Vehicle Trajectory Prediction at Intersections using Interaction based Generative Adversarial Networks

Debaditya Roy, Tetsuhiro Ishizaka, C. Krishna Mohan, and Atsushi Fukuda

***Abstract*—Vehicle trajectory prediction at either signalized or non-signalized intersections with heterogeneous traffic is challenging. This issue becomes more aggravated when traffic is predominantly composed of smaller vehicles that frequently disobey lane behavior. Existing macro approaches consider the trajectory prediction problem in lane-based traffic that cannot cannot account for non-lane based traffic where there is high disparity in vehicle size and driving behavior among different vehicle types. Hence, we propose a vehicle trajectory prediction approach that models the interaction among vehicles. These interactions are encapsulated in the form of a social context embedded in a Generative Adversarial Network (GAN) to predict the trajectory of each vehicle at either a signalized or non-signalized intersection. The GAN model helps in producing the most acceptable future trajectory among acceptable choices that conform to past driving behavior. We evaluate the proposed approach on aerial videos of intersections from the benchmark VisDrone dataset. A comparison with existing trajectory prediction approaches establishes the efficacy of the proposed framework.**

***Index Terms*—Traffic flow prediction, Generative Adversarial Networks, Interaction Modeling**

I. Introduction

Vehicle trajectory prediction at any intersection requires detection and tracking of vehicles plying in various directions. Traditionally, sensors such as magnetometer detectors, loop detectors, ultrasonic sensors, and surveillance video cameras have been used to monitor intersections. However, these sensors are prohibitively expensive to set up and operate at all intersections. Particularly, surveillance video cameras that are being increasingly employed for traffic monitoring suffer from issues like occlusion, shadows, and a limited field of view. Although many techniques have been proposed to mitigate these challenges [1], [2], traditional surveillance cameras are still not viable for monitoring all the lanes in an intersection. In contrast, an Unmanned Aerial Vehicle (UAV) can be deployed as a cost-effective solution to monitor all the lanes of an intersection. Especially, with the availability of lightweight, high-resolution cameras, even small vehicles like motorbike can be captured in detail. Furthermore, UAVs provide a top-view perspective that is devoid of occlusion and shadows that makes aerial videos ideal for capacity analysis of intersections.

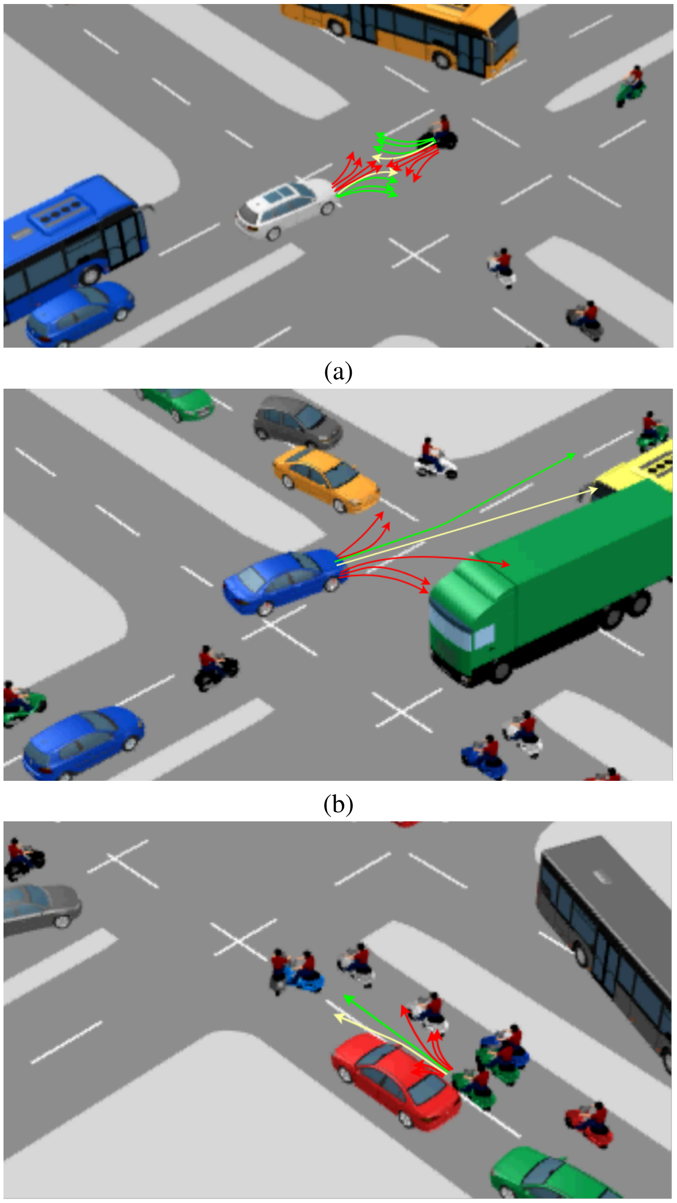
Vehicle trajectory prediction in aerial videos can be performed by detecting the different types of vehicles and tracking them individually throughout the intersection. This problem has been studied in the literature at a macroscopic level as homogeneous traffic flow (predominantly cars) [3] or heterogeneous traffic flow problems without considering motorbikes (or other two-wheelers) [4], [5] and with vehicles following lane discipline [6]. However, the traffic in developing countries contains a majority of two-wheelers that ply without any lane discipline, and the research is scant in this direction. Existing research in the area of mixed traffic analysis converts scooter, motorcycle, and bus flow to either equivalent passenger car units [5] or instead convert car and bus flows into equivalent scooter units [7]. Such a conversion does not represent the complex interactions among different types of vehicles especially in developing nations where the significant percentage of traffic volume is either buses or scooters. For example, motorcycles may maneuver between the gaps of large stationary vehicles stationary at a stop line to move to the front of the queue while in the presence of no gaps, a bus with its considerable size and slow acceleration may impede the progress of smaller vehicles behind it. To deal with such diverse scenarios, we propose an alternative system that predicts the best possible route for each vehicle given the possible interactions among vehicles possible in the future based on their proximity to each other. Some examples of the multiple prediction paths available during various vehicle maneuvers like overtaking and merging at intersections are presented in Figure 1.

Estimating the interactions between vehicles during the maneuvers shown in Figure 1 requires generating future trajectories that are aligned with the past behavior for each vehicle. The past behavior acts as a prior or condition for the future trajectories and hence, we propose to use a conditional Generative Adversarial Network (GAN) [8] that has been shown to generate multiple predictions from the same prior distribution. These predictions are then pruned based on its closeness to ground truth during training using a Markov Decision Process (MDP) to get a robust estimation of the trajectories from the vehicle detections. These tracks are then used to seed the GAN to produce useful predictions. The combinations of an MDP with the conditional GAN based on social interactions between vehicles shows impressive accuracy in predicting vehicle trajectories for different vehicle maneuvers like merging and overtaking in both signalized and non-signalized intersections. The main contributions of the paper are as follows:

- To the best of our knowledge, this is the first use of adversarial training for vehicle trajectory prediction for non-lane based traffic.

D. Roy, T. Ishizaka, A. Fukuda are with the Department of Transportation Systems Engineering, Nihon University, Chiba, 274-8501, Japan. C. Krishna Mohan is with the Department of Computer Science and Engineering, Indian Institute of Technology Hyderabad, India.

- The prediction mechanism is vehicle independent that does not use vehicle or road dimensions in order to predict trajectories
- The proposed GAN based solution can handle vehicle maneuvers like merging, overcoming, and avoiding oncoming traffic at intersections.


(a)

(b)

(c)

Fig. 1: Multiple prediction paths are available during various vehicle maneuvers (a) avoiding oncoming traffic, (b) overtaking, and (c) merging. The paths in green are acceptable while the ones in red are not. The paths in yellow are not optimal for the target vehicle but can be considered based on the decision taken by another vehicle (s). The choice for all vehicles are not shown for clarity. Simulations are obtained using PTV Vissim. Best viewed in color.

The rest of the paper is organized as follows. Section II reviews relevant literature in the area of multiple object detection and multiple object tracking. In Section III, the entire proposed approach is described in detail, and the evaluation results are presented in Section IV. Finally, the conclusion is presented in Section V.

II. Related Work

In recent years, several methods for traffic monitoring from aerial video have been presented. Most of these methods rely on either hand-crafted features like Scalar Invariant Feature Transform (SIFT) [9], or background subtraction [10], or motion history image [11] or using sparsity-based reconstruction [12]. However, in this paper, the traffic monitoring problem is treated as a Multiple Object Tracking (MOT), or Multiple Target Tracking (MTT) problem that involves the location of multiple objects and maintaining their trajectories throughout their presence in the input video. There are multiple challenges encountered in MOT mostly related to the drifting of tracking points due to appearance variations caused by illumination, pose, cluttered background, interactions, and camera movement. Further, monitoring an intersection means that the number of vehicles constantly changes every few seconds that may lead to a decrease in tracking accuracy. Hence, most of the existing MOT approaches focus on a single class of object, usually people or vehicles.

The various MOT approaches can be categorized as either *batch* or *online* trackers. Batch trackers combine past, present, and future tracking information for associating the detections to the correct tracks. However, batch methods have higher computational cost whereas online methods only consider past and current frame information for data association. Hence, online methods are more suitable for real-time applications like traffic monitoring.

Tracking vehicles in aerial videos have been studied in the past using optical flow based methods like Kanade-Lucas optical flow [13], features like Kanade-Lucas-Tomasi (KLT) [14], and particle filters [15] have been used. These methods apply interest point based tracking, but because of the background complexity in aerial videos, some irrelevant interest points can be extracted from the background. Further, such deterministic trackers cannot handle drifts in the object trackers as compared to stochastic methods like Bayesian filters. This is the reason many online trackers use Bayesian filtering for capturing motion dynamics and observation models to estimate posterior likelihoods of vehicle position.

One of the popular Bayesian filtering methods are Markov chain Monte Carlo (MCMC)-based methods [16] that can handle various object moves and interactions of multiple objects. However, naive MCMC methods assume that the number of objects does not change over time that is not applicable at intersections. Hence, reversible jump MCMC (RJMCMC) based methods were proposed in [17], where a variable number of objects can be handled with the help of update, swap, birth, and death operations for each track. As variations in appearances, interaction, occlusions and changing the number of objects introduce computation overhead, an MCMC sampling with low computation overhead by separating motion dynamics into birth and death moves was proposed in [18]. However, as birth and death are determined in separate MCMC chains, each Markov chain has no dimension variation and can reach to stationary states with less computation overhead. However, such a simple approach for separating birth and death dynamics cannot deal with complex situations like track drifts due to appearance variations. Further, the dynamically varying number of objects cause multiple track drifts in crowded or high traffic scenarios as generally observed in intersections [19], [20], [21].

In [22], a simpler Markov Decision Process (MDP) based online-tracker was proposed that can address track drifts by learning and updating an appearance model for a target to handle appearance changes based on the object detection outputs at every frame. Further, MDP can effectively handle birth/death and appearance/disappearance of targets by associating them with state transitions. In [23], recurrent neural networks (RNN) were proposed to model the birth/death/update of each target track and data association was performed using long short-term memory (LSTM) cells for each target. The training of the RNN and LSTM units is based on synthetic data generated from the actual data. Hence, the RNN is not being able to handle sudden changes in motion and direction and results in poorer accuracy in tracking as compared to MDP based tracking. As traffic movement of motorbikes and other vehicles contain such variations at intersections, RNNs are unsuitable for tracking such movements. Thus, MDP based tracking seems to be the most reasonable choice for tracking traffic movement at intersections.

While MDP based tracking can account for appearance changes, it cannot estimate the influence of other vehicles on the target vehicle. An interaction model captures the influence of an object on other objects. This is especially true for heterogeneous traffic with no specific demarcation of lanes for motorbikes, cars, and trucks. This results in the adjustment of speed and direction to avoid collisions. In such cases, smaller vehicles change lanes based on the "force" experienced from other vehicles, especially larger ones. The car-following model [24] that is generally used to describe homogeneous traffic with lane discipline cannot be used to describe the aforementioned behavior. To accommodate motorcycle-heavy traffic, a tri-class flow (considering bus, car, and motorcycle as separate flows) was empirically studied in [25]. The traffic flow problem was described as two-wheeler accumulation in different lanes alongside buses and cars which were segmented as vehicle packets. However, these vehicle packets were still segregated by lanes. Such a packet formation fails to account for the unique kinetic characteristics of two-wheelers riding between lanes as suggested by the authors in [25]. Hence, interaction models also known as social force models were developed where each object is considered to be dependent on other objects and environmental factors [26]. Understanding these forces and accounting for them allow effective tracking even in crowded traffic scenes generally encountered at intersections. Social force models categorize target behavior based on two aspects, individual force and group force.

Individual force is defined for each target, and is further subdivided into two forces - *fidelity* that means that the target should not change its desired direction, and *constancy* which means that one should not suddenly change its speed and direction. Group force is further categorized into three types of forces - *attraction* between individuals moving together as a group, *repulsion* that refers to the minimum distance maintained between members in a group, and *coherence* that means individuals moving together in a group move with similar velocity.

The majority of existing publications focus on modeling pedestrian dynamics with social force models [27], [28], [29], [30] but there is limited literature on traffic modeling using social force dynamics [31], [32], [33]. However, the traffic models developed with social force carefully consider vehicle dimensions, turning radius, the exact distance between vehicles, etc. In real scenarios, for any arbitrary vehicle at any intersection, information about vehicle dimensions and exact distances are difficult to obtain from aerial videos. Hence, the relative distance between the trajectories of the neighboring vehicles which is independent of the dimensions of the target vehicle is used [27], [28]. However, these approaches focus on predicting the average future trajectory by minimizing the L2 distance from the ground truth future trajectory whereas the goal should be to generate multiple good trajectories for every vehicle given the current position. This leads us to choose a Generative Adversarial Network (GAN) based encoder-decoder architecture to predict the most likely vehicle trajectory. This GAN uses a pooling layer to model vehicle-vehicle interactions and a loss function that allows the network to produce multiple diverse future trajectories for the same observed sequence. These future trajectories are based on both the distance and probability of collision in the future with neighboring vehicles.

III. Vehicle Interaction Modeling using GAN

In order to estimate the influence of various vehicles in the vicinity of the target vehicle, there is a need to jointly reason and predict the future trajectories of all the vehicles involved in an intersection. Assuming that the trajectories for the vehicles in a scene are obtained from a tracking algorithm as $\mathbf{X} = X_1, X_2, \cdots, X_n$, the goal is to predict the future trajectories $\hat{\mathbf{Y}} = \hat{Y}_1, \hat{Y}_2, \cdots, \hat{Y}_n$ of all the vehicles simultaneously. The input trajectory of a vehicle i is defined as $X_i = (x_i^t, y_i^t)$ from time steps $t = 1, ..., t_{obs}$, the ground truth future trajectory is defined as $Y_i = (x_i^t, y_i^t)$ from time steps $t = t_{obs} + 1, \cdots, t_{pred}$, and the predicted trajectory is defined as $\hat{Y}_i$.

A Generative Adversarial Network (GAN) comprises of two neural networks - a generative model G to capture the data distribution, and a discriminative model D to estimate whether a sample arrived from the training data rather than G. The generator G takes a latent variable z as input, and outputs a sample $G(z)$ while the discriminator D takes a sample x as input and outputs $D(x)$ which represents the probability that it is real. The training procedure is akin to a two-player min-max game with the objective function

$$\min_G \max_D V(G, D) = \\ \mathbb{E}_{x \sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z \sim p(z)}[log(1 - D(G(z)))]. \quad (1)$$

Conditional GAN expands the functionality of the traditional GAN architecture by accepting an additional input c at both the generator and discriminator to produce $G(z,c)$ and $D(x,c)$, respectively [8]. Such conditional GANs can be used to replicate models conditioned on a prior distribution (in this case, the prior trajectory of the vehicle during $t = 1, ..., t_{obs}$.

Trajectories of vehicle movements are a form of time-series data with many possible futures based on different intentions like giving more space to larger vehicles and avoiding overtaking maneuvers, left turn, right turn, or U-turn on a multi-lane road, etc. This makes the vehicle trajectory prediction problem truly multimodal and GANs can help predict all the different possibilities. In a nutshell, the GAN used in this work consists of a generator, a pooling stage, and a discriminator. The generator is an encoder-decoder framework where the hidden states of encoder and decoder are linked with the help of a pooling module. The generator takes in input X_i and outputs predicted trajectory $\hat{Y}_i$. The discriminator receives the entire sequence comprising both input trajectory X_i and future prediction $\hat{Y}_i$ (or Y_i) as input and classifies them as either real or fake.

In order to produce the input for the generator, the location of each person is embedded into a fixed length vector $\mathbf{e}_i^t$ using a 1-layer multi-layer perceptron (MLP) as in [27]. These embeddings are then used to initialize the hidden state of the encoder in the LSTM at time t as

$$\begin{aligned} e_i^t &= \phi(x_i^t, y_i^t; \mathbf{W}_{ee}), \\ h_{ei}^t &= LSTM(h_{ei}^{t-1}, e_i^t; \mathbf{W}_{encoder}), \end{aligned} \tag{2}$$

where $\phi(.)$ is an embedding function with ReLU non-linearity, $\mathbf{W}_{ee}$ is the embedding weight, h_{ei}^t is the hidden state of the i^{th} encoder at time t and the LSTM weights, $\mathbf{W}_{encoder}$, are shared between all the vehicles to provide global context of the scene. The encoder learns the state of the vehicle and stores the motion pattern for that particular vehicle. Similar to the social LSTM model [27], a pooling stage (PS) is designed to share the information between the different encoders that models vehicle-vehicle interaction. After observing the motion of each vehicle till t_{obs}, the hidden states of all the vehicles present at the intersection are pooled (max-pooled in our implementation) to obtain a tensor $\mathbf{P}_i$ for each vehicle. As the goal is to produce future trajectories that are synchronized with past driving behavior in the observation period, the hidden state of the decoder is conditioned based on the combined tensor as

$$\begin{aligned} c_i^t &= \gamma(\mathbf{P}_i, h_{ei}^t; \mathbf{W}_c), \\ h_{di}^t &= [c_i^t, z], \end{aligned} \tag{3}$$

where $\gamma()$ is a multi-layer perceptron (MLP) with ReLU non-linearity, h_{di}^t is the hidden state of the i^{th} decoder at time t, and $\mathbf{W}_c$ is the embedding weight.

After initializing the decoder states as above, the predictions can be obtained as follows:

$$\begin{aligned} e_i^t &= \phi(x_i^{t-1}, y_i^{t-1}; \mathbf{W}_{ed}), \\ \mathbf{P}_i &= PS(h_{d1}^{t-1}, ..., h_{dn}^t), \\ h_{di}^t &= LSTM(\gamma(\mathbf{P}_i, h_{di}^{t-1}), e_i^t; \mathbf{W}_{decoder}), \\ (\hat{x}_i^t, \hat{y}_i^t) &= \gamma(h_{di}^t), \end{aligned} \tag{4}$$

where $\phi(.)$ is an embedding function with ReLU non-linearity with $\mathbf{W}_{ed}$ as the embedding weights. The LSTM weights are given by $\mathbf{W}_{decoder}$ and $\gamma(.)$ denotes an MLP.

The discriminator uses a separate encoder which takes as input $T_{real} = [X_i, Y_i]$ or $T_{fake} = [X_i, \hat{Y}_i]$ and classifies them as real or fake. The discriminator learns interaction behavior and classifies unacceptable trajectories as “fake”. While a GAN is trained using adversarial loss given in 2, L2 loss is used to estimate the distance of the generated path from the actual ground truth.

To estimate the trajectory of multiple vehicles, we need to share information across the LSTMs representing each vehicle. However, the number of vehicles at an intersection is high, and the number varies depending on the traffic condition. Therefore, there is a need for a compact representation to store shared information. Further, local interactions are not always sufficient to determine future trajectories, and far-away vehicles might impact the path taken by a vehicle. Hence, the network needs to model the global context. In social pooling [27], [28] based approaches, a grid-based pooling scheme is proposed that considers only local context and fail to capture global context. As per [34], both a compact representation and global context can be learned using a symmetric function on transformed elements of the input set of points. Hence, in this paper, the input coordinates are passed through an MLP followed by a symmetric function like Max-Pooling. The pooled vector $\mathbf{P}_i$ summarizes all the information needed for a vehicle to choose a path. Also, the relative position of each person in relation to person i is augmented with input to the pooling module.

Though GAN produces good predictions, these predictions are the “average” prediction in case of multiple outputs. In order to encourage the generation of diverse samples, we use a variety loss given in [35]. For each scene, k possible output predictions are generated by randomly sampling z from $\mathcal{N}(0, 1)$ and the best prediction in terms of L2 distance from the ground truth trajectory is kept as the prediction.

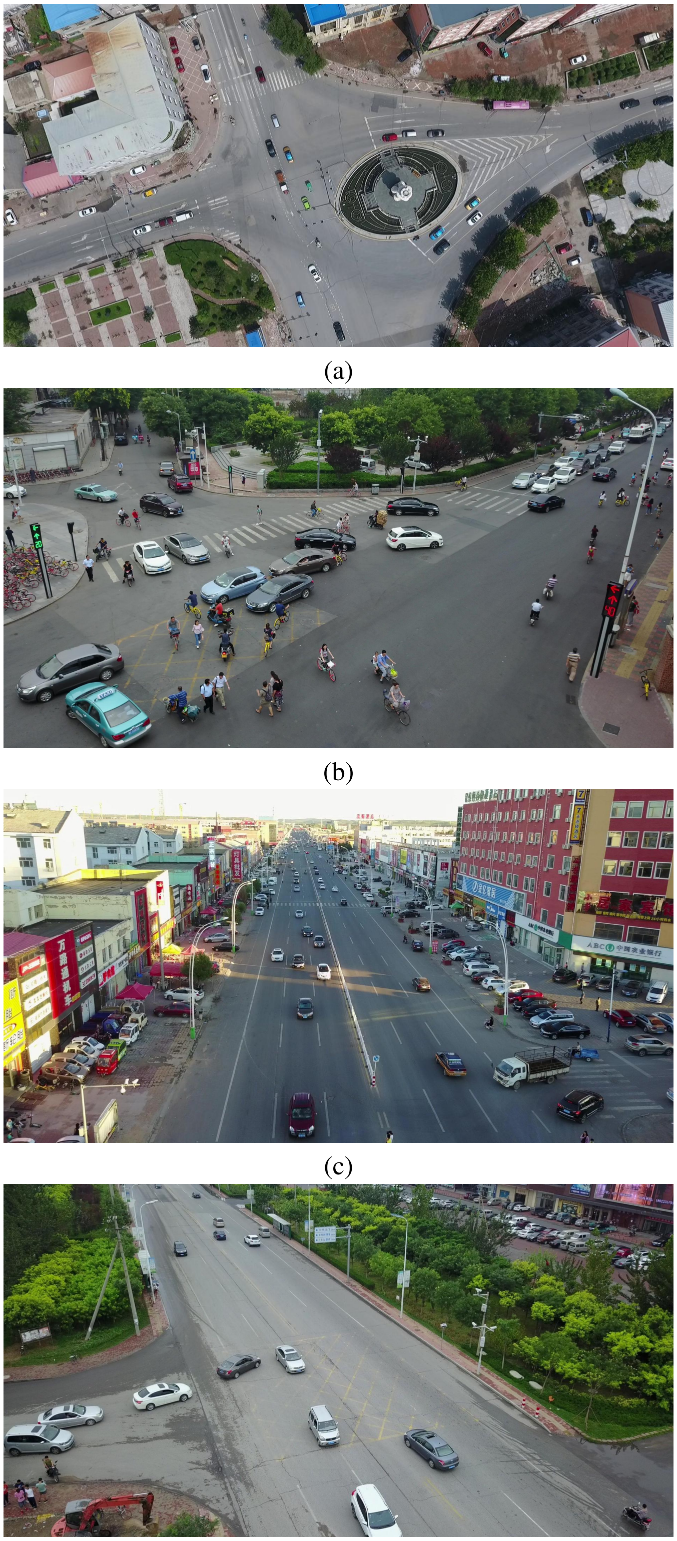
$$\mathcal{L}_{variety} = \min_k \|Y_i - \hat{Y}_i^{(k)}\|_2, \tag{5}$$

where k is a hyper-parameter. By considering only the best trajectory, this loss encourages the network to explore the space of outputs that are closest to the past trajectory.

IV. Experimental Results

A. Dataset description

The VisDrone dataset contains 96 video clips that include 56 clips for training, 7 for validation, and 33 for testing. Among them, we chose only the videos that depict vehicular traffic at intersections. Finally, the dataset considered for evaluation in this paper consists of 23 clips for training (10,239 frames with approximately 11,000 vehicles), 5 for validation (2,033 frames with approximately 2,400 vehicles), and 6 for testing (2,110 frames with approximately 2,500 vehicles) from the VisDrone dataset. The detailed statistics of the different types of vehicles in these videos are reported in Table I. Some signalized and non-signalized intersections used during testing are shown in Figure 2.



(a)

(b)

(c)

(d)

Fig. 2: Examples of signalized intersections (a) and (b) and non-signalized intersections (c) and (d) in VisDrone dataset.

TABLE I: Statistics of intersection videos in VisDrone dataset

Vehicle type	**Number of vehicles**
Bicycle	2,245
Bus	782
Car	7,345
Motorcycle	4,823
Van	372
Truck	56
Total	15,623

B. Vehicle tracking

It is important to determine the best tracking algorithm to supply vehicle trajectories to the prediction framework. Hence, we compare the tracking performance of three popular online tracking algorithms that can efficiently track hundreds of vehicles at every intersection - 1) Markov Decision Process (MDP) based tracker [22], 2) simple, online, real-time tracking (SORT) [36], and 3) deep SORT [37] that integrates the SORT algorithm with appearance features from the YOLO [38] detection framework. The different metrics used for comparison in Table II that are specific to multi-object tracking are:

- Multiple Object Tracking Accuracy (MOTA) combines three sources of errors - false negatives (FN), false positives (FP) and ID switches (IDs) as follows
$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDs_t)}{\sum_t GT_t}$$
where t is the frame index, and GT is the number of ground truth objects.
- Multiple Object Tracking Precision (MOTP) is the average difference between all true positives and their corresponding ground truth targets.
$$MOTP = \frac{\sum_{t,i} d_{t,i}}{\sum_t c_t}$$
where c_t denotes the number of matches in frame t and $d_{t,i}$ is the bounding box overlap of target i with its assigned ground truth object.
- FM shows the number of times a ground truth trajectory is interrupted during tracking.
- Each ground truth trajectory can be classified as mostly tracked (MT) if it is tracked more than 80% of its lifespan, partially tracked (PT) if it is tracked more than 50%, and mostly lost (ML) if it tracked less than 20%.

While we were able to test SORT and MDP with different detection frameworks, deepSORT integrates the appearance features provided by different layers of the YOLO [39] architecture to associate the targets. This prevents the tracking framework to be decoupled from the detection network. It can be observed from Table II that the MDP tracker has the highest recall and MOTA among the other trackers. It can be observed although R-FCN consistently provides better precision, F-RCNN has the best recall which is justified by the more number of proposals evaluated by F-RCNN. AS SSD has a similar architecture to YOLO, both of them fail at detecting smaller objects that translate to poor tracking performance for the trackers utilizing their detection outputs. Overall, the tracking strategy of MDP yields better accuracy in tracking with much less false negatives and a superior recall score as well. This can be attributed to the state-based handling of the appearance and disappearance of vehicles that prove to be a better strategy for associating the tracks with the detection outputs.

C. Trajectory Prediction

The obtained tracks are then used to train the vehicle trajectory prediction algorithms. We compare the GAN based prediction system to existing approaches using social LSTM (S-LSTM) [27] and social attention-based structural recurrent neural network (S-RNN) [30]. For both the encoder and decoder in the proposed GAN, LSTM is used as the RNN, and the dimensions of the hidden state for the encoder is 16 and the decoder is 32. The input coordinates obtained from the tracker output are transformed into a relative coordinate system with the center of the video taken as the origin in order to achieve translation invariance. These relative coordinates are then embedded as 16-dimensional vectors to be provided as input to the encoder. The Generator and Discriminator are iteratively trained with a batch size of 64 for 200 epochs using Adam optimizer with an initial learning rate of 0.001. For the S-LSTM and S-RNN, we use the implementation provided by the respective authors. For the GAN model, we modify and use the implementation given by the authors in [35]. All the prediction networks follow different protocols for observation and prediction lengths. For simplicity of comparison, we follow the same observation length of 8 time steps ($t_{obs} = 8$) and prediction length of 8 time steps ($t_{pred} = 8$) for all the networks.

The comparison is done using the following evaluation metrics:

- Average Displacement Error (ADE) which measures the average L2 distance between ground truth and the prediction over all the predicted time steps.
- Final Displacement Error (FDE) that measures the distance between the predicted final destination actual destination at the end of the prediction period t_{pred}.

In Table III, we present the comparison of vehicle trajectory prediction performance of GAN with S-RNN and S-LSTM. It can be observed that GAN produces better ADE and FDE scores than the other two prediction approaches. This can be attributed to the generator in GAN being trained with a variety loss that is able to predict more diverse set of trajectories than both S-LSTM and S-RNN. Further, the global context employed in GAN is more apt for vehicle trajectory prediction at intersections as vehicles can rapidly accelerate or decelerate at intersections. Even vehicles which are separated initially can become close rapidly and a global strategy helps in keeping track of such movements for better predictions.

D. Qualitative Analysis

Traffic prediction using GAN having a social structure helps us predict two basic movement types used by smaller vehicles like motorcycles and scooters in traffic - merging and overtaking. While merging, vehicles avoid collisions while continuing towards their destination by either slowing down or altering their course slightly or a combination of both. This behavior is highly dependent on the context and behavior of other surrounding vehicles. The proposed model (GAN + MDP + SSD) can predict the variation in both the speed and direction of a vehicle to effectively navigate nearby traffic. For instance, the model predicts that either vehicle Y (yellow) slows down or both vehicle R (red) and Y change direction to avoid collision (Figure 3 (a) and (b)).

Another common scenario encountered in traffic is where a vehicle might want to either maintain pace or maybe overtake the vehicle in front. This has been studied with car-following models in literature [24]. The decision making ability while overtaking is restricted by the field of view. However, as the GAN model has access to the ground truth positions of all the vehicles involved in the scene, it results in some interesting predictions. For example, in Figures 3 (c) and (d), the model predicts that vehicle R (in red) is obstructed by vehicle B (blue) and will give way by changing their direction. This global knowledge allows GAN to correctly predict that vehicle Y (yellow) will overtake vehicle B.

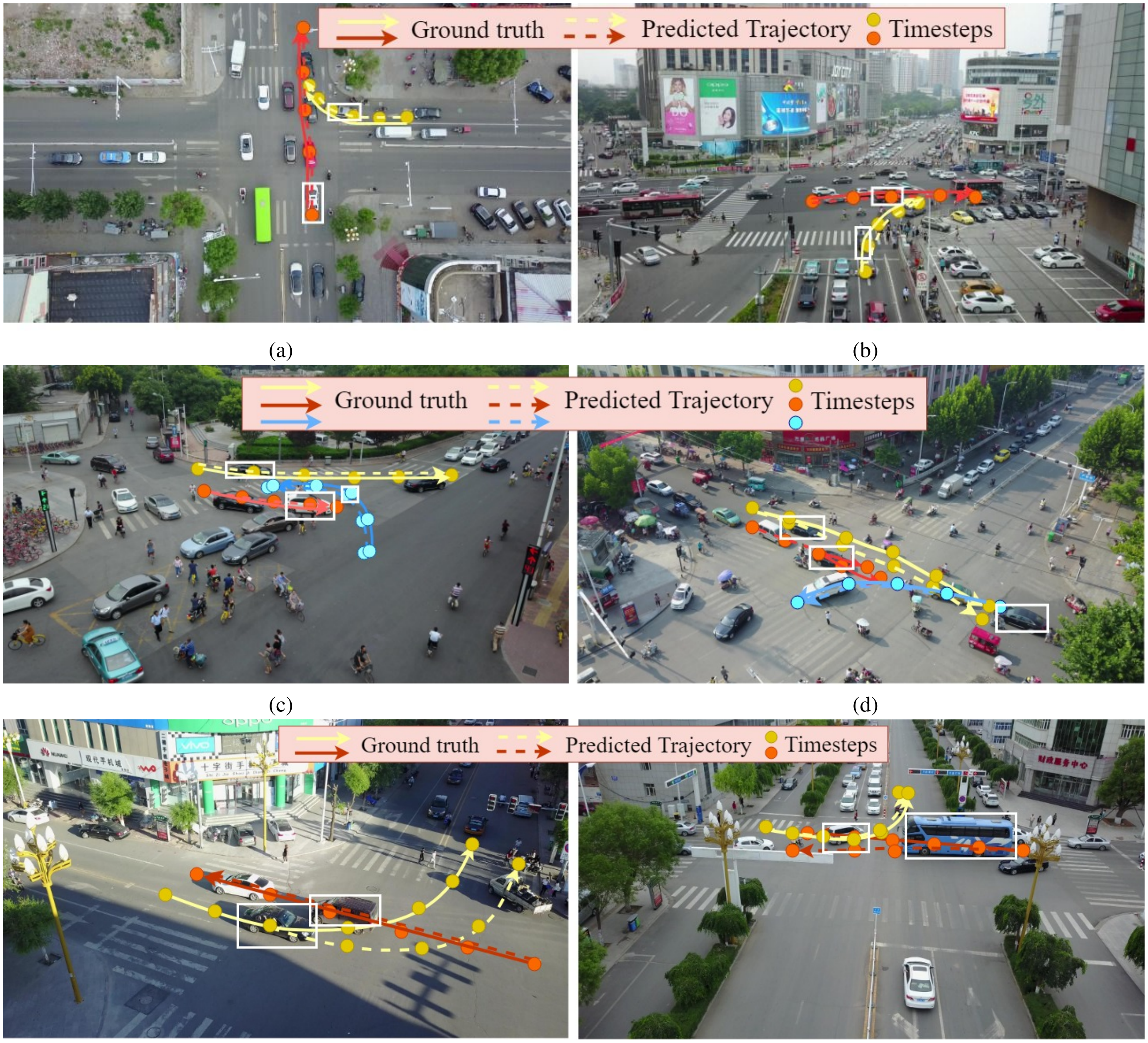
Vehicles also avoid each other when moving in opposite directions without any physical barrier separating both the streams of traffic. This tendency manifests in smaller vehicles generally bunching with other vehicles moving in the same direction. Also, smaller vehicles (vehicle Y) mostly observe the movement of larger vehicles (vehicle R) in the opposite direction and overtake only if they predict that there is adequate clearance distance (Figure 3 (f)). However, in the presence of smaller vehicles, the driver in vehicle Y (yellow) makes a choice very late and close to the oncoming vehicle R (red) (Figure 3 (e)). The model is not able to distinguish between these two behaviors as the type of vehicle is not taken into consideration during prediction and the prediction is not aligned with the ground truth (Figure 3 (e)). In such case, we hypothesize that decisions based on local vicinity can produce better predictions rather than accounting for vehicles that are further away (global context).

V. CONCLUSION

In this paper, we proposed a vehicle trajectory prediction approach that considers the interaction among vehicles. These interactions provide a global context for training a Generative Adversarial Network (GAN) in order to predict trajectories of each vehicle at both signalized and non-signalized intersections. The prediction algorithm can predict vehicle trajectory resulting from different traffic maneuvers like overtaking, merging, and avoiding oncoming traffic without any additional information about the dimension of the road. The proposed GAN based network produces multiple future trajectories and chooses the best based on the past driving behavior of a vehicle encapsulated in the vehicle trajectory. An evaluation on the intersection videos of the VisDrone dataset demonstrates the efficacy of GAN is predicting trajectories with minimal deviation compared to the actual trajectories followed by different types of vehicles. Further, as there are no assumptions

TABLE II: Comparison of different trackers on intersection videos for all vehicle types in the VisDrone dataset. The detection frameworks used are also mentioned.

Method		**Recall (%)**	**Precision (%)**	**GT**	**MT**	**PT**	**ML**	**FP**	**FN**	**IDs**	**FM**	**MOTA (%)**	**MOTP (%)**
DeepSort [37]	YOLO	7.8	79.3	526	42	64	420	1522	69039	342	623	5.2	73.5
SORT [36]	F-RCNN	17.7	93.5	526	51	79	396	902	60306	305	520	16.0	**79.4**
	SSD	5.5	82.6	526	2	44	480	856	69172	256	437	4.0	75.7
	R-FCN	16.6	**91.5**	526	49	77	400	1132	61045	340	539	14.6	78.8
MDP [22]	F-RCNN	**25.5**	89.4	526	69	118	339	2216	54564	192	497	**22.2**	78.0
	SSD	11.1	78.4	526	20	68	438	2244	65088	157	403	7.8	74.4
	R-FCN	24.6	87.6	526	70	111	345	2547	55213	248	552	20.8	77.0



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 3: Prediction of traffic flow using GAN in signalized and unsignalized intersections during various maneuvers - (a) and (b) merging, (c) and (d) overtaking, and (e) and (f) preventing oncoming traffic. While the traffic flow during merging and overtaking is predicted correctly, overtaking is more challenging as the size of the vehicle dictates the distance maintained by the drivers as seen in (e) and (f). The vehicles in consideration are enclosed in white boxes. Best viewed in color.

TABLE III: Comparison of prediction performance with state-of-the-art on the intersection videos of VisDrone dataset.

Method			**ADE**	**FDE**
S-RNN [30]	DeepSort		0.88	1.60
	SORT	F-RCNN	0.92	1.76
		R-FCN	0.92	1.75
		SSD	0.85	1.52
	MDP	F-RCNN	0.89	1.56
		R-FCN	0.86	1.61
		SSD	0.94	1.66
S-LSTM [27]	DeepSort		0.89	1.57
	SORT	F-RCNN	0.77	1.62
		R-FCN	0.91	1.66
		SSD	0.92	1.67
	MDP	F-RCNN	0.82	1.53
		R-FCN	0.89	1.51
		SSD	0.79	1.42
GAN	DeepSort		0.91	1.65
	SORT	F-RCNN	0.87	1.66
		R-FCN	0.87	1.65
		SSD	0.77	1.42
	MDP	F-RCNN	0.82	1.53
		R-FCN	0.84	1.58
		SSD	**0.72**	**1.32**

about the type of traffic and their movements, the method can be applied for the analysis of any type of intersection. In the future, we would like to consider the vehicle type to improve predictions in some traffic scenarios.

REFERENCES

[1] S.-P. Lin, Y.-H. Chen, and B.-F. Wu, "A real-time multiple-vehicle detection and tracking system with prior occlusion detection and resolution, and prior queue detection and resolution," in *18th International Conference on Pattern Recognition (ICPR'06)*, vol. 1, Aug 2006, pp. 828–831.

[2] T. Gandhi and M. M. Trivedi, "Vehicle surround capture: Survey of techniques and a novel omni-video-based approach for dynamic panoramic surround maps," *IEEE Transactions on Intelligent Transportation Systems*, vol. 7, no. 3, pp. 293–308, 2006.

[3] S. Lyu, M.-C. Chang, D. Du, L. Wen, H. Qi, Y. Li, Y. Wei, L. Ke, T. Hu, M. Del Coco *et al.*, "Ua-detrac 2017: Report of avss2017 & iwt4s challenge on advanced traffic monitoring," in *Advanced Video and Signal Based Surveillance (AVSS), 2017 14th IEEE International Conference on*. IEEE, 2017, pp. 1–7.

[4] P. Preethi, A. Varghese, and R. Ashalatha, "Modelling delay at signalized intersections under heterogeneous traffic conditions," *Transportation Research Procedia*, vol. 17, pp. 529–538, 2016.

[5] X. Liang, L. Zhili, and Q. Kun, "Capacity analysis of signalized intersections under mixed traffic conditions," *Journal of Transportation Systems Engineering and Information Technology*, vol. 11, no. 2, pp. 91–99, 2011.

[6] J. Apeltauer, A. Babinec, D. Herman, and T. Apeltauer, "Automatic vehicle trajectory extraction for traffic analysis from aerial video data," *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 40, no. 3, p. 9, 2015.

[7] N. Y. Cao and K. Sano, "Estimating capacity and motorcycle equivalent units on urban roads in hanoi, vietnam," *Journal of Transportation Engineering*, vol. 138, no. 6, pp. 776–785, 2012.

[8] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[9] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using sift features and mean shift," *Computer Vision and Image Understanding*, vol. 113, no. 3, pp. 345 – 352, 2009, special Issue on Video Analysis. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314208001331

[10] B. Bose, X. Wang, and E. Grimson, "Multi-class object tracking algorithm that handles fragmentation and grouping," in *2007 IEEE Conference on Computer Vision and Pattern Recognition*, June 2007, pp. 1–8.

[11] F. Meng, Z. Qu, Q. Zeng, and L. Li, "Traffic object tracking based on increased-step motion history image," in *2007 IEEE International Conference on Automation and Logistics*, Aug 2007, pp. 345–349.

[12] C. Qian and Z. Xu, "Robust visual tracking via sparse representation under subclass discriminant constraint," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 7, pp. 1293–1307, July 2016.

[13] R. Ke, S. Kim, Z. Li, and Y. Wang, "Motion-vector clustering for traffic speed detection from uav video," in *Smart Cities Conference (ISC2), 2015 IEEE First International*. IEEE, 2015, pp. 1–5.

[14] A. C. Shastry and R. A. Schowengerdt, "Airborne video registration and traffic-flow parameter estimation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 4, pp. 391–405, Dec 2005.

[15] X. Cao, C. Gao, J. Lan, Y. Yuan, and P. Yan, "Ego motion guided particle filter for vehicle tracking in airborne videos," *Neurocomputing*, vol. 124, pp. 168–177, 2014.

[16] E. Richter, M. Obst, M. Noll, and G. Wanielik, "Tracking multiple extended objectsa markov chain monte carlo approach," in *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*. IEEE, 2011, pp. 1–8.

[17] M. Bocquel, H. Driessen, and A. Bagchi, "Multitarget tracking with ip reversible jump mcmc-pf," in *Proceedings of the 16th International Conference on Information Fusion*, July 2013, pp. 556–563.

[18] H. Sakaino, "Video-based tracking, learning, and recognition method for multiple moving objects," *IEEE transactions on circuits and systems for video technology*, vol. 23, no. 10, pp. 1661–1674, 2013.

[19] K. Peng, G. Cai, B. M. Chen, M. Dong, K. Y. Lum, and T. H. Lee, "Design and implementation of an autonomous flight control law for a uav helicopter," *Automatica*, vol. 45, no. 10, pp. 2333–2338, 2009.

[20] R. Du, Z. Peng, and Q. Lu, "Comparison of sms calculation methods based on ngsim data for uav detection," in *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, vol. 2, Aug 2012, pp. 112–115.

[21] K. Kanistras, G. Martins, M. J. Rutherford, and K. P. Valavanis, "A survey of unmanned aerial vehicles (uavs) for traffic monitoring," in *Unmanned Aircraft Systems (ICUAS), 2013 International Conference on*. IEEE, 2013, pp. 221–234.

[22] 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.

[23] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *AAAI*, February 2017.

[24] G. F. Newell, "A simplified car-following theory: a lower order model," *Transportation Research Part B: Methodological*, vol. 36, no. 3, pp. 195–205, 2002.

[25] C. Lan and G. Chang, "Empirical observations and formulations of tri-class traffic flow properties for design of traffic signals," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2018.

[26] D. Helbing and P. Molnar, "Social force model for pedestrian dynamics," *Physical review E*, vol. 51, no. 5, p. 4282, 1995.

[27] A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social lstm: Human trajectory prediction in crowded spaces," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 961–971.

[28] A. Robicquet, A. Sadeghian, A. Alahi, and S. Savarese, "Learning social etiquette: Human trajectory understanding in crowded scenes," in *European conference on computer vision*. Springer, 2016, pp. 549–565.

[29] A. Sadeghian, A. Alahi, and S. Savarese, "Tracking the untrackable: Learning to track multiple cues with long-term dependencies," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 300–311.

[30] A. Vemula, K. Muelling, and J. Oh, "Social attention: Modeling attention in human crowds," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 1–7.

[31] D. N. Huynh, M. Boltze, and A. T. Vu, "Modelling mixed traffic flow at signalized intersection using social force model," *Journal of the Eastern Asia Society for Transportation Studies*, vol. 10, pp. 1734–1749, 2013.

[32] W. Huang, M. Fellendorf, and R. Schönauer, "Social force based vehicle model for 2-dimensional spaces," in *91st Annual Meeting of the Transportation Research Board. Washington, DC, USA*, 2011.

[33] N. Deo and M. M. Trivedi, "Convolutional social pooling for vehicle trajectory prediction," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[34] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 77–85.

[35] A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social gan: Socially acceptable trajectories with generative adversarial networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 2255–2264.

[36] 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)*, 2016, pp. 3464–3468.

[37] 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.

[38] 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.

[39] J. Redmon and A. Farhadi, "Yolo9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2017, pp. 6517–6525.