple object tracking accuracy (mota) indicates the overall accuracy of detection and tracking while multiple object tracking precision (motp) shows the precision of predicted bounding boxes. According to the test results, crop with $k = 1$ is a test anomaly that needs to be addressed, while the rest of the results show a consistent trend from which we could generalize a pattern.

In terms of accuracy, there is an increase in mota and motp as *k* increases and at *k* = 7 Crop-CenterTrack outperforms baseline CenterTrack. The baseline CenterTrack has the best mostly tracked and partially tracked than Crop-CenterTrack, as the latter has the inherent disadvantage of not being able to detect or track new objects that just entered the frame, since it will not perform detection on new objects except during full frame detection. However, it makes up for the disadvantage by keeping track of objects well. Aside from the increase in accuracy, Crop-CenterTrack also sees a speedup of about 30% compared to the baseline, thanks to the smaller image size. The gain in speed plateaus around 17.5 fps however, as between *k* = 7 and *k* = 15 there is an ignorable improvement. The test results indicate that for Crop-CenterTrack the most consistent and overall best value for *k* would be 7 as it has second best accuracy and speed, with least mostly lost objects.

One interesting observation on the side is that while there is an increase in speed, Crop-CenterTrack also uses less GPU utility, as when running CenterTrack the usage stays above 70% and the crop version GPU usage fluctuates between 25% and 65%. In theory, the observation could mean that Crop-CenterTrack is more suitable for running on over 300 cameras as it would require less computing power than the baseline.

III. Conclusion

We could conclude from the test results that the crop-based detection extension yields an improvement both in accuracy and speed on the baseline. Although testing on the actual footage from the I-24 Smart Corridor cameras is needed, the extension should see an even better improvement on the I-24 cameras as the cameras would be higher up than the ones in UA-DETRAC dataset and hence the issue of not initializing objects would be somewhat alleviated by the fact that the camera would cover a much longer portion of the highway, in which case keeping track of vehicles would be more important than initialization.

There is also room for improvement for the extension as well. A detector network more specialized in vehicle detection could be used in place of coco tracking. More tests on parameters other than the choice of *k* could also allow for an improvement, for example the crop size could be adjusted to see whether bigger crop size would yield higher accuracy or whether smaller crop size would speed up the model even further without compromising the accuracy.

Finally, together with Gloudemans' testing results on Crop-KIOU, crop-based detection is shown to have the potential to be a desired extension for MOT models intended for vehicle monitoring scenarios.

References

[1] Derek Gloudemans, Daniel B. Work. Vehicle Tracking with Crop-based Detection, 2021.

[2] Varga, Leon Amadeus and Kiefer, Benjamin and Messmer, Martin and Zell, Andreas. Seadronessee: A maritime benchmark for detecting humans in open water, 2022.

[3] Longyin Wen, Dawei Du, Zhaowei Cai, Zhen Lei, Ming-Ching Chang, Honggang Qi, Jongwoo Lim, Ming-Hsuan Yang, Siwei Lyu. UA-DETRAC: A New Benchmark and Protocol for Multi-Object Detection and Tracking, 2022.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, 2016.

[5] Ross Girshick. Fast R-CNN, 2015.

[6] Zhaowei Cai, Mohammad Saberian, Nuno Vasconcelos. Learning Complexity-Aware Cascades for Deep Pedestrian Detection, 2015.

[7] Xingyi Zhou, Vladlen Koltun, and Philipp Krahenbuhl. Objects as Points, 2019.

[8] Xingyi Zhou, Vladlen Koltun, and Philipp Krahenbuhl. Tracking Objects as Points, 2020.