Segmentation of Moving Objects in Traffic Video Datasets

Anusha Aswath[1], Renu Rameshan[1], Biju Krishnan[2] and Senthil Ponkumar[2]
[1]*Indian Institute of Technology, Mandi, Himachal Pradesh, India*
[2]*Continental Tech Centre, Bengaluru, Karnataka, India*

Keywords: Multi-object Tracking, CNN Model, Re-identification, Instance Segmentation, Ground Truth, Interactive Correction, Annotation Tool.

Abstract: In this paper, we aim to automate segmentation of multiple moving objects in video datasets specific to traffic use case. This automation is achieved in two steps. First, we generate bounding boxes using our proposed multi-object tracking algorithm based on convolutional neural network (CNN) model which is capable of re-identification. Second, we convert the various tracked objects into pixel masks using an instance segmentation algorithm. The proposed method of tracking has shown promising results with high precision and success rate in traffic video datasets specifically when there is severe object occlusion and frequent camera motion present in the video. Generating instance aware pixel masks for multiple object instances of a video dataset for ground truth is a tedious task. The proposed method offers interactive corrections with human-in-the-loop to improve the bounding boxes and the pixel masks as the video sequence proceeds. It exhibits powerful generalization capabilities and hence the proposed tracker and segmentation network was applied as a part of an annotation tool to reduce human effort and time.

1 INTRODUCTION

Advanced Driver Assistance Systems (ADAS) are systems with camera sensors for vision, radar sensors for detection and lidar sensors for distance measurements. The camera images and information obtained from fusion of different sensors are utilized by computer vision algorithms to train models to perform functions like vehicle and pedestrian detection, traffic sign recognition, lane change assist and assisted speed control to name a few. Algorithms developed for computer vision tasks require machine learning models and deep neural networks to be trained. To train these networks, huge amounts of labeled data are required to obtain generalised results. The data available for ADAS mostly consists of long video recordings (some millions of kilometers).

Data labeling task requires human-in-the-loop to annotate images using an annotation tool. Some of the latest annotation tools provide markers for line, box, polyline, polygon and pixel annotation. Mechanisms like superpixel (Achanta et al., 2012) or watershed (Beucher and Meyer, 1993) have also been introduced for assisted pixel labeling. Artificial Intelligence (AI) assisted web-based platforms are also being developed for automating labeling of data. Currently, the effort for manual data labeling for one hour of recording (which consists of 1500000 frames) is 149 hours for cars and 79 hours for pedestrians for fine pixel masks (outlining the object). It is also of concern that a company has a few hundreds of in-house workers to do annotation. Hence a lot of capital is invested by companies and time of individuals lost for doing annotation.

Deep learning methods have helped in achieving state-of-the-art results in various computer vision tasks. We shall utilize deep learning algorithms to segment multiple moving objects and identify the objects with same labels throughout the video. We shall apply this solution as a part of an annotation tool to automate labeling process. As the segmentation of multiple objects requires a bounding box around the object to be segmented, we start with the process of tracking for multiple objects in a video frame. This is followed by segmentation of objects inside the bounding boxes obtained. We also aim at interactive corrections for labeling apart from only reviewing and adjusting the generated labels on the annotation tool. In case of any correction required in the bounding boxes generated, the annotator should be able to shift the box and consequently improve the tracking accuracy in further frames which is enabled in our solution.

Though there exist various deep learning algorithms which achieve multiple object tracking (Chu

Aswath, A., Rameshan, R., Krishnan, B. and Ponkumar, S.
Segmentation of Moving Objects in Traffic Video Datasets.
DOI: 10.5220/0008940403210332
In *Proceedings of the 9th International Conference on Pattern Recognition Applications and Methods (ICPRAM 2020)*, pages 321-332
ISBN: 978-989-758-397-1; ISSN: 2184-4313

et al., 2017; Gordon et al., 2018), we propose to use a confidence score based CNN tracker that initalizes on a target and updates online for tracking (which also facilitates interactive corrections). For tracking multiple objects, a single object tracker (Yun et al., 2017) is modified to a multi-object tracker using multi-domain learning technique (Nam and Han, 2016). The position of the target in the next frame is predicted using the appearance and motion information to predict linear transformations in the form of certain actions. To automate the process of incorporating new objects in the tracking framework we make use of a data association module. It uses similarity learning to associate detections with their corresponding tracks and to handle re-identification.

For instance segmentation, using the multiple bounding boxes for all the tracked objects in a video frame we outline the objects using polygon vertices predicted by Polygon RNN++ (Acuna et al., 2018). This mimics the common technique used for generating pixel masks through polygon or polyline markers. We combine these two solutions for segmenting multiple objects in traffic videos and thus provide a solution to reduce the effort of manual annotation for multiple object instances.

This is an application based paper which aims to assist annotation of multiple objects in traffic video datasets. The main contributions of this paper are summarized as follows-

- A novel multiple object tracking network for annotation
  - CNN based multi-object tracker with online update based on a dynamic tracking score for each object.
  - Maintain track with consistent identities during occlusions or complex interactions.
  - Perform re-identification of targets for traffic datasets on a search area determined by the motion model.
- Integrating the tracker and segmentation network as part of a custom label tool.

The rest of the paper is organised as follows: Section 2 gives the related work, the proposed solution is discussed in Section 3, followed by results and conclusion in Sections 4 and 5, respectively.

2 RELATED WORK

Multi-object tracking (MOT) is the problem of simultatneously solving for the trajectories of individual objects, while maintaining their identities over time through occlusions, clutter and complex interactions. There are two broad categories for solving the MOT problem - 1) global data association and 2) visual tracking. Global data association method formulates the tracking problem as forming trajectories by recursively connecting the detections. It uses optimization methods to minimize cost functions formulated through network flow (Pirsiavash et al., 2011) or using continuous energy minimization (Milan et al., 2013). It also includes linear programming (Jiang et al., 2007) and MAP (Maximum a posteriori) estimation (Pirsiavash et al., 2011) to track multiple objects simultaneously. Tracking-by-detection is a tracking paradigm where tracked objects are linked to detections, treated as a data association problem (Andriluka et al., 2008). All the above methods heavily rely on the detection performance.

With the increasing work on appearance based models for visual tracking, trackers can be broadly classified into two groups - discriminative and generative trackers. Discriminative methods define the tracking problems as a binary classification task, which attempts at designing a classifier to separate targets from their surrounding background. It is important to update the target appearance model to take into account appearance changes, cluttered background, blur or deformations. Various online update techniques include online mixture model (Jepson et al., 2003), incremental subspace update (Ross et al., 2008) and online boosting (Grabner et al., 2006). For discriminative models, the main issue has been improving the sample collection part to make the online-trained classifier more robust (Grabner et al., 2008; Babenko et al., 2009; Kalal et al., 2010; Hare et al., 2015).

A discriminative single object tracker can also be used for tracking multiple objects (Chu et al., 2017). This also demonstrates the problem of online update in MOT scenarios which include complex interactions among targets. For tracking multiple objects, we propose to use a single object tracker which uses a sophisticated appearance model through online update along with incorporating a motion model for each target (Yun et al., 2017). The proposed tracker carefully updates the model in MOT scenarios by maintaining the discriminative appearance model through time.

To handle this issue of maintaining the temporal information of the object to avoid drifts, we resolve to generative method of tracking. These methods search for the most similar regions of the object appearance at each frame, based on learning only the appearance model for object representation. In correlation filter based trackers features are learnt by minimizing the distance between embeddings (measuring similarity)

learnt from the network. Pre-trained CNN models are used to obtain feature maps to correlate two images (Tao et al., 2016). A fully convolutional network that produces a score map from the correlation of a target and search patch was proposed (Bertinetto et al., 2016) with element-wise logistic loss function on the score map. New loss functions such as triplet loss were applied to Siamese networks to learn embeddings (Zhuang et al., 2016). We shall use the offline model for similarity measurement (Bertinetto et al., 2016) for maintaining temporal information while updating our network for tracking.

Recurrent Neural Network (RNN) is another architecture which can be used to model the object motion information along with modeling appearance information. GOTURN (Held et al., 2016) uses a CNN model to regress the location of the object in next image from the previous image. This was improved upon in Real-time Recurrent Regression network (Re3 tracker) (Gordon et al., 2018) using an LSTM (Long Short Term Memory) to model the temporal dependencies. The input to the LSTM is in the form of current and previous frame which helps it learn motion information between pixels. Re3 adapts itself to appearance changes in a single forward pass, through its LSTM cell states and requires resetting at every 32 frames to avoid model drift. This resetting is done as the LSTM states are trained only to remember a maximum of past 32 cell states of the tracked object. It is reset with the first forward pass of the tracked object to retain the previous information instead of setting it all to zero. However it easily drifts in case of significant occlusions and does not track the same object on disappearance and reappearance of the object or during object interactions.

Semi-automatic semantic segmentation is used to obtain labels with human-in-the-loop to obtain guiding signals like bounding boxes, points, edges, scribbles etc. Interactive mechanisms like DeepMask (Pinheiro et al., 2015) provide instance segmentation through its pixel wise prediction map inside a bounding box. Deep Extreme Cut (Xu et al., 2017) offers a guided and interactive annotation method using extreme points. The grab-cut based method extended to Deep Grab Cut (Maninis et al., 2018) produces pixel-wise classification inside bounding boxes. All these techniques classify each pixel inside the bounding box as an object class. Such methods are unsuitable as the labelers need to unmark each pixel carefully when labeled incorrectly, which makes it time consuming. Hence, we propose to use polygon vertices for semantic segmentation which is generated through deep learning based network (Acuna et al., 2018).

3 PROPOSED SOLUTION

In this work, we provide an instance aware segmentation solution for multiple objects in traffic video data sets. Our solution is two-fold, firstly the tracking of multiple objects through traffic videos and secondly the segmentation of these object instances. The proposed multi-object tracker is discussed along with addressing the problems of maintaining consistent tracks in MOT scenarios. This is followed by using the boxes to generate instance aware masks using a segmentation algorithm.

3.1 Tracking Multiple Objects using a Single Object Tracker

We have modified Action Decision Network (ADNet) (Yun et al., 2017), which is a single object tracker for tracking multiple objects.

3.1.1 Overview of Single Object Tracker

ADNet tracks objects through a sequential Markov Decision Process (MDP) which consists of a set of states and actions. The actions (a_t) are taken on the basis of the probabilities predicted by the trained network to provide transitions between states. The state information consists of appearance and motion information. The appearance information (p_t) is the image (F_l) cropped by the bounding box and resized to $112 \times 112 \times 3$. The motion information (d_t) is given by a constant vector of length 110 which includes the past ten actions encoded in the form of one-hot eleven length vector. In case the action taken is left then the bounding box is moved as $[x - \delta x, y, w, h]$, where δ is some small value. The appearance information for the next step in the MDP (p_{t+1}) is given by the image crop obtained from the moved bounding box. The motion information (d_{t+1}) is given by adding the left action one-hot vector and removing the past action vector using the last in first out rule.

This process of tracking through sequential actions is continued for the next time steps in the MDP if the class confidence score for a given target is greater than 0.5. The tracking is continued for a maximum of twenty sequential actions or till stop action is reached on the image (F_l). Once this iteration for taking sequential actions gets completed the bounding box from the previous frame position (F_{l-1}) is said to have reached the target in F_l. This is taken to be the first box in the next frame (F_{l+1}) to start with the MDP to reach the target in F_{l+1}. The network architecture and tracking mechanism using sequential actions and class confidence scores is illustrated in Figure 1.

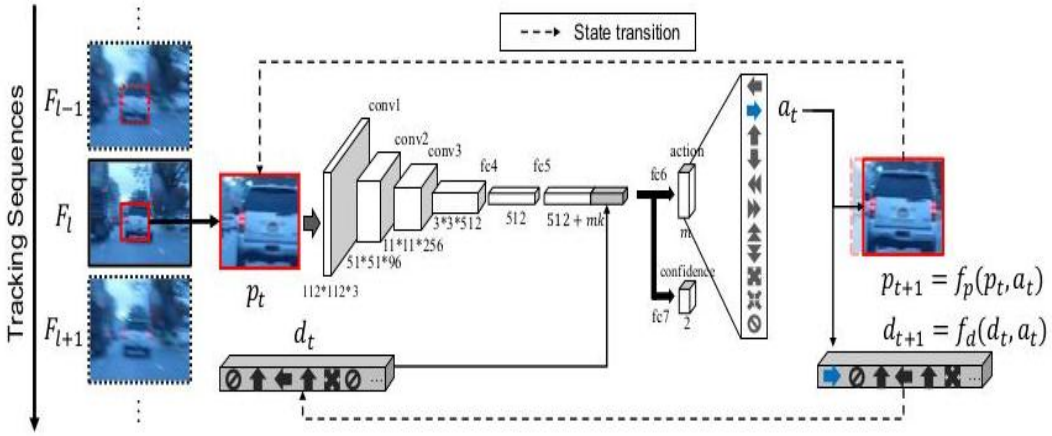


Figure 1: Action Decision Network for tracking single object. The network architecture and tracking sequence illustration is taken from (Yun et al., 2017).

If the class confidence score is less than 0.5 for the given state information then the tracking process is said to have failed and is stopped. We then perform redetection by sampling which involves sampling a set of candidate patches around the last bounding box obtained in the MDP. The candidate patch that has the highest class confidence score is selected as the target for continuing the track as given in Equation 1.

$$b = \arg\max_{b_i} \quad [\text{class confidence scores}(b_i)] \tag{1}$$

where i denotes the index of the sampled patch and the box with the highest class confidence score is selected.

During tracking, online adaptation for structured data (such as video) is performed for the targets using *p-n* (positive-negative) learning (Kalal et al., 2010). p (positive) and n (negative) samples are collected from the successfully redetected position whenever the class confidence score is greater then 0.5. Supervised training is done for the final layers using patches (p_i) sampled randomly around the tracked patch (tp). The corresponding action labels (a_i) and class confidence labels (c_i) are obtained for the patches (p_i) (Yun et al., 2017) through equation 2.

$$\begin{aligned} a_i &= \arg\max_a \text{IoU}(p_i^a, tp) \\ c_i &= \begin{cases} 1, & \text{if IoU}(p_i, tp) > 0.7 \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{2}$$

where IoU refers to intersection over union and a_i refers to that action a on patch p_i that gave the maximum IoU with the redetected patch. All sample patches that have an IoU greater than 0.7 with the redetected patch are positive class and the rest are negative class. Cross entropy loss is used to train the $fc6$ and $fc7$ layers for actions and class confidence scores. The samples from recent successfully tracked frames are collected for short-term update. For training the action layer only positive samples are used whereas for the class confidence layer a discriminative classifier is trained using both positive and negative samples.

3.1.2 Proposed Multi-object Tracker

To track multiple objects using actions and confidence scores, we propose different final layers for the last fully connected layer and prediction layers. The network (Yun et al., 2017) learns a generic representation from many videos during training to produce certain actions given certain states, using shared and domain specific layers for each new video using the multi-domain learning technique (Nam and Han, 2016; Dredze and Crammer, 2008). We utilize this trained network to do multiple object tracking by treating each target to be a new video domain. To learn multiple domains simultaneously, we utilize the shared layers and initialize domain specific ones to learn each of the targets. The $fc5$ layer is initialized with pretrained weights whereas the action vectors vary for each of the target. The $fc6$ and $fc7$ layers are initialized with new weights.

Before starting a track, the final layers initialized with new weights are adapted with p and n samples obtained from the first bounding box on the target. Once K different final layers are adapted for K different objects, the binary classifications and predictions of actions become domain specific. Online adaptations are performed through the respective final layers during tracking. In case of tracking failure due to low class confidence score for one of the actions taken in MDP, redetection using sampling is performed and we adapt to the features through their specific final

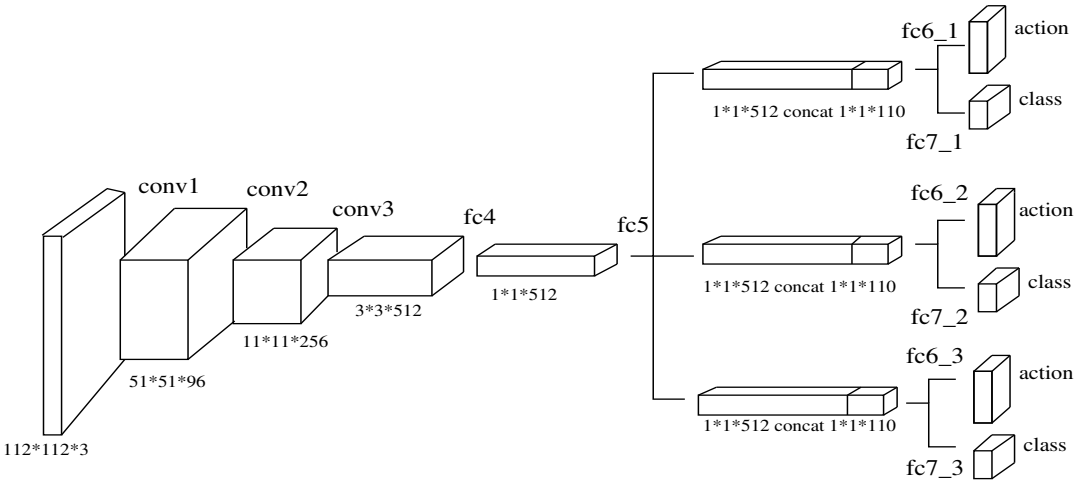


Figure 2: Proposed multi-object tracking network modified through ADNet.

branches. Figure 2 shows the final layers and the domain specific branches for action and class layers for three targets being tracked. As the number of targets increases, the number of branches initialized for tracking also increases. Once a track is said to have ended, then the corresponding branch is reinitialized with a new target when it appears.

We have modified a single object tracker for tracking multiple objects through online update. We shall now discuss our solution for handling MOT challenges in the next section.

3.1.3 Challenges in MOT Scenario

In case of tracking failure with the proposed method, the tracking process is continued from the redetected patch. If the class confidence score of the redetected patch were greater than 0.5, then we update the fully connected layers for adapting to changes in appearance of the patch that caused the tracking failure. However this method fails for targets with occlusions, frequent disappearance or reappearance and during target interactions, a common scenario in the case of multi-object tracking. This is because even if the score is greater than the threshold of 0.5, it does not indicate whether it is the entire object, partial object, cluttered background or noisy image due to occlusions.

However, performance of the tracker which depends on the threshold for class confidence score is kept as 0.5 for the proposed multi-object tracker. The justificaton for the selection of threshold is as follows:

1. During the MDP process, if we keep a higher threshold for class confidence score to continue tracking with linear actions, it leads to more failures in the tracking process. This is because the bounding box starts from the previous frame position and it requires a relaxation in the class confidence score to take actions to reach the target in the current frame.
2. Increasing the threshold for accepting the highest class confidence score based sampled patch during redetection is also not feasible. This is because fixing a high threshold for one of the targets may not be suitable for the other target which has a lower threshold for its full appearance. There is a need to select some dynamic threshold based on the target's features which is elaborated in section 3.2.2.

Thus, we keep our threshold at 0.5 and perform tracking. However, the tracker fails for multi-object tracking as explained below -

- Tracker fails to track the target through target drift, target loss, occlusions or confusion.
  - Drift can occur due to fast motion, blur, illumination variation etc.
  - Target loss occurs during disappearance of the target from the frame or during severe and long occlusions.
  - Confusions occur due to targets with similar appearance or cluttered background.
- Updating the tracker with other features leads to degradation of the model learnt online during short-term update. This is because the tracker only accounts for some of the recent past features which have chances of getting corrupted without the lack of target's temporal information.

As mentioned, there is a need to select a threshold for update based on the temporal and spatial features of the target.

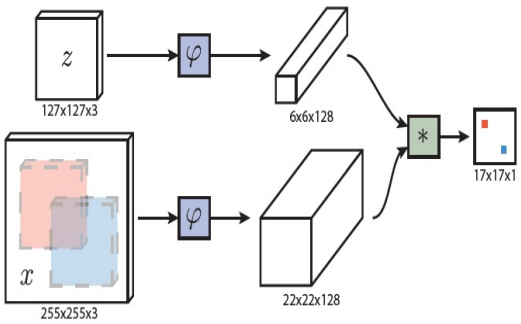


Figure 3: Fully convolutional siamese architecture. The network diagram is taken from (Bertinetto et al., 2016).

3.2 Handling Model Update and Target Drift

The issues arising from choice of class confidence score as 0.5 are handled by using correlation maps. We use correlation maps as an occlusion or noise attention mechanism. We give a brief overview of the network (Bertinetto et al., 2016) that is used for generating the correlation map for the ease of understanding.

3.2.1 Using Fully Convolutional Siamese Tracker

The Siamese architecture in Figure 3 is trained using similarity learning between a pair of positive and negative image pairs. The ground truth labels are generated from a pair of images (positive pairs) obtained from corresponding images of a video at most hundred frames apart.

The spatial map is obtained by correlating two images, the exemplar patch x and search image patch z using the below operation

$$f_\rho(x,z) = \phi(x) * \phi(z) + b \tag{3}$$

here $*$ is the cross correlation operator on the features of the exemplar image x and the search image z, by applying the function ϕ obtained by the trained network . As the network is fully convolutional, there is no risk of learning a bias on the centre of the search image, even if training is with ground truth maps centered on the positive search image.

The output of this network is not a single score between the target and the search image. Instead it is a list of scores between each translated sub-window in the search area with the target image, obtained through a single forward pass.

3.2.2 Handling Occlusions through Generated Correlation Maps

In case of tracking failure we use the generated correlation maps to determine a dynamic threshold for each of the targets. The maps are generated between an exemplar patch (target template) representing the full appearance of the target and a search area patch centered on the redetected patch (as given in Equation 1) or the associated detection (explained in section 3.2.3).

The Siamese model was trained to generate embeddings that produces a high score for positive pairs and a low score for negative (dissimilar) pairs. We obtain high peaks for similar targets and low peaks for dissimilar targets.

The correlation map generated provides a spatial support in the search area region centered on redetected patch or associated detection. We also get a temporal reference with respect to a full target appearance to check for occlusions or noise. During tracking the map has different values for the same target depending on the features in the search area. The peak to side lobe ratio (PSR) as employed in the MOSSE tracker (Bolme et al., 2010) can serve the purpose of providing a measure for dynamic threshold on the correlation map. To calculate PSR we have taken 150×150 area around the peak value and performed the following calculation

$$PSR = \frac{R_{max} - \mu}{\sigma}, \tag{4}$$

where R_{max} is the peak value of the response map. μ and σ are defined as the mean and standard deviation of the side lobe area.

The dynamic threshold (θ) for each target is set by the PSR value obtained from the correlation map. This is set by the initial exemplar image (first target template) and the search area image centered on the first target template or successfully associated detection that represents the full target appearance.

3.2.3 Long Term Tracking through Detections

To perform long term tracking and tracking of new targets, we use detections from a pre-trained detector (Redmon and Farhadi, 2018) to automate the process of tracking with minimal human intervention.

For associating with the tracked objects, we centre the search area on the detections and correlate it with the target templates to get PSR values. After feature association we also perform a proximity check using intersection over union (IoU). The method for using the PSR values based on the set dynamic thresholds (θ) for each of the targets is as follows

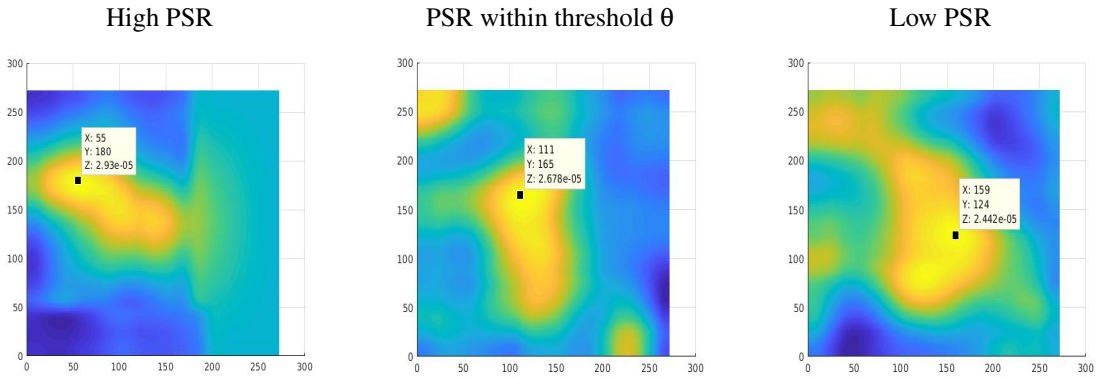


Figure 4: Correlation maps for PSR calculation.

1. Associate with an active track (object tracked successfully in previous frames and current frame)
   (a) Detections are associated with the track when PSR is less than or equal to θ. The track ID (identity) is assigned to the detection and θ value is reset using the detection features.
   (b) If none of the detections are associated with the track then the corresponding detection must have missed the frame. We then associate the track with the previous frame information and assign it the same track ID.
2. Associate with an inactive track (object tracked unsuccessfully in previous frames and current frame). If detection is associated with an inactive track, reassign θ with the detection template and change its state to active track.
3. When number of detections are more than the number of tracks, initialize new tracks for the unassociated detections.

4 SUMMARY OF PROPOSED MOT

The proposed tracker handles update through the mulit-domain network and performs re-identification on a search area maintained by the motion model. Our proposed method is called MDT_RCM (Multi-domain Tracking with Re-identification using Correlation Maps). We provide a summary of the proposed multi-object tracking method with the help of a pseudo code.

The assignment of bounding box as redetected patch for inactive state helps in re-identification as mentioned in step 18 of the algorithm. This is because inactive track's search area is near the target in cases where the motion model predicted linear actions successfully, but failed in the last few steps of the MDP. It is from this position given by the motion model that redetection is performed.

Algorithm 1: MDT_RCM.

1: Get bounding boxes for first frame using detections.
2: Initialize the first K detections through K multi-domain branches.
3: Set dynamic threshold values θ for each initial target.
4: **for** n=2 ; number of frames; n=n+1 **do**
5: Read image for frame n.
6: Track all the boxes parallelly through their corresponding multi-domain branches.
7: Associate active and inactive tracks with detections. Assign new tracks if necessary.
8: **for all** unsuccessful tracks **do**
9: Perform redetection by sampling as in eq 1.
10: Correlate with the target template and redetected patch to get PSR value.
11: **if** PSR $\leq$ θ **then**
12: **Active state:** Consider it a successful track. Collect positive and negative samples for online update.
13: **else**
14: **Inactive state :**Maintain track in inactive state till there is no re-identification or associated detection.
15: **if** Number of inactive states $>$ γ **then**
16: **Terminate state :** Remove this track. Here γ is a threshold for number of frames for which inactive state continues.
17: **end if**
18: Box position for tracking in the next frame is set to the redetected patch position.
19: **end if**
20: **end for**
21: **for all** active tracks **do**
22: Finetune network with samples collected in the recent past successfully tracked frames.
23: **end for**
24: **end for**

Table 1: Description of MOT16 dataset used for evaluation.

Name	Type of dataset	Camera	Description
MOT16-02	Training set	static	Elevated night view of pedestrian street, interacting objects, linear motion model, no long occlusions.
MOT16-04	Training set	static	People walking around a large square, full frontal view from camera, include far targets with cluttered background.
MOT16-13	Training set	dynamic	Camera mounted on a bus, frequent shaking and rotation of camera, provides an elevated view of cars and pedestrians.
MOT16-01	Test set	static	Side view of people walking around a large square, include both static and moving targets with interactions.
MOT16-06	Test set	dynamic	Street scene with moving platform, camera rotation and complex interactions with severe occlusions.

This helps in re-identification through successful tracking or redetection with acceptable PSR values in the consequent frames.

4.1 Generation of Pixel Masks

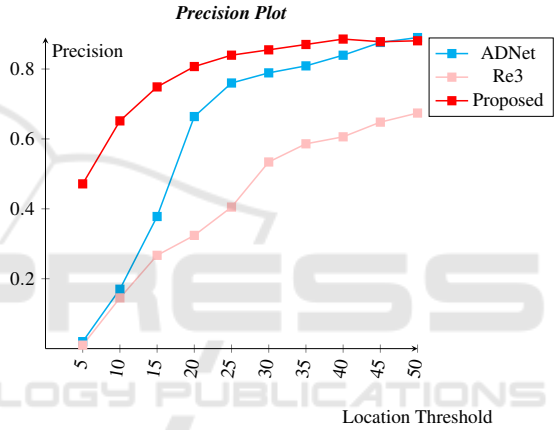
Using PolyRNN++ (Acuna et al., 2018), we have been able to obtain pixel masks for all the tracked objects in traffic video datasets. We have chosen a segmentation method that predicts pixel masks using polygon vertices which helps in easy correction by the human reviewer on an annotation tool.

5 RESULTS

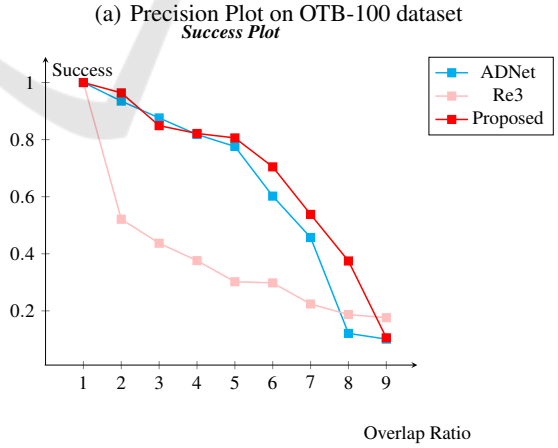
First, we improve our model update for the single object tracker on OTB-100 dataset (Wu et al., 2015). Second, we test the proposed method for multiple object tracking on MOT16 dataset (Milan et al., 2016). Third, we use PolyRNN++ (Acuna et al., 2018) model tested on the Cityscapes dataset (Cordts et al., 2016) on traffic video recordings. The tracker and instance segmentation network are then combined in a custom label tool to provide the annotator with a solution for instance aware segmentation of multiple objects in different traffic video datasets.

5.1 Dataset Description and Evaluation Metrics

OTB-100 dataset consists of 100 video sequences covering different challenges like illumination and scale variations, fast motion, motion blur, occlusions, deformations etc. MOT16 dataset has severe occlusions, interacting targets and frequent disappearance and appearance of objects. Table 1 provides a summary of the nature of objects in MOT16 dataset (Mi-



(a) Precision Plot on OTB-100 dataset



(b) Success Plot on OTB-100 dataset

Figure 5: Performance of single object tracking on OTB-100 dataset. Here the proposed method refers to MDT_RCM.

Figure 6: Multi-object tracking results on OTB-100 (top row) and MOT16 (bottom row) datasets.

lan et al., 2016) we use for evaluating our method.

To evaluate the performance of the tracker we have used the precision and success plots of OPE (One pass evaluation) (Wu et al., 2013).

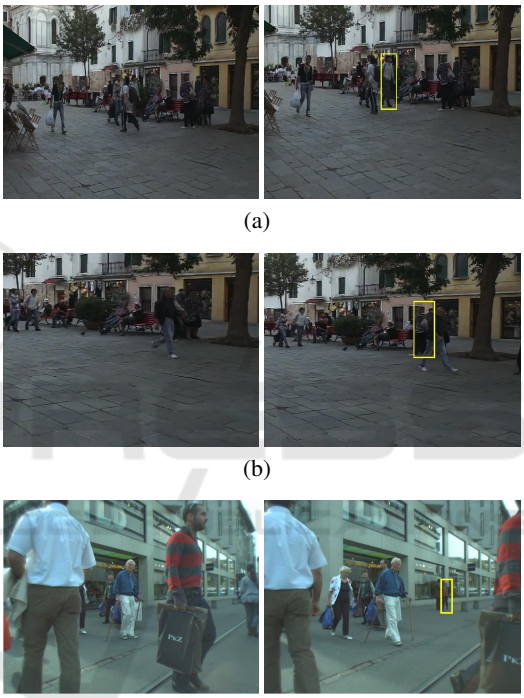
1. **Precision Plot** (Wu et al., 2013)**:** This plot calculates the percentage of frames within a location error threshold. The location value is given by the Euclidean distance between the centers of the tracked targets and the manually labeled ground truths.
2. **Success Plot** (Wu et al., 2013)**:** For this plot, we calculate the IoU of the tracked box with the ground truth box and check for the percentage of frames whose overlap is greater than a certain threshold.

5.2 Multi-object Tracking

Improving the model update and target drift improves the single object tracker as compared to the original ADNet and Re3 trackers. From the graphs in Figure 5, we see that the precision values at location error threshold of 20 pixels are 74.2, 40.5 and 80.7 for ADNet, Re3 (Gordon et al., 2018) and the improved ADNet tracker respectively. And the success rate at an overlap ratio of 0.5 are 78.6, 30.8 and 80.6 respectively.

The results on evaluating for multi-object tracking using MDT_RCM is shown in Figure 6. This is shown for two objects from the OTB-100 dataset and five objects from the MOT16 dataset.

Figure 7 shows re-identification of tracks where the left column refers to inactive tracks when PSR is below threshold and the right column shows re-identification when PSR is above threshold. Figure 7 (a) show the process of tracking through moving objects, where the model is not updated with features of the occluding person. Figure 7 (b) demonstrate successful tracking of a stationary and moving object. Finally, Figure 7 (c) indicate tracking in the case of moving cameras along with severe occlusion.



(a)

(b)

(c)

Figure 7: Re-identification of tracks in MOT scenario.

Figure 8 shows the performance of MDT_RCM on MOT16 dataset for both precision and success plot, obtained by averaging over all objects for all frames of the video. The values for precision and success ratios are shown in Table 2. The results vary in improvements on different videos on an average over varying number of objects and scenarios. We see that the proposed method performs well in general on success rates due to better motion model. As it is capable of re-identification, MDT_RCM shows better improvement on the precision rates over other methods.

The MDT_RCM has performed well in MOT scenarios with promising results. The timing performance of the proposed method has been improved using parallel computing from 2.9 s per object (AD-

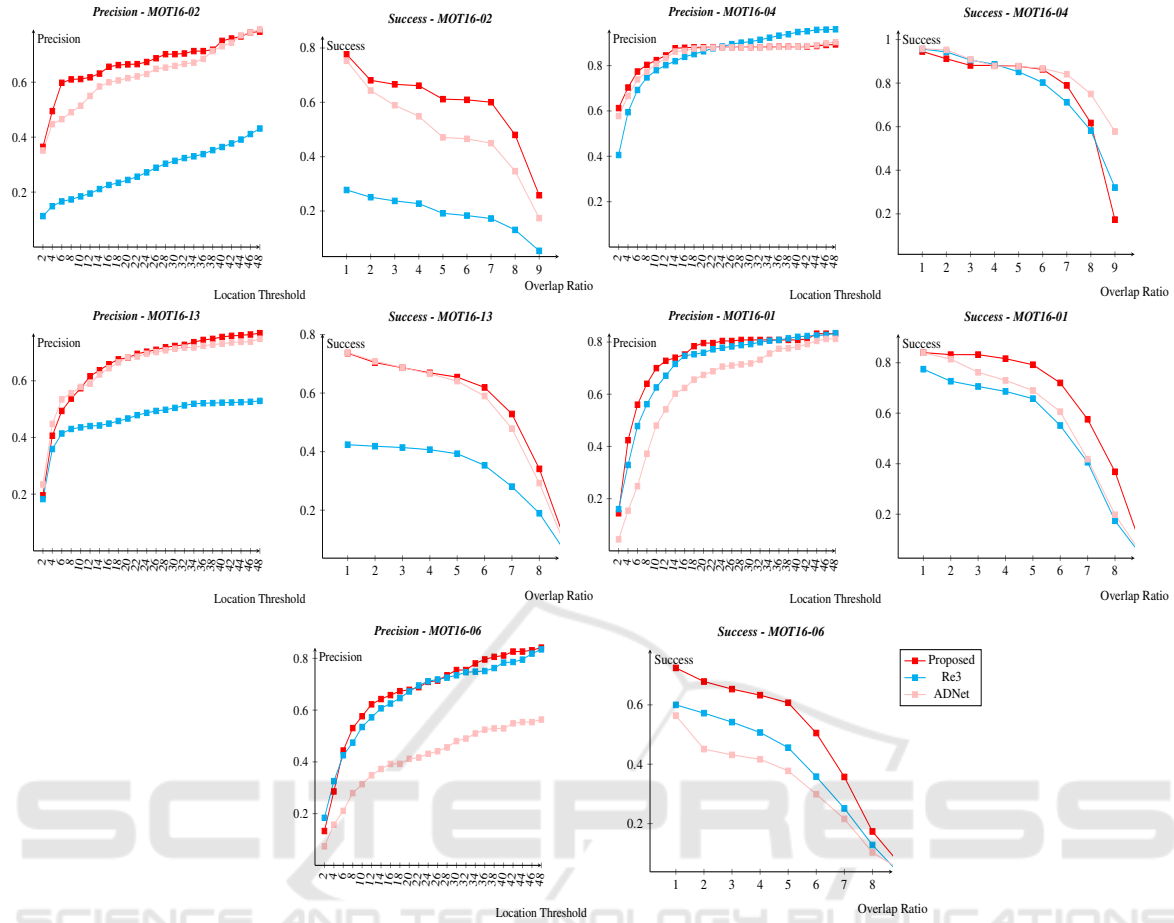


Figure 8: Precision and Success Plots for multi-object tracking on MOT16 datasets. In these plots the proposed method refers to MDT_RCM.

Table 2: Results on MOT16 training and test sets.

VideoSet	ADNet	Re3	Proposed method	ADNet	Re3	Proposed method
	Precision			Success		
MOT16-02	61.51	24.42	**66.38**	47.10	19.24	**61.17**
MOT16-04	87.69	86.28	**88.04**	87.69	85.17	**87.72**
MOT16-13	68.14	46.69	**68.23**	64.10	39.28	**65.45**
MOT16-01	67.40	75.9	**79.6**	69	65.77	**79.2**
MOT16-06	41.18	67.20	**67.86**	37.75	45.58	**60.71**

Net) to 0.6 s per object (MDT_RCM). Compared to the real-time Re3 tracker (0.3 s per object), it is a slower offline tracker which assigns the same track ID to all moving objects and has the advantage of re-identification.

5.3 Instance Aware Segmentation

We combine the multi-object tracking and segmentation method through a custom label tool, where the output of the tracker is fed to the segmentation network using a communication protocol (Varda, 2008) and pixel masks are obtained simultaneously for all the objects in the frame.

Figure 9 shows the masks of two objects using our multi-object tracker and segmentation algorithm displayed on a custom label tool. The annotator can adjust the box or mask when necessary on the annotation tool while reviewing the ground truth generated.

The annotator can choose to update the model of MDT_RCM for a slight correction in the bounding box and improve the accuracy of a particular target

(a) Bounding box and polygon mask (b) Polygon vertices outlining the target

Figure 9: Displaying the tracked boxes and pixel masks on annotation tool.

for future frames. In this case, the segmentation algorithm is also connected to the tool through the protocol to display the new pixel masks.

6 CONCLUSIONS

We have automated the process of segmenting multiple moving objects with instance-aware labels using deep learning techniques. To achieve this we proposed a two-fold solution, firstly to obtain tracks for multiple objects and secondly to use a segmentation algorithm on the boxes obtained.

We developed a scalable multi-domain network for tracking multiple objects in traffic video datasets called MDT_RCM (Multi-domain Tracking with Re-identification using Correlation Maps). It performs well in tracking multiple objects along with successful re-identification in MOT scenarios with high precision and success rates.

The initial layers of the CNN in ADNet can be improved upon for multi-object tracking by using Region Proposal Networks (RPN) for foreground and background classification. Anchor boxes for simultaneously predicting the boxes for all objects in the image could provide an advantage in terms of speed over parallel computing. The Siamese architecture could be used a part of the RPN network for tracking (Li et al., 2018) without any online update. An LSTM can also be used for predicting a series of actions instead of using a CNN network based on Markov Decision Process.

From the output of the tracker, we obtained identity aware pixel masks using PolyRNN++ as the segmenation algorithm. Both the boxes and segmentation masks generated can be visualized on an annotation tool to help in completing the labeling for an entire recording with minimal human effort and interactive corrections.

This solution can be used in diverse applications involving video data processing. Motion detection, intrusion detection, suspicious behaviour analysis, security access point monitoring, vehicle monitoring, parking management and people counting are a few examples of where the core solution can be applied.

REFERENCES

Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*, 34(11):2274–2282.

Acuna, D., Ling, H., Kar, A., and Fidler, S. (2018). Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 859–868.

Andriluka, M., Roth, S., and Schiele, B. (2008). People-tracking-by-detection and people-detection-by-tracking. In *2008 IEEE Conference on computer vision and pattern recognition*, pages 1–8. IEEE.

Babenko, B., Yang, M.-H., and Belongie, S. (2009). Visual tracking with online multiple instance learning. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 983–990. IEEE.

Bertinetto, L., Valmadre, J., Henriques, J. F., Vedaldi, A., and Torr, P. H. (2016). Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer.

Beucher, S. and Meyer, F. (1993). The morphological approach to segmentation: the watershed transformation. *Mathematical morphology in image processing*, 34:433–481.

Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2544–2550. IEEE.

Chu, Q., Ouyang, W., Li, H., Wang, X., Liu, B., and Yu, N. (2017). 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*, pages 4836–4845.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223.

Dredze, M. and Crammer, K. (2008). Online methods for multi-domain learning and adaptation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 689–697. Association for Computational Linguistics.

Gordon, D., Farhadi, A., and Fox, D. (2018). Re3: Re altime recurrent regression networks for visual tracking of generic objects. *IEEE Robotics and Automation Letters*, 3(2):788–795.

Grabner, H., Grabner, M., and Bischof, H. (2006). Real-time tracking via on-line boosting. In *Bmvc*, volume 1, page 6.

Grabner, H., Leistner, C., and Bischof, H. (2008). Semi-supervised on-line boosting for robust tracking. In *European conference on computer vision*, pages 234–247. Springer.

Hare, S., Golodetz, S., Saffari, A., Vineet, V., Cheng, M.-M., Hicks, S. L., and Torr, P. H. (2015). Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence*, 38(10):2096–2109.

Held, D., Thrun, S., and Savarese, S. (2016). Learning to track at 100 fps with deep regression networks. In *European Conference on Computer Vision*, pages 749–765. Springer.

Jepson, A. D., Fleet, D. J., and El-Maraghi, T. F. (2003). Robust online appearance models for visual tracking. *IEEE transactions on pattern analysis and machine intelligence*, 25(10):1296–1311.

Jiang, H., Fels, S., and Little, J. J. (2007). A linear programming approach for multiple object tracking. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE.

Kalal, Z., Matas, J., and Mikolajczyk, K. (2010). Pn learning: Bootstrapping binary classifiers by structural constraints. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 49–56. IEEE.

Li, B., Yan, J., Wu, W., Zhu, Z., and Hu, X. (2018). High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980.

Maninis, K.-K., Caelles, S., Pont-Tuset, J., and Van Gool, L. (2018). Deep extreme cut: From extreme points to object segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 616–625.

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

Milan, A., Roth, S., and Schindler, K. (2013). Continuous energy minimization for multitarget tracking. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):58–72.

Nam, H. and Han, B. (2016). Learning multi-domain convolutional neural networks for visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4293–4302.

Pinheiro, P. O., Collobert, R., and Dollár, P. (2015). Learning to segment object candidates. In *Advances in Neural Information Processing Systems*, pages 1990–1998.

Pirsiavash, H., Ramanan, D., and Fowlkes, C. C. (2011). Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*, pages 1201–1208. IEEE.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ross, D. A., Lim, J., Lin, R.-S., and Yang, M.-H. (2008). Incremental learning for robust visual tracking. *International journal of computer vision*, 77(1-3):125–141.

Tao, R., Gavves, E., and Smeulders, A. W. (2016). Siamese instance search for tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1420–1429.

Varda, K. (2008). Protocol buffers: Google's data interchange format. *Google Open Source Blog, Available at least as early as Jul*, 72.

Wu, Y., Lim, J., and Yang, M.-H. (2013). Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418.

Wu, Y., Lim, J., and Yang, M.-H. (2015). Object tracking benchmark. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1834–1848.

Xu, N., Price, B., Cohen, S., Yang, J., and Huang, T. (2017). Deep grabcut for object selection. *arXiv preprint arXiv:1707.00243*.

Yun, S., Choi, J., Yoo, Y., Yun, K., and Young Choi, J. (2017). Action-decision networks for visual tracking with deep reinforcement learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2711–2720.

Zhuang, B., Lin, G., Shen, C., and Reid, I. (2016). Fast training of triplet-based deep binary embedding networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5955–5964.