# Vehicle Tracking with Crop-based Detection

***Abstract*—End-to-end production of vehicle tracking data from video in real-time and with high accuracy remains a challenging problem due to the computational cost of object detection on each frame. In this work we present *Tracking with Crop-based Detection*, a method for speeding object tracking in constrained contexts (with stable cameras and relatively-predictable object motion) such as vehicle traffic monitoring. We leverage this context to provide a strong prior for object locations, which we use to 1.) boost detection speed by detecting objects only in regions corresponding to object priors on most frames and 2.) inform the selection of the detector output for each object. We evaluate Crop-based Detection as an extension to the KIOU object tracker (Crop-KIOU) on the UA-DETRAC dataset. The proposed tracker outperforms all other reported algorithms in terms of PR-MOTA, PR-MOTP, and mostly tracked objects on the UA-DETRAC benchmark, establishing a new state-of-the-art. Relative to tracking by detection with KIOU, Crop-KIOU achieves a 26% higher frame-rate and increases accuracy. Furthermore, Tracking with Crop-based Detection can be combined with frame skipping; we show a 149% increase in framerate relative to KIOU with no decrease in accuracy using this combination of methods.**

***Index Terms*—Object tracking, object detection, traffic monitoring.**

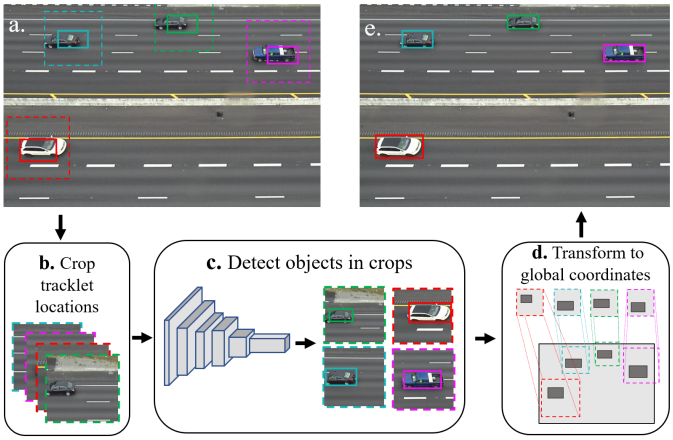


Fig. 1. Overview of Tracking with Crop-based Detection (proposed). **(a)** Tracklet *a priori* locations (solid boxes) are used to **(b)**. crop (dashed boxes) around likely object locations. **(c)**. Detection is performed on crops to localize each tracked object. **(d)**. The resulting bounding boxes (solid boxes) are then transformed back into the frame coordinate system, **(e)**. producing final object detections for each tracklet.

## I. Introduction

In this work we address the task of *multiple object tracking* (MOT) from raw video sequences in a traffic monitoring context. The goal of this task is to accurately localize and classify each vehicle within a traffic scene at each frame in the video, and to associate these bounding boxes across frames to provide matched identities for each unique vehicle across time. In particular, we concern ourselves with fixed traffic cameras with overhead fields of view (as in the UA-DETRAC dataset [1]), which is important but distinct from the self-driving car context (e.g., the KITTI dataset [2]) in which the camera moves and is taken from a vehicle-centric field of view. The real-time performance of object detection and tracking is paramount for a number of tasks in traffic modeling (a shortage of vehicle trajectory data is a persistent problem [3, 4]) and to enable next-gen *intelligent transportation systems* (ITS) that dynamically respond to real-time demand to better accommodate traffic [5, 6]. We argue that existing multiple object tracking methods are not well-posed to provide such traffic data in real-time.

The vast majority of algorithms for multiple object tracking decompose the problem into two distinct tasks: First, the *object detection* task locates relevant objects within a frame. Second, the *object association* (commonly referred to simply as object tracking) task associates or matches objects in the current frame with the same objects in the previous frame such that each object is uniquely identified across the entire video sequence. Importantly, though, the most accurate tracking by detection methods still run below 30 frames per second on a GPU for frames of modest size (e.g., 960×540 [7], 1392×512 [2], and 1920×1080 [8]. A variety of recent methods [9–16] have sought to leverage the generic tracking context to provide additional information for the object detection task, performing detection and tracking *jointly* rather than in series. These methods make use of the the relationship between objects in consecutive frames to boost object detection and tracking accuracy rather than speed. Unfortunately, this increased accuracy is not realizable if real-time requirements must be met.

In this work, we leverage the traffic domain to further increase the speed of object detection and tracking. Relative to other object tracking contexts, vehicles have predictable motion. They travel roughly along lane lines and undergo relatively small accelerations. This contextual information, combined with the fact that traffic monitoring cameras are static, means that extremely strong priors for object locations

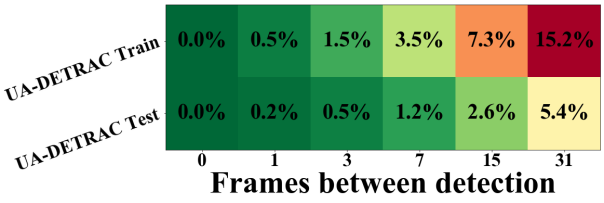


Fig. 2. Expected increase false negative rate due to new undetected objects appearing between detection steps based on average object longevities for UA-DETRAC training and testing datasets. If every $d$ frames is fully detected, on average a new object is missed in $\frac{d}{2}$ frames after it initially enters the field of view.

within a video frame are available before that frame is processed. Generalized object tracking methods [9–28] cannot take advantage of object priors to reduce their object search space because they are intended to also track in contexts where camera and object motion is unpredictable.

By contrast, our method (summarized in Figure 1) leverages this context to reduce the search space over which objects are detected within a frame. We make use of object priors from previous frames (Figure 1(a)) by (b) cropping portions of the frame we expect to contain existing tracked objects. Within each crop, (c) we solve the task of *localization* (identifying the location of a single object of interest) with an object detector trained specifically on images of this smaller crop size. We make use of object priors to weight each output *before* removing any outputs from consideration as would be done in NMS or soft NMS [29]. The selected object bounding boxes are (d) converted into their corresponding locations within the overall frame. We call this method *Tracking with Crop-based Detection*.

Because detection is not run on every full frame, there is a potential to increase the number of false negatives due to missed detections when objects first appear. However, Figure 2 shows that the expected increase in the false negative rate is small for real-world datasets. Moreover, we show in Section IV that crop-based tracking improves overall tracking performance because of a dramatic reduction in false positives that appear when performing detection on every frame.

The primary contributions of this work are as follows. First, in constrained tracking contexts such as traffic monitoring where cameras are static and object motion is predictable, we show that detection can be performed on a limited set of crops from each frame without a loss in overall tracking accuracy. Second, we demonstrate that crop-based detection can significantly speed the joint object detection and tracking process. We implement and evaluate a crop-based detector which we call Crop-KIOU. We demonstrate on the UA-DETRAC benchmark, the most widely used benchmark for tracking in a traffic monitoring context, that Crop-based Detection increases the speed of the baseline tracker without compromising on tracking accuracy (MOTA), and Crop-KIOU outperforms all existing methods on this benchmark in terms of PR-MOTA, PR-MOTP, and mostly tracked objects. Further, we show that Crop-KIOU extends the pareto-frontier of the speed-accuracy tradeoff for KIOU on this benchmark.

The remainder of this article is organized as follows. In Section II, existing approaches to multiple object tracking are reviewed. Section III introduces Tracking with Crop-based Detection and describes the implementation of Crop-KIOU. Section IV details the experiments used evaluate Crop-KIOU.

## II. Background

Multiple object tracking algorithms such as [9–28] can be roughly divided into three main categories: tracking by detection methods, integrated object detection and object association, and adaptations of single object trackers. We briefly review prominent methods in each category.

**Tracking by Detection**. Most modern object trackers follow the tracking by detection framework [30, 31] This paradigm divides the task of producing tracked objects from raw video into two steps. First, relevant objects are detected in each frame. Then, detected objects are associated across frames [19]. This subdivision of tasks into detection and tracking is encouraged by popular benchmark datasets for object tracking such as MOTChallenge [32], KITTI [2], and UA-DETRAC [1].

**Object Detection**. Most top-performing object detectors rely on *convolutional neural networks* (CNNs) for feature extraction from an image. Common approaches include one stage detection, such as in YOLO [33] and RetinaNet [34], where features output by a convolutional neural network are directly used to regress object bounding box coordinates. Two stage detectors such as Faster RCNN [35] and Evolving Boxes [36] attempt to boost accuracy by adding an intermediate step in which promising candidate regions are selected, and then only from these regions are bounding boxes regressed. Segmentation models such as Mask-RCNN [37] have also been adapted to perform bounding box-based detection. Recently, these approaches have been bolstered by adding additional awareness of foreground and background [38], by use of attention networks [39], or by regressing the location of keypoints such as bounding box corners [40] or object centers [41] with custom pooling layers that better convey keypoint information through convolutional layers. Importantly, though, the best-performing algorithms in terms of detection accuracy still run below 30 frames per second on a GPU for frames of modest size (e.g., 960×540 [7], 1392×512 [2], and 1920×1080 [8]).

**Object Association**. Object association methods compare objects from sequential frames in terms of position, appearance, and/or physical dynamics to match objects from one frame to the next. In *Simple Online Realtime Tracking* (SORT) [17], Kalman filtering is used to predict object locations and these predicted positions are matched to current frame detections based on straight-line distance. DeepSORT [18] refines SORT by additionally using an appearance embedding for each object to aid matching. The IOU tracker [19] utilizes bounding box overlap rather than straight-line distance as the distance metric, and KIOU [21] combines this method with Kalman filtering for more accurate matching. *Continuous Energy Minimization* (CEM) combines bounding box overlap ratio, color-based appearance dissimilarity, object physical dynamics, and logical

constraints to match objects [22]. Other successful recent approaches for cross-frame association leverage CNN-based reID features and [23, 42, 43] or otherwise explicitly regress future object positions from a frame using CNNs [13]. Weakly supervised methods such as SimpleReID [25] have also been successful for tracking tasks.

**Single Object Tracking Methods for MOT**. Recently, a few works have approached multiple object tracking as a set of parallel *single object tracking* (SOT) tasks. The parallel SOT task has been posed as a Markov Decision Process [26], or handled using a CNN with single-target-specific branches that utilize shared features [27]. The work proposed in [28] combines both object detector detections and rough single object tracking object positions to refine object position estimates. Similarly, VIOU [20] extends IOU tracking by employing a single object tracker to recover lost objects, reducing fragmentations. Thus far, SOT-based methods for multiple object tracking have not been scalable in terms of speed, due either to 1.) the need to update appearance models for each object online or 2.) the need to initialize new objects, which has been addressed by additionally performing detection on each frame [44].

**Joint Detection and Tracking.** Some tracking approaches have integrated information from the tracking context for object detection. Some pass pairs [9] (or larger sequences [10, 11]) of consecutive frames to CNNs to regress detections across multiple frames. [12] passes the previous frame as well as a "heatmap" of previous object positions to the object detector which predicts object locations and offsets used to aid greedy matching of objects. Other works perform tracking by object re-detection, passing previous inputs to an object detector as additional anchor boxes [13, 14]. [15] uses the same CNN to output bounding box coordinates and object embeddings for re-identification and matching. [16] regresses bounding box pairs for consecutive frames, performing IOU-based matching on the two sets of bounding boxes corresponding to each single frame. Recently, graph neural networks have also been used to leverage spatial-temporal relationships for both object detection and object association [45], and transformer networks have also been used to detect and track jointly [46].

Different from existing SOT and joint detection and tracking works, our approach is the first to perform detection only over a limited set of crops rather than each overall frame. Existing trackers in these categories utilize object priors to refine object detection and boost accuracy, rather than to increase the speed of object detection. While our approach is a joint detection and tracking method, the primary focus of our tracker extension is on faster rather than more accurate performance.

## III. Methodology

We detail the process for modifying the popular Kalman-filter enhanced Intersection-over-Union (KIOU) tracker [19, 21] to rely on Crop-based Detection in this section. We call this modified tracker Crop-KIOU. We note that Crop-based Detection is applicable to many other trackers following the tracking by detection paradigm.

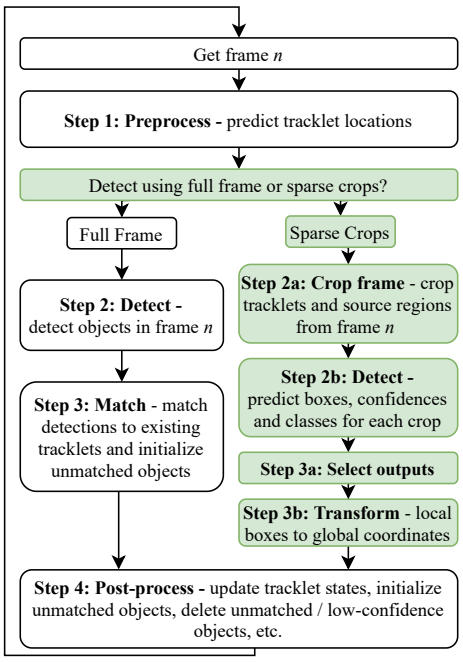


Fig. 3. Crop-based Detection (green) used to extend the base tracker (white).

### A. KIOU Tracker

A graphical overview of KIOU is shown in Figure 3 (white boxes). A Kalman Filter models the motion (with constant velocity assumptions in pixel-space) of each tracked object such that (Step 1) the position of each object in frame $n$ can be predicted *a priori* (before detection is performed). New objects are detected by (Step 2) running any object detector on the whole frame $n$. Detected objects in frame $n$ are (Step 3) matched to existing object tracklets by bipartite matching between the two sets, where intersection-over-union metric between tracklet *a priori* locations and new detections used as the distance metric between object pairs. Then, (Step 4) matched detections are used to update the estimated locations of each corresponding object tracklet within the Kalman Filter. Objects in frame $n$ without a suitable match are initialized as new tracked objects and objects from frame $n-1$ without a suitable match in frame $n$ are marked as inactive; after a number of inactive frames, objects are considered lost and are no longer tracked.

### B. Crop-KIOU Tracker

Figure 3 shows the added steps of Crop-KIOU in green boxes. Rather than performing (slow) object detection on every full frame, Crop-KIOU skips $d$ frames before performing frame object detection, where $d$ is a tuneable parameter. On detection frames, Crop-KIOU is identical to KIOU. On all other frames, (Step 2a) a set of crops is selected from the overall frame. Specifically, the estimated *a priori* location of each existing object is used to crop a portion of the frame around that object within the the overall frame. (Step 2b) These small crops are then processed by a crop detector (identical to

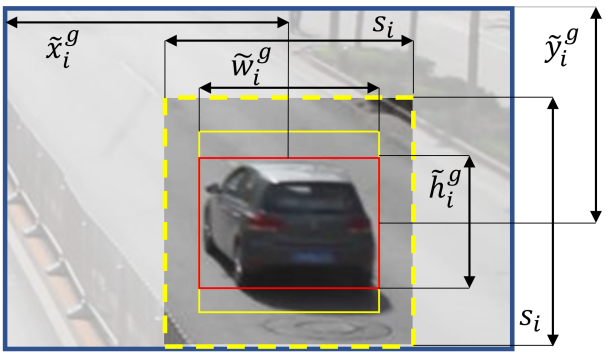


Fig. 4. *a priori* object $i$ location in global (frame) coordinates, $\tilde{\text{box}}_i^g = [\tilde{x}_i^g, \tilde{y}_i^g, \tilde{w}_i^g, \tilde{h}_i^g]$ (red), is made square (yellow solid) and expanded by a factor of $\beta$ to produce $\text{crop}_i := [\tilde{x}_i^g, \tilde{y}_i^g, s_i]$ (yellow dash) before being passed to crop detector. Expansion helps to ensure that actual tracked object will be contained within the crop area.

the detector but retrained on images of the expected crop size). All non-cropped portions of the image are ignored. The "best" box from each object crop is selected as the detection for that crop (Step 3a) . Because each object crop is already associated to an object when it is generated, there is no need to perform a (possibly error prone) data association step. All detections within crops can trivially be transformed back into the full frame coordinate space (Step 3b.). Lastly, object tracklets are updated within the Kalman Filter based on the set of detections as in KIOU. Each step new to Crop-KIOU is detailed more is subsequent sections.

**Step 2a: Crop frame.** Generate a square cropping box centered on each *a priori* object location in frame $n$. To ensure that the full object is contained within the crop, expand the crop to be larger than the a priori object estimate.

More precisely, let $\mathcal{O} := \{1, \cdots, i, \cdots, o_{max}\}$ be the set of all tracked objects, indexed by $i$. The *a priori* (denoted by tilde) bounding box for object $i$ in global (frame) coordinates is defined by the center x-coordinate $\tilde{x}_i^g$, the center y-coordinate $\tilde{y}_i^g$, the width $\tilde{w}_i^g$ and height $\tilde{h}_i^g$. Let $\tilde{\text{box}}_i^g := [\tilde{x}_i^g, \tilde{y}_i^g, \tilde{w}_i^g, \tilde{h}_i^g]$. Similarly, we define the corresponding square crop for object $i$ as $\text{crop}_i := [\tilde{x}_i^g, \tilde{y}_i^g, s_i]$, where $s_i$ is the scale. The scale is computed as:

$$s_i = \max\{\tilde{w}_i^g, \tilde{h}_i^g\} \times \beta, \tag{1}$$

where $\beta$ is a box expansion ratio (a parameter) used to ensure the full object is within the crop. Figure 4 shows a graphical representation of crop generation for a single object.

By construction, $\text{crop}_i$ is of size $(s_i \times s_i)$ pixels. Before detection, each crop re-scaled to a size $(C \times C)$ pixels, where $C$ is a constant across all crops.

**Step 2b: Detect in Crops.** All image crops corresponding to *a priori* object locations are processed by the crop detector, which produces bounding boxes that estimate the location of the object within each crop. In this work, a Retinanet detection network [34] is used for both the full frame detector and the crop detector, with each trained separately on data with images of the corresponding scale. Given $\text{crop}_i$, the localizer returns $l_{max}$ bounding boxes indexed by $j$ in the local crop coordinates. Each output is an estimated location of object $i$ within the crop, defined by the object center, box width, and box height. The $j$-th bounding box output of the detector corresponding to $\text{crop}_i$ is written as $\text{box}_{i,j}^l := [x_{i,j}^l, y_{i,j}^l, w_{i,j}^l, h_{i,j}^l, \text{conf}_{i,j}]$, where $\text{conf}_{i,j} \in [0, 1]$ is the confidence of the $j$-th detector output associated with $\text{crop}_i$.

**Step 3a: Select outputs.** We score each box with a weighted combination of detection confidence and IOU overlap with the object prior (IOU+Conf), thus incorporating information from the prior before removing any candidate boxes. Section IV-B describes the experiments motivating this choice. Each candidate bounding boxes is scored according to this IOU+Conf metric defined as:

$$\text{score}(\text{box}_{i,j}^l, \tilde{\text{box}}_i^l) = W \times \text{conf}_{i,j} + (1 - W) \times \Phi(\text{box}_{i,j}^l, \tilde{\text{box}}_i^l), \tag{2}$$

where $\Phi$ is the IOU similarity function between two boxes and $W$ is a scalar used to balance the two terms. The bounding box with the highest score is selected as the detected $\text{box}_i^l$ for object $i$.

The best detector output corresponding to $\text{crop}_i$ is written as $\text{box}_i^l := [x_i^l, y_i^l, w_i^l, h_i^l]$, in coordinates local to the crop. Since the set of detection outputs for a crop are compared to the single *a priori* object $i$'s location, output selection across all objects is $\mathbf{O}(o_{max} \times l_{max})$ in complexity, where $o_{max}$ is the total number of tracked objects and $l_{max}$ is the total number of detection outputs per crop. This operation is significantly less complex than a $\mathbf{O}(o_{max}^3)$ global min-cost matching problem in Step 3 of the base tracker [47]. Moreover it avoids object association errors that can occur in Step 3.

**Step 3b: Local to global transformation.** The best detection $\text{box}_i^l$ for each crop $i$ is converted back into global coordinates, where it can be used to update the $i$-th tracklet. Step 4 is then performed as for the base KIOU tracker.

## IV. Experiments

We perform experimental analysis using the UA-DETRAC Benchmark, the most comprehensive object tracking dataset in a traffic monitoring context. We first test our proposed IOU+Conf method for parsing crop-detection outputs against existing strategies (NMS and Soft NMS). Second, we test Crop-KIOU at a variety of parameter settings for $d$ and assess its performance relative to baseline KIOU. We next evaluate Crop-KIOU on the UA-DETRAC test benchmark where we achieve state of the art performance. Finally, we compare the use of Crop-based Detection to speed object tracking to the current standard method for speeding tracking (frame-skipping).

### A. UA-DETRAC

The UA-DETRAC Benchmark Suite contains 10 hours of video containing traffic sequences divided into 60 training and 40 testing videos. The training and test data contain an average of 7.1 and 12.0 objects per frame, respectively [1]. We further subdivide the training data into 45 training videos and 15

| d | Hz ↑ | PR-MOTA ↑ | PR-MOTP ↑ | PR-MT ↑ | PR-ML ↓ | PR-IDS/id ↓ | PR-FM/id ↓ | PR-FP/obj ↓ | PR-FN/obj ↓ |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 22.7 / 22.9 | 66.4 / 55.8 | 77.5 / 70.7 | **87.7% / 72.0%** | **3.9% / 8.6%** | 0.27 / 0.45 | **0.27 / 0.69** | 0.192 / 0.220 | **0.121 / 0.189** |
| 1 | 26.4 / 26.7 | **69.4 / 59.7** | **79.9 / 76.8** | 73.4% / 53.7% | 6.7% / 14.3% | 0.51 / 0.64 | 0.90 / 1.35 | 0.093 / 0.090 | 0.190 / 0.275 |
| 3 | 28.5 / 29.1 | 67.2 / 58.0 | 78.1 / 75.7 | 70.0% / 50.7% | 7.3% / 14.9% | 0.43 / 0.53 | 1.13 / 1.39 | 0.095 / 0.094 | 0.208 / 0.287 |
| 7 | 31.3 / 31.9 | 63.9 / 56.6 | 76.6 / 75.4 | 63.3% / 44.4% | 8.4% / 16.5% | 0.35 / 0.39 | 0.95 / 1.15 | 0.098 / 0.092 | 0.236 / 0.306 |
| 15 | 32.5 / 33.1 | 60.7 / 53.6 | 76.0 / 74.9 | 52.2% / 34.9% | 10.9% / 20.6% | 0.20 / 0.23 | 0.67 / 0.85 | 0.096 / 0.089 | 0.272 / 0.338 |
| 31 | **34.6 / 35.7** | 55.6 / 47.6 | 75.7 / 74.2 | 36.8% / 25.2% | 17.5% / 28.1% | **0.10 / 0.12** | 0.49 / 0.65 | **0.088 / 0.084** | 0.337 / 0.397 |

TABLE I

TRACKING METRICS (TRAINING/VALIDATION) FOR KIOU ($d = 0$) AND CROP-KIOU ON UA DETRAC DATASET. BEST RESULTS FOR EACH METRIC ARE SHOWN IN BOLD. WHEN APPLICABLE, METRICS ARE NORMALIZED BY NUMBER OF UNIQUE OBJECTS (ID), OR NUMBER OF UNIQUE OBJECT OCCURRENCES (OBJ)

validation holdout videos. We train all frame and crop detectors using the 45 training videos, and perform experimental analysis on each data partition (training/validation) separately. We evaluate Crop-KIOU according to the PR-metrics defined in [1] which evaluate tracking performance at varying levels of detection confidence levels.

### B. Evaluation of bounding box selection method

We test three strategies for selecting from amongst output bounding boxes for each frame. 1.) Perform *non-maximal suppression* (NMS) and subsequently select the remaining box with the highest IOU with the object prior location, evaluated at multiple threshold overlaps $N_t$. 2.) Same as 1, but use Soft-NMS instead of NMS, evaluated at several Gaussian factors $\sigma$ [29]. 3.) The proposed IOU+Conf box-scoring described in Section III evaluated at several weighting parameters $W$.

To evaluate each method we generate a set of crop-detector outputs for each of 500 object image inputs. We provide an associated prior for use in bounding box selection by adding random noise to the ground truth bounding box for each object. We use three levels of noise such that the object priors have an average IOU of 0.85, 0.75, and 0.60, respectively, with the ground truth bounding box. We use as comparison metric between methods the average IOU of the final selected bounding box with the ground truth bounding box, averaged over all examples. Figure 5 shows the result.

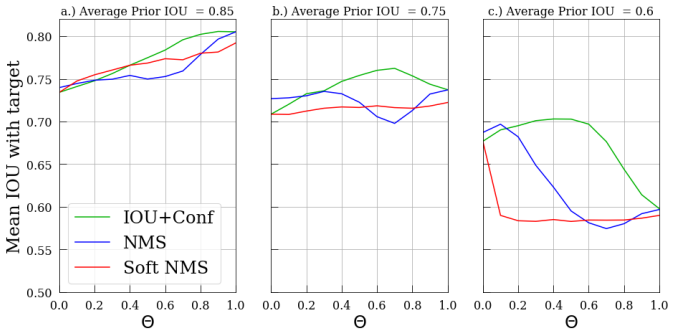


Fig. 5. Mean IOU of ground truth and bounding box selected with each method, given object priors that overlap with ground truth by a.) 0.85, b.) 0.75, and c.) 0.60 on average. $\theta$ is a stand-in variable for the changeable parameter for each method ($N_t$ for NMS, $\sigma$ for soft NMS, and $W$ for IOU+Conf).

The proposed IOU+conf approach outperforms both NMS and soft NMS across a wide range of parameter settings, and at all tested levels of object prior accuracy. The best overall output IOU (0.81,0.76, and 0.70 respectively) is achieved using IOU+Conf for each object prior accuracy condition. Intuitively, when the quality of the prior is lower (in terms of average IOU), the optimal value for $W$ is lower, as it places less emphasis on overlap with the prior and more emphasis on high detector confidence. IOU+Conf shows promise over NMS and SoftNMS in constrained tracking contexts because the prior is incorporated earlier into the box selection process before any candidate boxes are deleted.

### C. Crop-KIOU versus KIOU baseline

We evaluate Crop-KIOU for multiple object tracking on the UA-DETRAC training and validation partitions with varying numbers of frames between detection $d$. Tracking is performed at $d = 0, 1, 3, 7, 15$ and 31 frames. Note that $d = 0$ is the baseline (KIOU) because detection is performed on every full frame. Results are reported in Table I.

As seen in Table I, Crop-KIOU achieves increased accuracy (PR-MOTA) and increased frame-rate relative to the base tracker. When the number of frames between detection is small ($d = 1$ and $d = 3$), Crop-based Detection increases the overall accuracy (PR-MOTA) of KIOU by drastically reducing the number of false positives (PR-FP). Average tracking precision (PR-MOTP) is also increased, meaning more tracklets output by the tracker correspond to ground truth objects. Most importantly, Crop-based Detection also results in a speedup relative to the base tracker. At $d = 3$, a 25%/27% (train/validation) speedup relative to baseline is achieved in addition to an increase in accuracy. When $d = 7$ a 38%/39% speedup is achieved for a -2.5%/+0.8% change in PR-MOTA relative to the base tracker. This increase in speed does come at a slight penalty: False Negative rate (PR-FN), fragmentations (PR-FM), and identity switches (PR-IDS) all at first increase with increasing $d$. However, we find that these types of errors are more desirable than false positives in a traffic monitoring context. Since object motion is fairly regular, we can often impute missing object positions during post-processing, whereas false positive tracked objects that persist for several frames are more difficult to identify as anomalous.

Based on these results, we select $d = 7$ as the best parameter setting for Crop-KIOU as it results in the largest speedup while still increasing accuracy in terms of PR-MOTA, and we use this setting for evaluation on the UA-DETRAC test data.

| Tracker | PR-MOTA ↑ | PR-MOTP ↑ | PR-MT ↑ | PR-ML ↓ | PR-IDs* ↓ | PR-FM* ↓ | PR-FP*↓ | PR-FN*↓ |
|---|---|---|---|---|---|---|---|---|
| GOG | 23.9 / 11.7 | 47.4 / 34.4 | 20.5% / 10.8% | 21.0% / 21.1% | 0.0158 / 0.0124 | 0.0148 / 0.0119 | 0.119 / 0.123 | 0.70 / 0.70 |
| IOUT | 34.0 / 16.4 | 37.8 / 26.7 | 27.9% / 14.8% | 20.4% / 18.2% | 0.0109 / 0.0084 | 0.0115 / 0.0089 | **0.031 / 0.061** | 0.64 / 0.66 |
| JTEGCTD | 28.4 / 14.2 | 47.1 / 34.4 | 23.1% / 13.5% | 18.3% / 18.7% | 0.0013 / 0.0020 | 0.0050 / 0.0065 | 0.096 / 0.127 | 0.63 / 0.65 |
| JDTIF | - / 28.0 | - / 41.8 | - / 34.2% | - / 20.9% | - / 0.0034 | - / 0.0166 | - / 0.270 | - / 0.73 |
| MFOMOT | 34.6 / 14.8 | 46.6 / 35.6 | 30.2% / 11.9% | 12.0% / 20.8% | 0.0040 / 0.0042 | 0.0091 / 0.0098 | 0.073 / 0.103 | 0.52 / 0.73 |
| KIOU | 40.1 / 31.0 | 49.8 / 49.9 | 42.3% / 37.4% | **5.8% / 10.4%** | 0.0021 / 0.0035 | 0.0024 / 0.0048 | 0.165 / 0.253 | **0.25** / 0.46 |
| V-IOU | 37.9 / 29.0 | 41.7 / 35.8 | 38.1% / 30.1% | 24.7% / 22.2% | **0.0004 / 0.0007** | **0.0008 / 0.0012** | 0.073 / 0.069 | 0.66 / 0.70 |
| DMC | - / 14.6 | - / 34.1 | - / 11.6% | - / 20.6% | - / 0.0044 | - / 0.0062 | - / 0.078 | - / 0.68 |
| GMMA | - / 12.3 | - / 34.3 | - / 10.8% | - / 21.0% | - / 0.0030 | - / 0.0117 | - / 0.124 | - / 0.70 |
| SCTrack-3L | 25.9 / 12.1 | 47.2 / 35.0 | 15.0% / 7.7% | 20.6% / 24.8% | 0.0017 / 0.0018 | 0.0062 / 0.0046 | **0.047 / 0.040** | 0.74 / 0.79 |
| Crop-KIOU (Ours) | **64.5 / 46.4** | **79.3 / 69.5** | **50.1% / 41.1%** | 8.2% / 16.3% | 0.0028 / 0.0051 | 0.0091 / 0.0186 | 0.061 / 0.113 | 0.26 / **0.44** |

TABLE II
TRACKING METRICS FOR UA-DETRAC TEST DATA (BEGINNER/ADVANCED) PARTITIONS. A – INDICATES THE RESULT IS NOT AVAILABLE. RESULTS TAKEN FROM [7].

* PR-IDs,PR-Frag,PR-FP and PR-FN are normalized by the total number of ground truth object detections.

### D. UA-DETRAC testing results

Next, we present the result of Crop-KIOU compared to state of the art methods for the UA-DETRAC Benchmark test dataset as reported in [7]. Table II shows that Crop-KIOU outperforms all existing methods both on overall accuracy (PR-MOTA) and tracking precision (PR-MOTP). Additionally, Crop-KIOU performs best overall in terms of mostly tracked objects (50.1% and 41.1% PR-MT on beginner and advanced subsets, respectively.) Notably, Crop-KIOU has the lowest rate of false negatives (PR-FN) of any tracker on the advanced test dataset partition, and the second lowest PR-FN rate on the beginner partition. This means that, despite missing some new objects as they appear, Crop-KIOU tracks known objects accurately enough to produce very few false negatives, more than making up for missed new objects.

Using Crop-based Detection to extend KIOU establishes a new state-of-the-art for this benchmark. Further, Crop-KIOU process at an average of 29.1 fps (realtime for this 25 fps benchmark). This speed is not directly comparable to the other reported trackers as it includes the time of object detection.

### E. Comparison to frame skipping

We compare CBT against the main existing method for speeding up object trackers. *Frame skipping* involves skipping $s$ frames between frame detections, and imputing tracked object locations between these frames. We evaluate KIOU with frame skipping ($s = 1, 2, 3, 4$ and $5$) against Crop-KIOU results shown above. We also combine frame-skipping with Crop-KIOU by skipping $s$ frames between any detection and performing a varying number of crop-based detections between full frame detections. We evaluate each method on the UA-DETRAC test dataset at a single detector confidence threshold and use the *Multiple Object Tracking Accuracy* (MOTA) metric defined in [48] rather than PR-Metrics as in [7]. This disentangles the speed of each method from the slowdown caused by parsing and matching large numbers of detections when a low confidence threshold is used (a low confidence threshold is necessary for PR-Metric calculation, but is not generally used in real-world fast tracking applications). Results are shown in Figure 6.

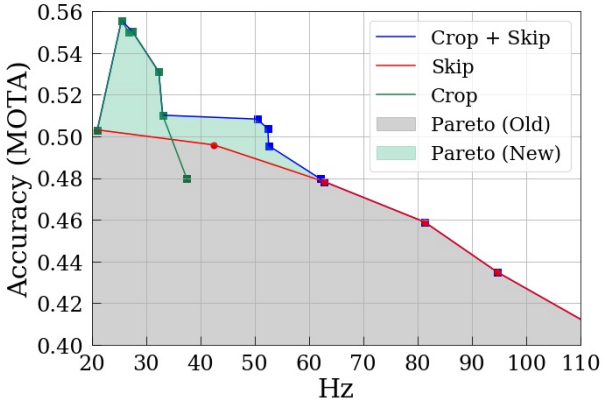


Fig. 6. Comparison of tracking algorithm extensions. Each point represents tracking results at a single parameter setting. Crop-based Detection (green) and Crop-based Detection + Skipping (blue) extend the state of the art (red) in terms of the tradeoff between accuracy (MOTA) and speed (Hz).

Crop-based Detection (green) extends the pareto-frontier of the accuracy-speed tradeoff for this traffic monitoring dataset. While frame-skipping (red) results in large speedups but exclusively reduces accuracy relative to the base tracker, Crop-based Detection increases accuracy considerably (50.3 to 55.6 MOTA) while also increasing speed by 21% relative to the baseline tracker, or increases speed by 57% without decreasing accuracy. Furthermore, the combination of Crop-based Detection and frame-skipping (blue) results in even larger speedups (21.0 to 52.4 fps or 149%) without a decrease in overall accuracy relative to the KIOU baseline. This above real-time performance allows for multiple traffic cameras to be processed simultaneously on the same device in real-time.

## V. CONCLUSION

This work presented *Tracking with Crop-based Detection*, a powerful technique for extending existing object trackers to boost detection and tracking speed. A tracker using Crop-based Detection (Crop-KIOU) was implemented and evaluated on the UA-DETRAC dataset, the leading traffic monitoring multiple object tracking benchmark. Crop-based Detection was

found to increase the speed of the baseline tracker (KIOU) by around 30% with no loss in PR-MOTA. Crop-KIOU achieved state of the art performance on the UA-DETRAC benchmark while producing tracklets from raw video at 29 fps. Moreover, Crop-based Detection was shown to compete and combine favorably with frame skipping, allowing for a 149% increase in framerate relative to KIOU with no decrease in accuracy.

This work establishes the potential of Tracking with Crop-based Detection to push the state of the art for object detection and tracking in real-time, constrained tracking applications such as traffic monitoring. Future work will use Crop-based Detection to extend trackers that utilize visual information for object association, and will further utilize 3D object detection and state estimation to more accurately localize vehicles in real-world coordinates as well as to predict vehicle motion more accurately. Code and full parameter settings for this work are available at https://github.com/DerekGloudemans/crop-tracking-detrac.

## Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. CNS-1837652 (Work) and by the National Science Foundation Graduate Research Fellowship under Grant No. DGE-1937963. This material is based upon work supported by the U.S. Department of Energy's Office of Energy Efficiency and Renewable Energy (EERE) award number CID DE-EE0008872. The views expressed herein do not necessarily represent the views of the U.S. Department of Energy or the United States Government.

## References

[1] 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.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[3] V. Alexiadis, J. Colyar, J. Halkias, R. Hranac, and G. McHale, "The next generation simulation program," *Institute of Transportation Engineers. ITE Journal*, vol. 74, no. 8, p. 22, 2004.

[4] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highd dataset: A drone dataset of naturalistic vehicle trajectories on german highways for validation of highly automated driving systems," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 2118–2125.

[5] J. Zhang, F.-Y. Wang, K. Wang, W.-H. Lin, X. Xu, and C. Chen, "Data-driven intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 4, pp. 1624–1639, 2011.

[6] Z. Yang and L. S. Pun-Cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments: A review," *Image and Vision Computing*, vol. 69, pp. 143–154, 2018.

[7] S. Lyu, M.-C. Chang, D. Du, W. Li, Y. Wei, M. Del Coco, P. Carcagnì, A. Schumann, B. Munjal, D.-H. Choi *et al.*, "Ua-detrac 2018: Report of avss2018 & iwt4s challenge on advanced traffic monitoring," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[8] P. Dendorfer, H. Rezatofighi, A. Milan, J. Shi, D. Cremers, I. Reid, S. Roth, K. Schindler, and L. Leal-Taixé, "Mot20: A benchmark for multi object tracking in crowded scenes," *arXiv:2003.09003[cs]*, Mar. 2020, arXiv: 2003.09003. [Online]. Available: http://arxiv.org/abs/1906.04567

[9] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3038–3046.

[10] B. Pang, Y. Li, Y. Zhang, M. Li, and C. Lu, "Tubetk: Adopting tubes to track multi-object in a one-step training model," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 6308–6318.

[11] S. Sun, N. Akhtar, X. Song, H. Song, A. Mian, and M. Shah, "Simultaneous detection and tracking with motion modelling for multiple object tracking," *arXiv preprint arXiv:2008.08826*, 2020.

[12] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *European Conference on Computer Vision*. Springer, 2020, pp. 474–490.

[13] P. Bergmann, T. Meinhardt, and L. Leal-Taixe, "Tracking without bells and whistles," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 941–951.

[14] W. Li, Y. Xiong, S. Yang, S. Deng, and W. Xia, "Smot: Single-shot multi object tracking," *arXiv preprint arXiv:2010.16031*, 2020.

[15] Z. Wang, L. Zheng, Y. Liu, Y. Li, and S. Wang, "Towards real-time multi-object tracking," *arXiv preprint arXiv:1909.12605*, 2019.

[16] J. Peng, C. Wang, F. Wan, Y. Wu, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, and Y. Fu, "Chained-tracker: Chaining paired attentive regression results for end-to-end joint multiple-object detection and tracking," in *European Conference on Computer Vision*. Springer, 2020, pp. 145–161.

[17] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[18] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[19] E. Bochinski, V. Eiselein, and T. Sikora, "High-speed tracking-by-detection without using image information,"

in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[20] E. Bochinski, T. Senst, and T. Sikora, "Extending iou based multi-object tracking by visual information," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2018, pp. 1–6.

[21] S. Chen and C. Shao., "Python implementation of the kalman-iou tracker." https://github.com/siyuanc2/kiout, accessed: 2021-03-12.

[22] A. Milan, S. Roth, and K. Schindler, "Continuous energy minimization for multitarget tracking," *IEEE transactions on pattern analysis and machine intelligence*, vol. 36, no. 1, pp. 58–72, 2013.

[23] Y. Zhan, C. Wang, X. Wang, W. Zeng, and W. Liu, "A simple baseline for multi-object tracking," *arXiv preprint arXiv:2004.01888*, 2020.

[24] Y. Xu, Y. Ban, X. Alameda-Pineda, and R. Horaud, "Deepmot: A differentiable framework for training multiple object trackers," *arXiv preprint arXiv:1906.06618*, 2019.

[25] S. Karthik, A. Prabhu, and V. Gandhi, "Simple unsupervised multi-object tracking," *arXiv preprint arXiv:2006.02609*, 2020.

[26] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4705–4713.

[27] Q. Chu, W. Ouyang, H. Li, X. Wang, B. Liu, and N. Yu, "Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4836–4845.

[28] X. Yan, X. Wu, I. A. Kakadiaris, and S. K. Shah, "To track or to detect? an ensemble framework for optimal selection," in *European Conference on Computer Vision*. Springer, 2012, pp. 594–607.

[29] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-nms–improving object detection with one line of code," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5561–5569.

[30] L. Fan, Z. Wang, B. Cail, C. Tao, Z. Zhang, Y. Wang, S. Li, F. Huang, S. Fu, and F. Zhang, "A survey on multiple object tracking algorithm," in *2016 IEEE International Conference on Information and Automation (ICIA)*. IEEE, 2016, pp. 1855–1862.

[31] W. Luo, J. Xing, A. Milan, X. Zhang, W. Liu, X. Zhao, and T.-K. Kim, "Multiple object tracking: A literature review," *arXiv preprint arXiv:1409.7618*, 2014.

[32] A. Milan, L. Leal-Taixé, I. Reid, S. Roth, and K. Schindler, "Mot16: A benchmark for multi-object tracking," *arXiv preprint arXiv:1603.00831*, 2016.

[33] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.

[34] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[35] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[36] L. Wang, Y. Lu, H. Wang, Y. Zheng, H. Ye, and X. Xue, "Evolving boxes for fast vehicle detection," in *2017 IEEE international conference on multimedia and Expo (ICME)*. IEEE, 2017, pp. 1135–1140.

[37] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[38] Z. Fu, Y. Chen, H. Yong, R. Jiang, L. Zhang, and X.-S. Hua, "Foreground gating and background refining network for surveillance object detection," *IEEE Transactions on Image Processing*, vol. 28, no. 12, pp. 6077–6090, 2019.

[39] H. Perreault, G.-A. Bilodeau, N. Saunier, and M. Héritier, "Spotnet: Self-attention multi-task network for object detection," in *2020 17th Conference on Computer and Robot Vision (CRV)*. IEEE, 2020, pp. 230–237.

[40] K. Duan, L. Xie, H. Qi, S. Bai, Q. Huang, and Q. Tian, "Corner proposal network for anchor-free, two-stage object detection," *arXiv preprint arXiv:2007.13816*, 2020.

[41] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6569–6578.

[42] Y. Zhang, C. Wang, X. Wang, W. Zeng, and W. Liu, "Fairmot: On the fairness of detection and re-identification in multiple object tracking," *arXiv e-prints*, pp. arXiv–2004, 2020.

[43] C. Liang, Z. Zhang, Y. Lu, X. Zhou, B. Li, X. Ye, and J. Zou, "Rethinking the competition between detection and reid in multi-object tracking," *arXiv preprint arXiv:2010.12138*, 2020.

[44] Q. He, J. Wu, G. Yu, and C. Zhang, "Sot for mot," *arXiv preprint arXiv:1712.01059*, 2017.

[45] Y. Wang, X. Weng, and K. Kitani, "Joint detection and multi-object tracking with graph neural networks," *arXiv preprint arXiv:2006.13164*, 2020.

[46] P. Sun, Y. Jiang, R. Zhang, E. Xie, J. Cao, X. Hu, T. Kong, Z. Yuan, C. Wang, and P. Luo, "Transtrack: Multiple-object tracking with transformer," *arXiv preprint arXiv:2012.15460*, 2020.

[47] L. R. Ford Jr and D. R. Fulkerson, *Flows in networks*. Princeton university press, 2015, vol. 54.

[48] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.