REALTIME MULTI-OBJECT TRACKING AND PIXELWISE SEGMENTATION

Undergraduate graduation project report submitted in partial fulfillment of
the requirements for the
Degree of Bachelor of Science of Engineering
in

The Department of Electronic & Telecommunication Engineering
University of Moratuwa.

Supervisor:
Dr Ranga Rodrigo
Dr Sadeep Jayasumana

Group Members:
150507H
150360A
150273J
150504V

February, 2020

Approval of the Department of Electronic & Telecommunication Engineering

. . . . . .
Head, Department of Electronic &
Telecommunication Engineering

This is to certify that I/we have read this project and that in my/our opinion it is fully adequate, in scope and quality, as an Undergraduate Graduation Project.

Supervisor: Dr Ranga Rodrigo

Signature: . . . . . .

Date: . . . . . .

Declaration

This declaration is made on February 15, 2020.

Declaration by Project Group
We declare that the dissertation entitled Project Name and the work presented in it are our own. We confirm that:

- this work was done wholly or mainly in candidature for a B.Sc. Engineering degree at this university,
- where any part of this dissertation has previously been submitted for a degree or any other qualification at this university or any other institute, has been clearly stated,
- where we have consulted the published work of others, is always clearly attributed,
- where we have quoted from the work of others, the source is always given,
- with the exception of such quotations, this dissertation is entirely our own work,
- we have acknowledged all main sources of help,
- parts of this dissertation have been published. (see List of Publications).

. . . . . .
Date

. . . . . .
W M D K N Ranasinghe(150507H)

. . . . . .
S C Liyanarachchi(150360A)

. . . . . .
B M H Jayawardhana(150273J)

. . . . . .
K D H P Ranasinghe(150504V)

Declaration by Supervisor

I/We have supervised and accepted this dissertation for the submission of the degree.

.....	.....
Dr Ranga Rodrigo	Date
.....	.....
Dr Sadeep Jayasumana	Date

Abstract

REALTIME MULTI-OBJECT TRACKING AND PIXELWISE SEGMENTATION

Group Members: W M D K N Ranasinghe, S C Liyanarachchi, B M H Jayawardhana, K D H P Ranasinghe

Supervisors: Dr Ranga Rodrigo, Dr Sadeep Jayasumana

Keywords: Vision, Perception, Detection, Tracking, Panoptic Segmentation, Siamese Network, Conditional Random Field, Recurrent Neural Network, Autonomous Systems.

Bleeding-edge technological pursuits ranging from self-guided robots at the research stage to mass scale industrial applications such as augmented reality, intelligent security systems and self-driving vehicles heavily rely on perception through vision. Vision based perception of the environment in autonomous systems extensively use object detection, segmentation and more importantly tracking as fundamental components. Despite the recent advancements in deep learning-based object detection on monocular images, the several highly publicized accidents involving self-driving vehicles and critical failures in monitoring and navigation systems highlight the need for significant further improvement for real-time tracking systems in critical applications. We identify two such key areas of improvement and introduce two separate novel frameworks to tackle each problem.

Firstly, we observe that trackers often perform underwhelmingly in object dense situations where occlusions and crossovers are prevalent. We identify that in order to perform better in these scenarios both appearance and motion information should be incorporated. Siamese networks have recently become highly successful at appearance based single object tracking while Recurrent Neural Networks (RNNs) have started dominating motion-based tracking. Our work focuses on combining Siamese networks and RNNs to exploit both (temporally varying) appearance and motion information to build a robust framework that can also operate in real-time. We further explore heuristics-based constraints for tracking in the Birds Eye View Space for efficiently exploiting 3D information as a constrained optimization problem for track prediction.

Our second observation is that most trackers lack precise (pixel-level) awareness of both object classes (countable) and back-ground classes in a frame. This is known as panoptic segmentation. We tackle the panoptic segmentation problem with a conditional random field (CRF) model. Panoptic segmentation involves assigning a semantic label and an instance label to each pixel of a given image. At each pixel, the semantic label and the instance label should be compatible. Furthermore, a good panoptic segmentation should have several other desirable properties such as the spatial and colour consistency of the labelling (similar looking neighbouring pixels should have the same semantic label and the instance label). To tackle this problem, we propose a CRF model, named Bipartite CRF or BCRF, with two types of random variables for semantic and instance labels. In

this formulation, various energies are defined within and across the two types of random variables to encourage a consistent panoptic segmentation. We propose a mean-field-based efficient inference algorithm for solving the CRF and empirically show its convergence properties. This algorithm is fully differentiable, and therefore, BCRF inference can be included as a trainable module in a deep network. In the experimental evaluation, we quantitatively and qualitatively show that the BCRF yields superior panoptic segmentation results in practice.

We integrate both these components in a joint tracking framework that is suitable for densely populated real world environments which are inherently chaotic, specifically in the domains of autonomous driving and fully automated stores.

Dedication

To our families, friends, supervisors, and all others that supported us in this work.

Acknowledgements

Add acknowledgements here!!!!!

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

Add all acronyms / abbreviations here

1 Introduction

Ability to track multiple objects simultaneously in real-time along with accurate (pixel-level) awareness for both stuff (background) and things (objects), are two of the fundamental vision-based challenges that modern autonomous and quasi-autonomous systems are faced with. We introduce two separate novel frameworks for both problems which improve on the current state-of-the-art and integrate them to form a joint 'perception' model capable of robust operation in object-dense and chaotic real-world scenarios.

1.1 Problem and Scope

Multi-object tracking has been a critical and unavoidable problem even at the level of cutting edge technology. State of the art multi-object tracking systems are computationally heavy for the end devices whereas real time tracking systems are performing at the expense of accuracy and even highly accurate systems [2] make errors in general and edge cases like occlusions, ego-motion, crossovers and rapid/random movements. The occlusions build up heavy risks in the automobile industry that looks forward for self driving cars. The inability to predict a pedestrian crossing behind the vehicle that slowed down on the side front or the incapability of the tracker to distinctly identify two persons at a point of crisscrossing can lead to critical issues in the field that is chaotic and requires memory other than detection for producing near intelligent results.

Most of the current systems are based on tracking through detection where the extensive development of efficiency and accuracy in state-of-the-art frame level detectors is used. The novel idea is to incorporate the temporal aspects into the algorithm due to the seemingly simple fact that objects do not disappear and should follow a time dependent progression within frame sequence given a satisfactory sampling rate in the sequence. Moreover, the improvement of depth sensation from both monocular and binocular image feeds [3] and new methods of inverse perspective mapping [4] build up the capacity to explore 3D tracking purely based on image data. This is important in the simultaneous localization of multiple real world objects with the observation of their dynamic aspects for decision making.

Through this work we develop an online real-time multi-object tracking system for efficient human and vehicle tracking through the exploitation of appearance and spatiotemporal information through a novel Long and Short Term Memory (LSTM) [5] based architecture along with the development of possible refinements to three dimensional track prediction through constraints observed in the Bird's Eye View (BEV) space.

Figure 1.1: In the above diagram the numbers on the top left represent the frame number with detection bounding boxes in red and track bounding boxes in other colors. The above diagram depicts the occlusion handling done by our system. It can be seen that track 5(blue) observed in frame 24 is well tracked through out even after been occluded, given that the maximum occlusion duration is less than 30 frames. Additionally tracks 17 and 19 (blue and purple) are reidentified in frame 183 after been occluded. This is because, we control how much the occluding image's features can affect the track's template when the track is partially occluded

Panoptic segmentation of images is a problem that has received considerable attention in computer vision recently. It combines two well-known computer vision tasks: semantic segmentation and instance segmentation. Panoptic segmentation differentiates between two types of semantic labels: *stuff* labels and *thing* labels. Stuff classes are semantic classes of shapeless regions of similar texture or material such as grass, sky, and road. Thing classes are semantic classes of countable objects such as people, animals, and vehicles [1]. The goal of panoptic segmentation is to assign a semantic label and an instance label for each pixel in the image. Clearly, the concept of instances is valid only for thing classes. Therefore, the instance label of a pixel labeled with a stuff semantic class is neglected.

Although semantic segmentation and instance segmentation are apparently very related problems, in the current state of the art methods in computer vision, they are solved in substantially different ways. The semantic segmentation problem is usually solved with a fully convolutional network architecture such as FCN [6] or DeepLab [7], whereas the instance segmentation problem is solved using an object detector based method such as Mask-RCNN [8]. Each of these architectures have their own strengths and weaknesses. For example, fully-convolutional network based semantic segmentation methods have a wide field of view, specially when used with dilated convolutions [9], and therefore can make semantic segmentation predictions with global information about the image. In contrast,

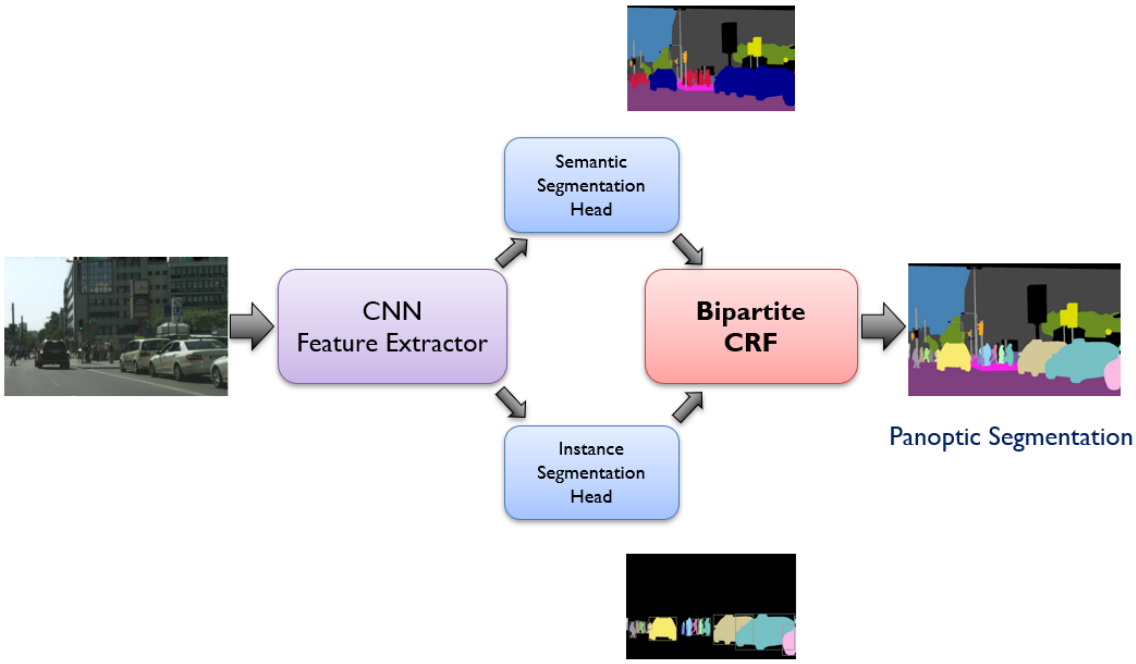

Figure 1.2: **BCRF in an end-to-end trainable deep net.** The Bipartite CRF proposed in this paper can be used to combine the predictions of a semantic segmentation model and an instance segmentation model to obtain a consistent panoptic segmentation.

region proposal based networks, such as Mask-RCNN, focus on specific regions of interest during the later stages of the network and make predictions using strong local features available within a given region of interest. It is natural to think of a systematic way of combining the complementary strengths of these two different approaches.

We propose a Conditional Random Field (CRF) based framework for panoptic segmentation. Our framework, named Bipartite Conditional Random Fields (BCRF), takes inputs from both a semantic segmentation module and an instance segmentation module, and uses additional prior ideas about a good panoptic segmentation. It then performs probabilistic inference on a graphical model to obtain the best panoptic label assignment given the semantic segmentation classifier, the instance segmentation classifier, and the image itself. Our framework provides a heuristic-free, probabilistic method to combine semantic segmentation results and instance segmentation results - yielding a panoptic segmentation with consistent labeling across the whole image. We formulate our bipartite CRF using different energy functions to encourage the spatial, appearance and semantic consistency of the final panoptic segmentation. The optimal labeling is then obtained by performing mean field inference on the bipartitle CRF - solving for both the semantic segmentation and the instance segmentation in a jointly optimal way.

Importantly, we show that our proposed BCRF inference is fully differentiable with respect to the various parameters used within the CRF and also the semantic segmentation and instance segmentation classifier inputs. Therefore, the BCRF module can be used as a first-class citizen of a deep neural network to perform panoptic segmentation. A deep network equipped with the BCRF module is capable of structured prediction of consistent panoptic labels and is end-to-end trainable. We show an example application of this framework and demonstrate that superior results can be used by probabilistic combination of a semantic segmentation classifier and an instance segmentation classifier in the BCRF framework.

1.2 Related Work

1.2.1 Single Frame Detectors

A considerable number of new network architectures have been developed for object detection and classification where growth in accuracy and speed had been the key goals. Out of the main state-of-the-art systems, the models based on Faster-RCNN [10] that had been developed from Fast RCNN [11] use a Region of Interest (RoI) to detect the objects and found to be highly accurate through the improvements with introduction of Region Proposal Network (Fully Convolutional Network that proposes regions). The single stage detectors like YOLO [12] (You Only Look Once) network on the other hand have been optimized for speed over accuracy. They explore the entire image as a whole grid instead of computing regions of interest and can achieve high performance in frame rate (exceeding 45fps). For the industry of self driving vehicles and the real-time automated market background both accuracy and speed are crucial. A notable aspect that each architecture has adopted to improve performance is approaching localization of an anchor-based classification task followed by regression as opposed to a purely regression task. This idea will serve as one of the baselines for our work.

1.2.2 Single Object Trackers

Tracking algorithms move for deep architectures (ex: Fully Convolutional Siamese) that use deep similarity learning for tracking [13, 14] to solve the key challenges of changes in lighting conditions, orientation and viewpoint. Extension of these methods to the multi-object setting is yet to be achieved. Further, some algorithms have been designed to learn online to track generic objects. However, learning online needs higher computational capacity at the end device which is not a luxury that could be afforded. The work by David Held et al. on GOTURN i.e. Generic Object Tracking Using Regression Networks [15]

depict the capability of achieving 100fps at test time with frozen weights. But their work is limited to single object tracking.

1.2.3 Joint Tracking and Detection

The novel idea in tracking and video recognition is the ability of improving the detection and tracking inter-dependently. That is to enhance the detection using temporal information and improve the track using both detection as well as temporal information. There has been significant progress in this area too [16, 17]. Tracking by detection had been a considerably successful topic in the field, but this aggregation of the temporal information has turned the efforts to a different path of exploration that could predict the next level of action which is steps beyond ordinary tracking. The common idea behind these methods is the usage of temporally aware feature maps for tackling the task of detection. The key shortcoming is the lack of direct track outputs which are a requirement for tracking.

1.2.4 Multi-Object Trackers

SORT (Simple Online and Real Time Tracking [18]) with a deep association metric [2] presents an implementation of the Kalman filter for exploiting the temporal information and a neural network incorporating the detections and deep appearance descriptor. The key challenge faced by this work is its failure to tackle crossovers, occlusions, and modeling non-linear object motion. Improvement of the temporal aspect using the LSTMs in single object setting [19] has presented promising results in catering to these problems. Further, the possibility of data association of random cardinality, specifically through the birth and death of characters (track initiation and termination) using LSTMs alone [19] is equally promising. The exploitation of multiple fields of view by relating deeper layers in Siamese networks [20] show the potential of Siamese matching even though it is considerably inhibited by the scenarios that have occlusions.

1.2.5 BEV space and 3D tracking

The methods for inverse perspective mapping and 3D detection have been extensively researched as means of depth sensation through both monocular [3, 4] and binocular [3] image feeds. The achievement of accuracy in depth sensation through images has approached the level of expensive range sensor data to a considerable extent. However, the task of 3D tracking is currently dominated by the algorithms that run on range sensor information [21].

1.2.6 Panoptic Segmentation

The task of semantic segmentation has historically captured much attention [22, 23, 24, 25] with multiple innovations emerging as a direct result [26, 27, 28]. With the popularity of deep convolutional feature extractors, multiple recent works have focused on multi-scale feature extraction [29, 28, 9, 30] and end-to-end structured predictions [31, 7, 32, 33, 34, 35] to better solve this task. While the former allows networks to capture objects of all scales, the latter allows better granularity in outputs. Further, the wider field of view in these networks, especially since the introduction of dilated convolutions [9, 30], provides better contextual understanding that directly benefits the task of semantic segmentation. This greater awareness of global information is a key uniqueness of most recent works. Also note how multiple approaches based on structured predictions [31, 34, 33, 36] have been highly successful in the task of semantic segmentation.

Along with the emergence of high accurate object detection works [37, 38], instance specific semantic segmentation started gaining significant attention. Early approaches use structured prediction based methodologies [39, 40], some often involving CRFs [41, 42]. With the advent of deep learning based approaches, instance segmentation methodologies have mostly taken the form of two-stage proposal based approaches [43, 44, 45, 46]. These methods were superseded by Mask R-CNN [8], laying the foundation for most current state-of-the-art instance segmentation approaches. Mask R-CNN builds off a conceptually simple extension of Faster R-CNN [37] obtained by adding a separate object mask prediction branch in parallel to the existing ones, capturing information local to each instance. This key contrasting feature is common even in later works built off Mask R-CNN [47]. Another similar recent work by Arnab *et al.* [48] moves in a slightly new direction by using a CRF to obtain instance segmentation outputs from a semantic segmentation using bounding box (from an object detection network) and instance shape cues. Our work differs from this in three significant ways: presence of pixel-wise cross potentials, using instance mask cues from a region-based network, and the ability to explicitly learn and model relationships between classes.

Since its formal introduction by Kirillov *et al.* [1], the task of panoptic segmentation has gained popularity, with multiple works attempting to transform existing network architectures to tackle this task [49, 50, 51, 52, 53, 54]. A key feature common among most of these works is fusing the logit outputs of existing semantic and instance segmentation networks to obtain a panoptic segmentation using some unique approach. The work of Arnab *et al.* [48] which emerged prior to this, may also be considered as an initial step in this direction. The work by Kirillov *et al.* [55] explores extending a feature pyramid

network [56] based Mask R-CNN [8] to output semantic segmentation as well, followed by heuristic based fusion to produce a panoptic output. Another similar approach is seen in the work by Xiong *et al.* [54] where the outputs are combined using a simple resizing and addition of semantic and instance logits alongside a method to output additional unknown labels for difficult pixels. Our work differs from these approaches with the inclusion of a CRF based layer for combining the two semantic and instance heads.

Conditional Random Fields (CRFs) are known as excellent models for structured prediction tasks such as semantic segmentation. Early works that used CRFs for semantic image segmentation includes [41, 42]. Most of these early methods of CRFs for semantic segmentation used 4-connected or 8-connected locally connected graphs. In [57], the authors proposed an efficient mean field based inference algorithm to solve fully connected CRFs with Gaussian edge potentials. The authors of [31] later showed that this CRF inference algorithm can be formulated as a Recurrent Neural Network (RNN). This module, known as CRF-RNN, was plugged into a fully convolutional network to obtain the state-of-the-art in semantic image segmentation at the time. Similar trainable CRF models have been used in works such as [58], for semantic segmentation with higher-order potentials and, [48] for instance segmentation. In [59], where the problem of panoptic segmentation with weak and semi supervision was addressed, the authors used a CRF for refining instance segmentation labels. However, it worked on homogeneous instance labels only and therefore was similar in spirit to previous fully connected CRFs.

In our work, we propose a bipartite CRF operating on the semantic segmentation task and the instance segmentation task *simultaneously*. This CRF has energies within semantic segmentation labels, energies within instance segmentation labels, and also energies *across* semantic and instance segmentation labels. To the best of our knowledge, this is the first time a bipartite CRF with cross connections between semantic and instance labels has been proposed in the context of pixel-wise labeling.

1.3 Method of Investigation

add stuff here

1.4 Principal Results

add stuff here

2 Methodology

2.1 Overview

The two main core objectives of our work involve in producing an online multi object tracker, which incorporates the spatiotemporal coherence of object motion together with the appearance consistency and producing a panoptic segmentation of the video feed by taking into account the bipartite potentials.

2.2 Multi Object Tracker

In this section, we describe our approach for online real-time multi-object tracking. The core of our work surrounds three key attributes; an LSTM network tackling track position estimates as a probabilistic classification problem, a methodology for similarity extraction and track association that is aware of occlusions, crossovers, and other identified key challenges and finally, the extension of track predictability to the BEV space exploiting its properties. Further, we explore the possibility of propagating input uncertainties through the LSTM network. The naive integration of a generic LSTM network to exploit the temporal aspect overlooks some key aspects of the problem including uncertainty of detection positions and requirement for estimating a possible region of object presence. To address this work, we introduce an LSTM network with probabilistic outputs also capable of capturing the input uncertainties. Our final model shown in 2.2 performs on-par with current state-of-the-art object trackers and operates in real-time. Our model is tested on popular tracking datasets, MOT16 [60] and KITTI-tracking [61]. These two datasets are from slightly different domains (the former focuses on general indoor and outdoor scenes while the latter contains videos of roads taken from the perspective of a vehicle). This allows us to verify that our work solves a generalized tracking problem as opposed to a single-domain specific solution or being optimized onto a single dataset. Our work is explained under six subsections. The LSTM network along with our unique contribution is outlined initially. Then we lay out the appearance similarity usage in a multi-object domain followed by the track association problem and overall 2D tracker. Finally, we explain the extensibility of our LSTM model for accurate 3D track prediction and the improvements gained from BEV space constraints.

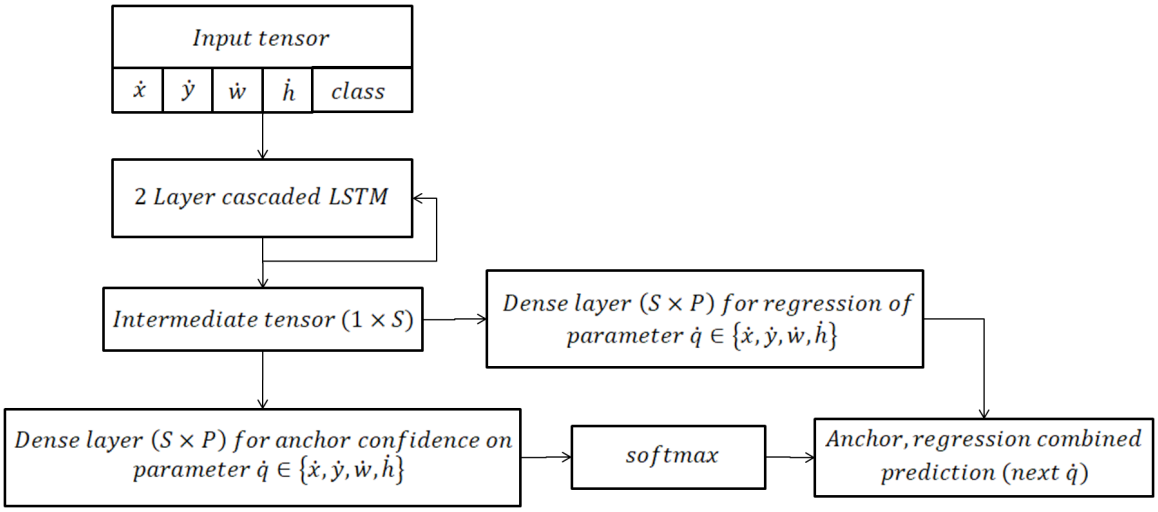

Figure 2.1: Structure of the proposed LSTM network

2.2.1 LSTM network

Consider a video as a sequence of image frames i.e. $V = I_0, I_1, ..., I_n$ where I_k s a matrix of fixed dimensions. Given detections $D = D_0, D_1, ..., D_n$, for the objects present in each frame , where each is a list of bounding box locations, class predictions, and other information corresponding to objects contained in the image , our goal here is to estimate the bounding box co-ordinates $B_{k,i}$ for each object in the following frame; I_{k+1}. Note that $B_{k,i} = (x_{k,i}, y_{k,i}, h_{k,i}, w_{k,i})$ where x, y, h, w correspond to x, y co-ordinates of centre, height and width of the bounding box for the i^{th} object in the k^{th} frame. Further, this system would operate in an online setting where at any given instance when time $t = k$, the frames, hence detections too are present only up to I_k and D_k espectively. Further the i^{th} object will be consistent across consecutive frames (obtained using the output of the system) until the object disappears. The LSTM component can be viewed as a function L with $L(D_0, D_1, ..., D_k) = F_k$ where F_k is a list of temporally aware feature maps $F_{k,i}$ corresponding to each object i present within D_k. The remaining two functions; C and R correspond to classification (selecting anchor) and regression (estimating deviation from anchor) of the exact bounding box targets. Each bounding box datum (x, y, h, w) is interpreted as a deviation from the previous time step $(\dot{x}, \dot{y}, \dot{h}, \dot{w})$ which reduces the mean of those variables. Note that due to the discrete nature of data, $\dot{x} = x - x_{i-1}$ Using normalized co-ordinates (x, y, h, w) values divided by relevant image dimensions); this range will be within $(-1, 1)$ and an optimum number of anchors can be used to estimate this value as a classification problem.

Having laid down the classifications on to the targets, the required estimates from the classification function would be a one hot encoded tensor; C_{out} of shape $(P, 4)$ for P bins of anchors and 4 bounding box parameters. In our work, we use four bins; 0, 0.1, 0.5, 0.8 leading to a $(4, 4)$ tensor where the bin closest to the target value (ex: $\dot{x}$) on each row would

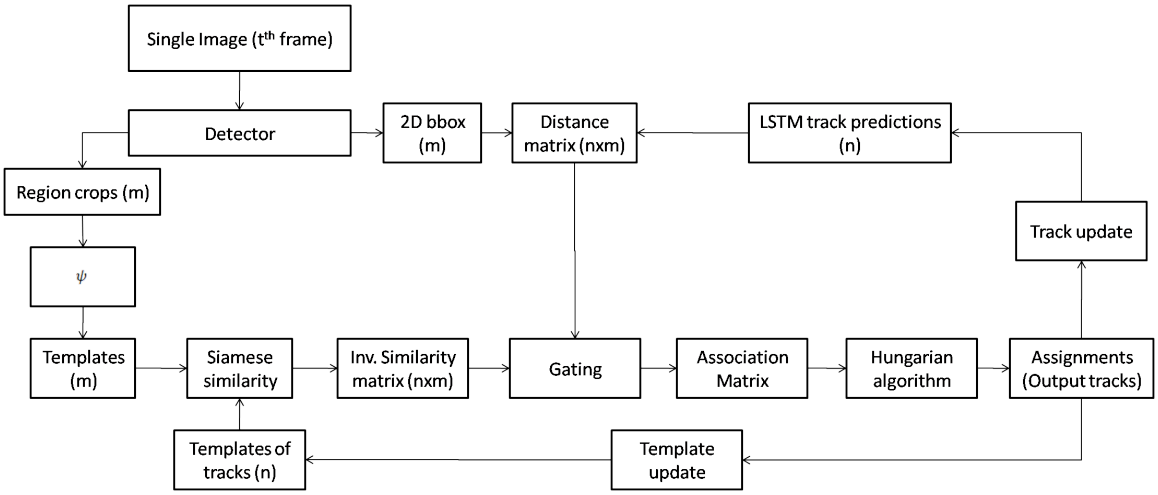

Figure 2.2: Overview of the overall 2D tracking system

contain one and the rest zero. Each selected bin is an anchor located at a specific distance away from the next expected value for the parameter considered. The classification function can be presented as $C(F_{k,i}) = C_{out}$ The regression function output would be a similarly shaped tensor R_{out}. It is essential for the loss function to consider the nature of both the classification as well as regression outputs of the network. The overall model of estimator is illustrated in 2.1. Here intermediate tensor corresponds to the temporally aware feature maps $F_{k,i}$ of the i^{th} object and the system has four similar but separate instances of the dense layers to handle each parameter and that finally results in outputs C_{out} and R_{out}. In essence the network estimates how far an object would move from its current position over the next time step. The x, y components capture motion along the image axes while the h, w components correspond to the motion along the depth axis as well as morphological change of the object to some extent.

When training; the loss function is obtained as a weighted sum of the classification and regression losses. The classification loss $LOSS_C$ is a simple cross-entropy loss function. The regression loss takes into account the sparse nature of the ground truth regression tensor. Here $\odot$ denotes the Hadamard product of two tensors or matrices.

$$LOSS_C = -\sum(C_{out_{true}} \odot log(C_{out_{pred}})) \tag{1}$$

$$LOSS_R = argmax(C_{out_{pred}}) \odot L_{Huber}(R_{out_{pred}}, R_{out_{true}}) \tag{2}$$

$$LOSS_{Total} = \lambda_C * LOSS_C + \lambda_R * LOSS_R \tag{3}$$

2.2.2 Appearance similarity

One of the most challenging problems in this context is handling occlusions. Object tracking with the use of a Kalman filter or an LSTM network to handle spatial coherence among

tracks has been a common approach. However, the uncertainty involved in the track prediction increases when tracks are exposed to prolonged occlusions. Hence it is required to re-identify occluded tracks. Deep SORT [2] introduces the use of feature vectors to define an appearance descriptor for the purpose of track re-identification. Results presented in this work have proven this to be a successful approach. However, this comes with the additional burden of training a large network for the sole purpose of re-identification of a particular class of objects. Hence this approach is not versatile for multi-class object tracking or for online implementation. Our approach has the ability to handle multiple classes of objects and can be implemented online with ease.

The approach implemented in this paper involves the use of a Siamese network to determine the appearance consistency of tracks. The Siamese networks described in the SiamFC [13, 62] and SiamMask [14] works have proven to be highly successful in single object tracking but have not been incorporated into multi-object tracking yet. It has been trained on ImageNet datasets for similarity learning and can operate online. Thus, it can give a class independent measure for appearance consistency of tracks and therefore would be ideal for track re-identification. The network discussed in SiamFC [62] extracts the features of the exemplar image and search image to produce a cross correlation map whose peak position corresponds to the position of the object in the exemplar image within the search image. Similarly, we use a Siamese network to produce a similarity measure between two images of the same size by building up templates through a convolution neural network (a convolution function as in [62] shown in template generation step through in 2.2)

The cross-correlation map produced by the Siamese network is passed through a similarity function to produce the similarity score, or more accurately an appearance cost. The similarity function in this context is defined as follows,

$$Appearance\ Cost = A \exp(-k \sum f(x, y)) \tag{4}$$

where $f(x, y)$ is the cross correlation value at x, y position in the cross correlation map and A, k are tunable parameters.

2.2.3 Track Association

The track association is based on the association cost which depends on the appearance cost (from the Siamese network) as well as a distance metric. The distance metric is the measurement of how far the detection bounding box is from the bounding box of a track predicted by the LSTM. The distance metric between two bounding boxes is defined based on the IOU distance (intersection over union) between the bounding boxes. Let $a_{i,j}, c_{i,j}, d_{i,j}$

represent the association cost, appearance cost and the distance metric between the i^{th} detection and the j^{th} track.

$$a_{i,j} = \begin{cases} c_{i,j} \; if \; d_{i,j} < T \\ K \; if \; d_{i,j} \geq T \end{cases} \tag{5}$$

where K is the gating constant and T is the gating threshold. Track association is treated as an assignment problem and is carried out using the Hungarian algorithm [63] following very closely the approach discussed in Deep SORT [2].

2.2.4 Overall online tracking system

The Siamese network for similarity measurement is implemented in two stages. The first stage involves producing feature maps (templates) for detections in the current frame and the next stage involves producing cross correlation maps by convolving the detection templates with track templates and generating an appearance cost matrix between track, detection pairs in that frame. These two stages have been isolated to improve the efficiency of the approach.

In a given frame, a crop of the bounding box corresponding to each detection is extracted. These crops are resized to 127x127 and passed through the first stage of the Siamese network to generate templates for each detection in that particular frame. These templates are passed through the second stage of the Siamese network along with the templates of tracks in order to generate a matrix of appearance costs. This cost matrix is gated according to the distance metric and subjected to the Hungarian algorithm to obtain track assignments for the detections.

For matched track, detection pairs; the template of the track is updated using a rolling average between the track's current template and the template of the detection which was matched to it,

$$temp_{track} = \gamma * temp_{track} + (1 - \gamma) * temp_{det} \tag{6}$$

where γ is the occluded percentage of the matched detection and defined as the maximum of the Intersection over area distances (IOA distances) between the detection bounding boxes and the bounding box of the matched detection. is one when fully occluded and zero when the object is fully visible. Therefore, when the matched detection is fully visible, it replaces the template of the track with the template of the matched detection and when the matched detection is fully occluded, it does not update the template of the track so as not to contaminate the template with the features from occlusions.

Deletion of tracks and addition of new tracks is carried very similar to the approach

carried out in the Deep SORT [2] work.

2.2.5 Extensibility to BEV space

The seemingly simple but effective fact that 'overlapping in BEV space projections cannot happen for the objects detected and predicted in 3D' is exploited here through a constrained optimization problem.

An LSTM network is trained to predict the change of parameter 'q' between consecutive frames. That is, for given $q_{t-k}, ..., q_{t-1}, \dot{q_t} \rightarrow q_{t+1}$ is predicted where $\dot{q_t} = q_t - q_{t-1}$ and $q \in (C, S, \theta)$. Here;$C = C_x, C_y, C_z$ (the centre co-ordinates of the object), $S = (h, w, l)$ (object dimensions) and θ is the angle of rotation around the vertical axis. The loss function for training the parameter predictor (LSTM) is as follows.

$$LOSS_{pred}(p, \beta, \alpha, \delta, \theta) = \sum_{i=1}^{N} \beta_{class_i} \left(\left(\sum_{p \in (C,S)} \alpha_p L_{Huber,\delta_p}(p_{pred}, p_{gt}) \right) + \alpha_\theta L_\theta(\theta_{pred}, \theta_{gt})_{object=i} \right) \quad (7)$$

Here P_{pred} refers to the predicted parameter and P_{gt} refers to the ground truth parameter.

δ_p is a parameter based learnable which in turn is the quadratic-linear margin of the Huber loss function and α_p or α_θ is a regressed parameter based learnable (where in the case of α_θ, the regressed parameter is θ and α_p is similarly interpreted whereas the scope of α_p is different from that of δ_p, considering the impact on cost function) and β_{classi} is the class based learnable parameter w.r.t. the class of the i^{th} object.

Here, $p = C_x, C_y, C_z, h, w, l$, $\beta = \beta_{class} | class \in classes$, $\alpha = [[\alpha_p]_{p \in parameters}, \alpha_\theta]$ and $\delta = [\delta_p]_{p \in parameters}$.

Due to the discontinuous nature of the parameter θ at the two extreme ends of its domain $[-\pi, \pi]$, and due to the fact that $\theta = \pi$ and $\theta = -\pi$ depict the same orientation, it is not directly incorporated into the Huber loss function. It is handled separately using L_θ function [64], where $\theta_{pred}, \theta_{gt}$ are predicted and ground truth values of the parameter θ respectively.

$$L_\theta(\theta_{pred}, \theta_{gt}) = 0.5(1 - \cos(\theta_{gt} - \theta_{pred})) \quad (8)$$

2.2.6 Constraints as penalties

First, we introduce the hard constraint on BEV space that projections of the objects on to the x-z plane in general co-ordinates have no intersection. However, most of the research is focused on building up 3D bounding boxes of objects where the rectangular projection does not create a clear cut segmentation of the object (ex: human) on BEV space. Therefore, we minimize an additional term as follows.

$$I = \sum_{v_i, v_j \in objects_{pred}, i \neq j} (1 + \xi^2_{class_i, class_j})(v_{i_{BEV}} \cap v_{j_{BEV}}) \tag{9}$$

Where $v_{i_{BEV}}$ is the projection of the bounding box of the object v_i onto the BEV space and $\xi_{class_i, classj}$ is a learnable based on object classes under intersection which in turn forms a set $\xi_{class \times class}$ and each term is squared to ensure positivity. Therefore, the final minimization function is as follows,

$$L(p, \beta, \alpha, \delta, \theta, \{\xi\}) = LOSS_{pred}(p, \beta, \alpha, \delta, \theta) + I \tag{10}$$

However, at an optimum point $(p^*, \beta^*, \alpha^*, \delta^*, \theta^*, \{\xi\}^*)$; the loss function obeys a feature observed in Lagrange constrained optimization that; $\nabla L = 0$ where ∇ refers to the discrete derivative (this statement is intuitive only with the discrete derivative).

This implies that:

$$\nabla_{p,\theta} Loss_{pred} = -(1 + \xi^2_{class_i, class_j}) \nabla_{p,\theta} (v_{i_{BEV}} \cap v_{j_{BEV}}) \tag{11}$$

for all classes at optimum parameters p^*, θ^*. Therefore $(1 + \xi^2_{class_i, class_j})$ behaves similar to a Lagrange multiplier. This setting helps to build up a network that trains not only based on the individual performance per object but also encountering the joint effect of multiple object scenarios.

2.3 Panoptic Segmentation

2.3.1 Background: Conditional Random Fields

Conditional Random Fields (CRFs) are a class of statistical modeling method used for structured prediction. A CRF, used in the context of pixel-wise label prediction, models pixel labels as random variables that form a Markov Random Field (MRF) when conditioned upon the image. CRFs have primarily been used in computer vision for semantic image segmentation. In this setting, CRFs encourage the desirable properties of a good

segmentation, such as the spatial consistency (e.g. spatially neighboring pixels should have the same label) and color consistency (e.g. a semantic segmentation boundary should correspond to a edge in the image) through various energy functions used in the formulation. A CRF formulation usually has energy terms arising from an imperfect classifier (sometimes known as the unary energy) and energy terms encouraging the consistency properties of the segmentation (sometimes known as the pairwise energy). Some semantic CRF models also include higher order energy terms to encourage higher order consistency properties such as consistency of the labeling within super-pixels [58].

Once an appropriate energy function is formed, the optimal labeling is found as the labeling that minimizes the CRF energy (or equivalently, maximizes the probability). This is known as the inference of the CRF. The exact inference of a CRF with dense pairwise connections is intractable and hence approximate inference methods such as mean field variational inference has to be utilized to solve the CRF in reasonable time [57]. For a detailed treatment of CRFs, the reader is referred to [65].

Algorithm 1 Inference on Bipartite CRF

$Q_i(l) := \text{softmax}_i(-\phi_i(l))$ and $R_i(t) := \text{softmax}_i(-\psi_i(t))$ ▷ Initialization
while not converged **do**
  $Q'_i(l) \mathrel{-}= \phi_i(l)$ ▷ Update due to the first term
  $Q'_i(l) \mathrel{-}= \sum_{l' \in \mathcal{L}} \left(\mu(l, l') \sum_{j \neq i} \text{Sim}_\Phi(i, j)\, Q_j(l') \right)$ ▷ Update due to the second term
  $R'_i(t) \mathrel{-}= \psi_i(t)$ ▷ Update due to the third term
  $R'_i(t) \mathrel{-}= \sum_{t' \in \mathcal{T}} \left([t \neq t'] \sum_{j \neq i} \text{Sim}_\Psi(i, j)\, R_j(t') \right)$ ▷ Update due to the fourth term
  $Q'_i(l) \mathrel{-}= \sum_{t \in \mathcal{T}} \left(f(l, \text{class}(t))\, R_i(t) \right)$
  $R'_i(t) \mathrel{-}= \sum_{l \in \mathcal{L}} \left(f(l, \text{class}(t))\, Q_i(l) \right)$ ▷ Updates due to the fifth term
  $Q'_i(l) \mathrel{-}= \sum_{t \in \mathcal{T}} \left(f(l, \text{class}(t)) \sum_{j \neq i} \text{Sim}_\Omega(i, j)\, R_j(t') \right)$
  $R'_i(t) \mathrel{-}= \sum_{l \in \mathcal{L}} \left(f(l, \text{class}(t)) \sum_{j \neq i} \text{Sim}_\Omega(i, j)\, Q_j(l') \right)$ ▷ Updates due to the sixth term
  $Q_i(l) := \text{softmax}_i \left(Q'_i(l) \right)$ and $R_i(t) := \text{softmax}_i \left(R'_i(t) \right)$ ▷ Normalization
end while

2.3.2 Bipartite CRFs

We propose a CRF formulation with bipartite random variables to capture interactions between semantic labels and instance labels. Inference of this CRF gives the jointly most probable semantic and instance segmentation (and therefore, the panoptic segmentation)

for a given image.

For each pixel i, define a pair of discrete random variables (X_i, Z_i) to denote its semantic label and the instance label, respectively. For each i, X_i can take values in $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$, where each l_j is a semantic label and L is the number of semantic labels (includes both stuff and thing classes). Therefore, $\mathcal{L} = \mathcal{L}_{\text{stuff}} \cup \mathcal{L}_{\text{things}}$, where $\mathcal{L}_{\text{stuff}}$ is the set of stuff class labels and $\mathcal{L}_{\text{things}}$ the set of thing class labels. Similarly, for each i, Z_i can take values in $\mathcal{T} = \{\texttt{inst}_0, \texttt{inst}_1, \ldots, \texttt{inst}_{N\,\text{inst}}\}$, where N inst is the number of instances detected in the image, and the label $\texttt{inst}_0$ is reserved to represent the "no instance" case (the pixel belongs to a stuff class).

Let $\mathbf{X} = [X_1, X_2, \ldots, X_N]$ and $\mathbf{Z} = [Z_1, Z_2, \ldots, Z_N]$, where N is the number of the pixels in the image. A joint assignment $(\mathbf{x}, \mathbf{z})$ to these two random vectors $(\mathbf{X}, \mathbf{Z})$ gives a unique semantic label and an instance label to each pixel i, and therefore represents a panoptic segmentation of the image. Note that, $\mathbf{x} \in \mathcal{L}^N$ and $\mathbf{z} \in \mathcal{T}^N$. In this work, we discuss the probability of such assignments and formulate the probability distribution function so that the "good" panoptic segmentation will have a high probability. We then perform inference on this formulation to find the assignment that maximizes the probability to obtain the best panoptic segmentation.

The probability of a panoptic segmentation $(\mathbf{x}, \mathbf{z})$, given the image I, can be modeled as a Gibbs distribution of the following form:

$$\Pr(\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}|I) = \frac{1}{\mathcal{Z}(I)} \exp(-E(\mathbf{x}, \mathbf{z}|I)), \tag{2.1}$$

where $\mathcal{Z}(I) = \sum_{(\mathbf{x},\mathbf{z})} \exp(-E(\mathbf{x}, \mathbf{z}|I))$, is a normalization constant, sometimes known as the partition function. The term $E(\mathbf{x}, \mathbf{z}|I)$ is known as the energy of the configuration $(\mathbf{x}, \mathbf{z})$. Hereafter, we drop the conditioning on I in the notation for brevity. The energy of our bipartite CRF is defined as follows:

$$\begin{aligned} E(\mathbf{x}, \mathbf{z}) = &\sum_i \phi(x_i) + \sum_{i<j} \Phi(x_i, x_j) + \\ &\sum_i \psi(z_i) + \sum_{i<j} \Psi(z_i, z_j) + \\ &\sum_i \omega(x_i, z_i) + \sum_{i<j} \Omega(x_i, z_j), \end{aligned} \tag{2.2}$$

where x_i and z_i are the elements of the vectors $\mathbf{x}$ and $\mathbf{z}$, respectively. The meaning of each term will be described in detail below. Note that, since a "good" panoptic segmentation should have a high probability, it should have a low energy. Various terms in Eq. (2.2)

should therefore encourage a good panoptic segmentation by penalizing disagreements with our prior knowledge about a consistent panoptic segmentation.

2.3.3 Semantic Component of the CRF

In the following, we discuss the first two term of the energy function in Eq. (2.2). The first term encourages the semantic segmentation result to be consistent with the initial classifier.

$$\phi(X_i = x_i) = -\log(\mathrm{Pr}_0(X_i = x_i)), \tag{2.3}$$

where $\mathrm{Pr}_0(.)$ is the classifier probability score for the semantic segmentation.

The second term in Eq. (2.2) encourages the smoothness of the semantic labeling:

$$\Phi(X_i = x_i, X_j = x_j) = \mu(x_i, x_j)\,\mathrm{Sim}_\Phi(i, j), \tag{2.4}$$

where $\mu : \mathcal{L} \times \mathcal{L} \to \mathbb{R}$ is the label compatibility function, and $\mathrm{Sim}_\Phi(i, j)$ is a similarity measure between the pixels i and j. This term penalizes assigning different labels to a pair of pixels that are "similar". Following [57], we use a mixture of Gaussians as the similarity measure. Therefore,

$$\mathrm{Sim}_\Phi(i, j) = \sum_m w_{\Phi,m} \exp\left(-\frac{\|\mathbf{f}_i^{(m)} - \mathbf{f}_j^{(m)}\|^2}{2\sigma_{\Phi,m}^2}\right) \tag{2.5}$$

where $\mathbf{f}_i$ is a feature vector for pixel i containing information such as its spatial location and bilateral features (RGB + spatial coordinates). We use the same spatial and bilateral features used in [57].

2.3.4 Instance Component of the CRF

For the instance classification, we also assume the existence of an initial classifier, such as Mask R-CNN, that provides a confidence score for each instance at each pixel. Note that Mask R-CNN provides fixed-size instance segmentation predictions with respect to the bounding boxes of the detections. However, these predictions can be easily mapped to the full image by using bilinear interpolation and trivial coordinate transforms.

In the following, we use $z_i \in \{\texttt{inst}_0, \texttt{inst}_1, \ldots, \texttt{inst}_N \,\mathrm{inst}\}$, where $N\,\mathrm{inst}$ is the number of instances detected in the image. The label $\texttt{inst}_0$ is reserved for the special case where the pixel does not belong to an instance, i.e., it belongs to a stuff class.

Similar to the semantic segmentation case, the third term in Eq. (2.2) encourages the

panoptic segmentation to be consistent with the instance classifier probabilities Pr_0:

$$\psi(Z_i = z_i) = -\log(\mathrm{Pr}_0(Z_i = z_i)). \tag{2.6}$$

The fourth term in Eq. (2.2) encourages instance label consistency across the whole image by penalizing assigning different instance labels to similar pixels:

$$\Psi(Z_i = z_i, Z_j = z_j) = [z_i \neq z_j] \, \mathrm{Sim}_\Psi(i, j). \tag{2.7}$$

The compatibility transform in this case is fixed to be $[z_i \neq z_j]$, where $[.]$ is the Iverson bracket. The similarity measure Sim_Ψ has a similar form to Eq. (2.5).

2.3.5 Cross Potentials in the CRF

An important contribution of this paper is the introduction of cross potentials between the semantic segmentation and instance segmentation. The semantic segmentation and the instance segmentation are highly related problems and therefore the solutions should agree: the semantic label at any pixel has to be compatible with the instance label at that pixel. For example, if the instance labeling says that the pixel i belongs to an instance of a person class, the semantic label at pixel i should also have the person label. If the initial classifier results for the instance segmentation and the semantic segmentation do not agree, one of them should correct itself depending on the interactions of other terms in the CRF.

The first cross potential term (the fifth term in Eq. (2.2)), encourages instance label and the semantic label at a given pixel to agree:

$$\omega(X_i = x_i, Z_i = z_i) = f(x_i, \mathrm{class}(z_i)). \tag{2.8}$$

Here, $\mathrm{class}(z_i)$ is the class label of the instance z_i with $\texttt{inst}_0$ mapped to a special class null. Note that, for all valid instances, the class label can be obtained from the instance classifier (e.g. Mask R-CNN). The function $f(.,.) : (\mathcal{L}, \mathcal{L}_{\text{things}} \cup \{\text{null}\}) \rightarrow \mathbb{R}_0^+$, captures the cost of incompatibility and is defined as follows:

$$f(x_i, \mathrm{class}(z_i)) = \begin{cases} 0, & \text{if } x_i = \mathrm{class}(z_i) \\ 0, & \text{if } x_i \in \mathcal{L}_{\text{stuff}} \text{ and } \mathrm{class}(z_i) = \mathrm{null} \\ \eta(x_i, \mathrm{class}(z_i)), & \text{otherwise.} \end{cases} \tag{2.9}$$

The above function covers three cases: 1) If the semantic label and the class label of the instance label match, there will be no penalty for such assignment since there is no incom-

patibility in this case. 2) If the semantic segmentation assigns a stuff label and the instance segmentation assigns $\texttt{inst}_0$ label, there will be no penalty in that case either. 3) If the semantic label and the instance label mismatch, there will be a penalty with the magnitude decided by the function $\eta(.,.) : \mathcal{L}_{\text{things}} \cup \{\text{null}\} \times \mathcal{L}_{\text{things}} \cup \{\text{null}\} \rightarrow \mathbb{R}^+$. This function is learned from data as described in Section 2.3.6.

The last term in Eq. (2.2), encourages the consistency of semantic label and the instance label among similar looking pixels and has the form:

$$\Omega(X_i = x_i, Z_j = z_j) = f(x_i, \text{class}(z_j))\ \text{Sim}_\Omega(i, j), \tag{2.10}$$

where each symbol has the meaning described above.

2.3.6 Inference and Parameter Optimization

The best panoptic segmentation given the model described in Section 2.3.2 is the assignment $(\mathbf{x}, \mathbf{z})$ that maximizes the probability in Eq. (2.1). However, since the graphical model used in BCRF has dense connections between the pixels, the exact inference is infeasible. We therefore use an approximate parallel mean field inference algorithm following [57].

In this setting, the joint probability distribution is approximated by the product of marginal distributions:

$$\Pr(\mathbf{X} = \mathbf{x}, \mathbf{Z} = \mathbf{z}) \approx \prod_i Q_i(x_i)\, R_i(z_i), \tag{2.11}$$

where $Q_i(x_i) = \Pr(X_i = x_i)$ and $R_i(z_i) = \Pr(Z_i = z_i)$ are the marginal distributions. Out of all the distributions that can be written down in this factorized form, the closest distribution to the original joint distribution is found by minimizing the KL divergence [65, 57]. For our BCRF formulation, this results in the iterative algorithm detailed in Algorithm 1.

To make our model flexible, we deliberately include a number of parameters in the BCRF model, which we automatically learn from the training data. More specifically, the BCRF model has the following parameters:

1. Weight multipliers for different energy terms: each term in Eq. (2.2) is multiplied with a weight parameter, which decides the relative strength of the term. This parameterization helps learn the optimal combination of different energies in the CRF. For example, if the initial semantic segmentation model has better accuracy than the instance segmentation model, the ϕ unary energy might be weighted more than the ψ unary energy.

Method	PQ	SQ	RQ
DeeperLab [66]	67.35	-	-
Ours (baseline)	70.50	88.65	78.83
Ours (CRF only)	67.72	87.62	76.48
Ours (BCRF)	71.76	89.63	79.33

Table 2.1: Comparison of results on Pascal VOC dataset. The baseline used contains DeepLab-v3 for semantic branch and Mask-RCNN for instance branch followed by combination using the simple logical method outlined in [1]. CRF only corresponds to setting the BCRF cross-potential terms to zero. BCRF is our complete network.

2. Parameters for similarity functions: Each similarity function $\mathrm{Sim}_X(i, j)$ of the form shown in Eq. (2.4) has its own parameters. These learn the relative strength of spatial and appearance consistency of the panoptic segmentation.

3. Label compatibility matrices: The two functions $\mu(.,.)$ and $\eta(.,.)$ are initialized to have a zero cost for a pair identical labels and a fixed cost for any combination of two different labels. They are then given the freedom to automatically learn the relative penalty strengths for different label combinations.

2.3.7 BCRF in a Deep Network

In this section, we discuss how BCRF can be used in a deep network. In [31], authors showed that, in the semantic segmentation setting, mean field inference of a CRF with Gaussian pairwise potentials can be formulated as a Recurrent Neural Network (RNN). Since our BCRF also uses an iterative mean field algorithm of similar nature, it is readily adaptable into the RNN based inference described in [31]. Therefore, BCRF can be a first-class citizen of a deep network performing panoptic segmentation. Importantly, this formulation allows automatic optimization of the BCRF parameters described in Section 2.3.6, using backpropagation and a gradient descent algorithm such as stochastic gradient descent (SGD). This is a major advantage since it allows us to increase the number of parameters used in BCRF, and hence increase its flexibility, without adding to the burden of manual parameter optimization.

In the current state-of-the-art methods, semantic segmentation and instance segmentation are solved with different network architectures with complimentary strengths. The BCRF formulation given a systematic way of combining these strengths in a probabilistic framework. Such an example usage of BCRF is shown in Figure 1.2. The CNN feature extractor here can be a common backbone network such as ResNet-101 or ResNeXt. The semantic segmentation branch is usually a fully convolutional network that is capable of

seeing a wide field of view, whereas the instance segmentation branch is a region-proposal based network such as Mask R-CNN. The semantic segmentation branch's output is taken as the ϕ unary potential input to the BCRF, and instance segmentation branch's output as the ψ unary potentials. In addition, the raw image is also fed into the BCRF to derive the similarity functions $\mathrm{Sim}_X(.,.)$ using the pixel locations and the RGB values.

During the training of the network, in the forward pass, BCRF inference is performed using Algorithm 1. A suitable loss function for panoptic segmentation can then be used at the output of the network. In the backward pass, differentials with respect to the loss function will be passed into the BCRF inference to optimize various parameters used in the BCRF model. Importantly, during the backward pass, after BCRF inference, the error differentials can be passed on to the semantic branch and the instance branch both to optimize their parameters, and subsequently, the feature extractor CNN's parameters. Therefore, the whole network, including the BCRF component, can be jointly trained.

3 Results

3.1 Multi Object Tracking Evaluation

Table 3.1: Comparison of our performance on MOT16 dataset with recent works

Method	Mode	MOTA↑	MOTP↑	MT↑	ML↓
Deep SORT [2]	ONLINE	61.40%	79.10%	32.80%	18.20%
SORT [18]	ONLINE	59.80%	79.60%	25.40%	22.70%
RNN LSTM [19]	ONLINE	19.00%	71.00%	05.50%	45.60%
MDP [67]	ONLINE	30.30%	71.30%	13.00%	38.40%
DMAN [68]	ONLINE	46.10%	73.80%	17.40%	42.70%
LSTM+Similarity (Ours)	ONLINE	66.70%	69.00%	39.18%	16.80%
Kalman Filter (Ours)	ONLINE	61.00%	69.00%	17.00%	17.00%

Table 3.2: Comparison of our performance on KITTI-trracking dataset with recent works

Method	Mode	MOTA↑	MOTP↑	MT↑	ML↓
Regionlets Only [69]	ONLINE	76.40%	81.50%	54.10%	9.30%
MS-CNN Only [69]	ONLINE	81.23%	85.60%	66.30%	4.60%
Regionlets MS-CNN [69]	ONLINE	82.60%	85.00%	70.50%	5.30%
SMES [70]	ONLINE	70.78%	80.38%	51.68%	7.77%
LSTM + Similarity (Ours)	ONLINE	83.58%	78.50%	48.23%	2.25%

3.1.1 Datasets and Evaluation metrics

Experiments are conducted on the MOT16 [60] and KITTI [61] tracking datasets. The MOT16 dataset contains 7 videos in its training set. The KITTI tracking dataset contains 21 videos in its training set. The Siamese Network for appearance consistency is trained completely on external data (ImageNet datasets) and there is no overlap with any of the MOT16 or KITTI data. The LSTM network is trained only with the use of bounding box locations of objects and class information for a partition of the training sets of these two datasets (the remainder is kept aside for testing purposes). Results are reported for our test partition (in the case of LSTM usage) and for the entire datasets (in cases they are not used for training).

Evaluation of our system is carried out for the entire system as well as for the study of LSTM network alone. For the case of the entire system, we consider the metrics used

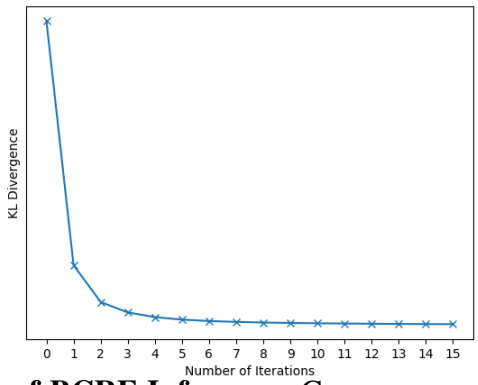

Figure 3.1: **Convergence of BCRF Inference.** Convergence of KL divergence with the number of iterations.

	PQ		SQ		RQ		
Category	W/O BCRF	BCRF	W/O BCRF	BCRF	W/O BCRF	BCRF	Classes
All	41.4	41.7	78.3	79.1	50.8	51.1	133
Things	47.4	47.4	80.4	80.4	57.3	57.3	80
Stuff	32.5	33.2	75.1	77.1	40.9	41.6	53

Table 3.3: **COCO dataset.** Panoptic segmentation results on the COCO validation set.

by the MOT benchmarks for evaluation. This includes Multiple Object Tracking Accuracy (MOTA), Multiple Object Tracking Precision (MOTP), the ratio of Mostly Tracked targets (MT), and the ratio of Mostly Lost targets (ML). In the case of the LSTM network, the Average Precision (AP) value for the predicted frames across the dataset and classes is reported.

3.1.2 Evaluation

The evaluations on the MOT16 Dataset for the end to end system are reported in Table I. Evaluations mainly focus on two aspects: improvement in accuracy with the introduction of the similarity measure to a traditional tracker using only a Kalman filter or an LSTM network and how closely related the accuracy is with state of the art multi-object trackers. Similar results on the KITTI tracking dataset are presented for our work alongside comparisons (note that few state-of-the-art works report on this dataset) in Table II. Separate evaluations for the LSTM in the case of single object tracking for individual tracklets in the KITTI dataset was carried out. An average IoU of 61.45 and AP of 0.96 at 0.5 IoU were obtained for this experiment.

	PQ		SQ		RQ	
Class	W/O BCRF	BCRF	W/O BCRF	BCRF	W/O BCRF	BCRF
Background	90.8	92.33	93.39	94.69	97.22	97.51
Aeroplane	78.55	80.37	88.57	92.6	88.68	86.79
Bicycle	29.78	31.71	67.36	68.46	44.21	46.32
Bird	84.98	85.09	93.05	93.24	91.32	91.25
Boat	65.83	66.21	85.33	86.48	77.14	76.56
Bottle	67.44	64.05	92.05	90.68	73.26	70.63
Bus	82.68	82.58	94.56	95.46	87.44	86.51
Car	72.22	70.93	93.69	91.7	77.08	77.35
Cat	77.41	83.4	91.24	93.73	84.85	88.97
Chair	43.3	41.79	82.5	82.64	52.49	50.57
Cow	76.91	80.42	92.81	93.95	82.87	85.6
Diningtable	51.33	51.8	80.81	82.88	63.51	62.5
Dog	76.63	81.59	90.5	93.29	84.67	87.46
Horse	76.86	81.4	89.38	91.11	86	89.34
Motorbike	78.07	80.21	87.5	89.89	89.23	89.23
Person	76.33	77	89.75	89.73	85.05	85.81
Pottedplant	58.98	60.62	85.41	85.32	69.06	71.05
Sheep	74.29	74	93.86	93.48	79.15	79.15
Sofa	60.37	62.12	88.47	89.5	68.24	69.41
Train	78.52	80.05	88.7	90.43	88.52	88.52
Tvmonitor	79.23	79.34	92.8	92.93	85.38	85.38
Mean Value	**70.5**	**71.76**	**88.65**	**89.63**	**78.83**	**79.33**

Table 3.4: **Pascal VOC dataset.** Detailed class-wise panoptic segmentation results on the Pascal VOC validation set comparing results without BCRF vs with BCRF on a standard network.

3.2 Panoptic Segmentation Evaluation

In this section, we first show the convergence of the mean field based inference algorithm for BCRF and then show the usefulness of the BCRF model by evaluating its performance on the Pascal VOC dataset and the COCO dataset.

3.2.1 Convergence of the Inference

It is difficult to provide a theoretical convergence guarantee for mean field algorithms with parallel updates [71, 65]. We therefore provide empirical evidence to show that the presented mean field inference algorithm for our BCRF with cross potentials converge under normal conditions. To this end, we estimate the KL divergence between the original joint distribution and the factorized distribution (see Eq. (2.11)), at the end of each iteration in

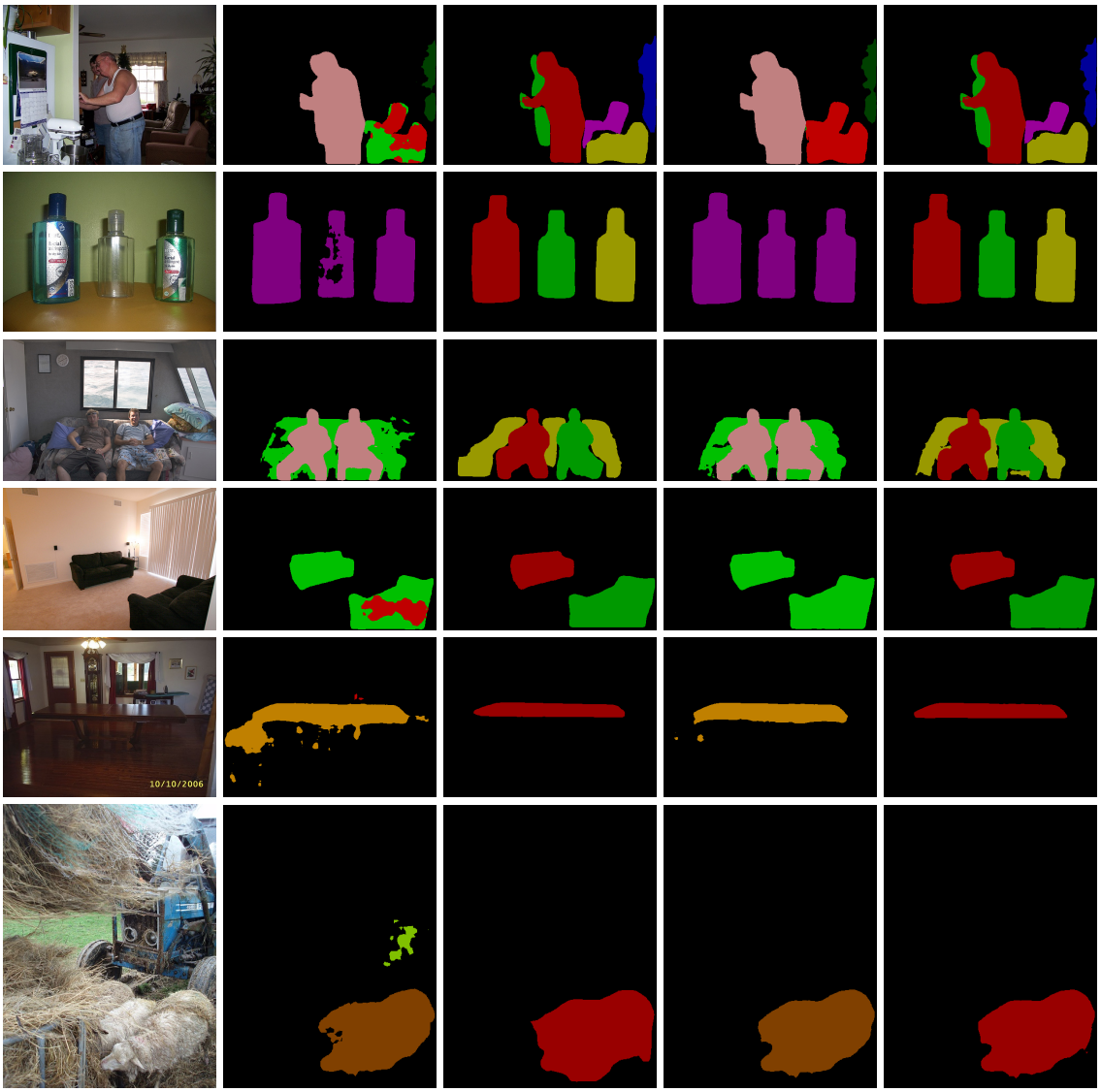

Table 3.5: **Visualizations on Pascal VOC.** Example images from the Pascal VOC validation set. Columns left to right: original image, semantic output before BCRF, instance output before BCRF, semantic output after BCRF, instance output after BCRF. Each row contains a new image. The standard Pascal VOC color map is used for the semantic segmentation results.

Algorithm 1. Note that this KL divergence can be estimated up to a constant using the method described in [72]. We pick 20 random images from the Pascal VOC validation set and average the KL divergence for each iteration across these images. The resulting plot is shown in Fig. 3.1. It can be seen that the KL divergence measure, and therefore the inference algorithm, converges within a few iterations. We also note that visual results do not change after about 5 iterations.

Method	PQ	SQ	RQ
DeeperLab [66]	67.35	-	-
Ours (baseline)	70.50	88.65	78.83
Ours (CRF only)	67.72	87.62	76.48
Ours (BCRF)	71.76	89.63	79.33

Table 3.6: Comparison of results on Pascal VOC dataset. The baseline used contains DeepLab-v3 for semantic branch and Mask-RCNN for instance branch followed by combination using the simple logical method outlined in [1]. CRF only corresponds to setting the BCRF cross-potential terms to zero. BCRF is our complete network.

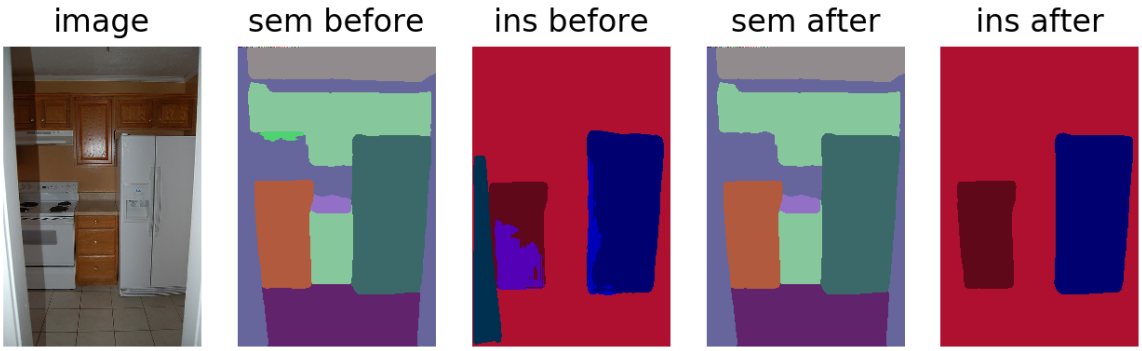

Figure 3.2: Visualisation of improvements on COCO Dataset

3.2.2 Bipartite Potentials Learning

Figure 3.3 illustrates how important logits belonging to each class in the instance branch are for predicting each class in the semantic branch when the model has been fully trained. Our BCRF module allows the network to learn complex relationships between the semantic and instance features belonging to each class. While there is room for it to learn a simple logical relationship, the variation of learned parameters in Figure 3.3 verifies that a complex class-specific mapping has been learned by the network.

3.2.3 Results on the Pascal VOC Dataset

In this experiment we use the architecture shown in Figure 1.2 and CNN components similar to the ones used in [54]. More specifically, we use a ResNet-50 with an FPN as the backend, to which we attach a fully convolutional network as the semantic segmentation head and a Mask R-CNN network as the instance segmentation head.

During both training and inference we used 5 mean-field iterations for BCRF. At the output, we calculate the loss function as a summation of two components: the usual pixel-wise categorical cross entropy loss for the semantic component [26] and the loss used in [48] for the instance component. We used full-image training with batch size 1 and SGD

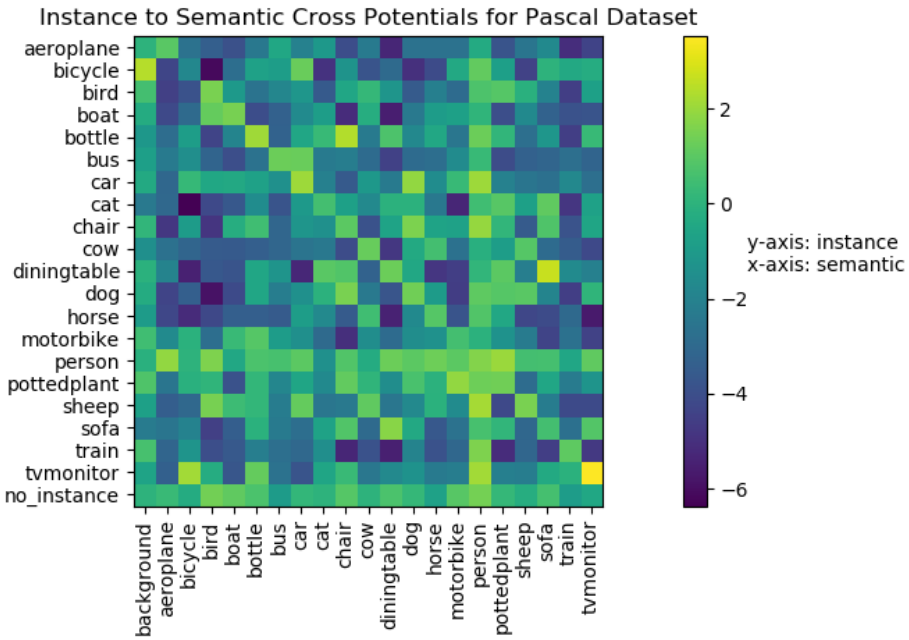

Figure 3.3: The heatmap illustrates inter-class dependencies learned by the cross-potential term weights of BCRF. Note that a logarithmic scale has been used.

with learning rate 0.0007 and momentum 0.99. In Table 3.6, we report the summary of the quantitative results. Table 3.4 shows the class-wise results. Qualitative results are shown in Table 3.5, where benefits of optimally combining the semantic segmentation classification and instance segmentation classification with BCRF can be seen.

3.2.4 Results on the COCO Dataset

To further evaluate the usefulness of BCRF without any efforts for end-to-end training, experiments were conducted on the COCO dataset by simply plugging in the BCRF on an existing pre-trained model. We used a combination of publicly available models of [54, 73], which produced a PQ score of 41.4% on the COCO validation set. The parameters of the BCRF were hand-tuned using a small subset of train images. Results obtained from that BCRF model without end-to-end training are listed in Table 3.3.

4 Discussion and Conclusion

We propose two components essential for autonomous systems that interact with their surrounding environments. These are in fact two of the key computer vision problems that have been attempted for a long time.

Firstly, we present an end-to-end system capable of performing multi-object tracking by combining a range of advances in object detection and reidentification along with our novel architectures and loss functions. Further, we work on a novel step by building a separate LSTM branch to estimate the similarity feature map for the next time step of a given track. The Siamese Networks may be viewed as a two-step version of our extension, whereas this replacement with an LSTM is more of a generalized version capable of generating a better feature set. The key expectation with this addition is the overcoming of identity switches and lost tracks in the case of occlusions. Appearance features tend to change significantly during an occlusion, especially when an object undergoes rotations, and our extension overcomes this by modeling the appearance changing pattern over time.

Thereafter, we proposed a probabilistic graphical model based framework for panoptic segmentation. Our CRF model with two different kinds of random variable, named Bipartite CRF or BCRF, is capable of optimally combining the predictions from a semantic segmentation model and an instance segmentation model to obtain a good panoptic segmentation. We use different energy functions in our BCRF to encourage the spatial, appearance, and instance-to-semantic consistency of the panoptic segmentation. An iterative mean field algorithm was then used to find the panoptic labeling that approximately maximizes the conditional probability of the labeling given the image. We further showed that the proposed BCRF framework can be used as an embedded module within a deep neural network to obtain superior results in panoptic segmentation.

4.1 Principles, Relationships and Generalizations inferred from results

As depicted in the results section our tracker has shown improvements and in relation to MOT evaluation metrics. The improvements shown in the KITTI dataset (which has 9 separate classes) shows how our system has generalized multi class tracking without the need for training separate computationally expensive re-identification networks. MOT16 contains data belonging to the pedestrian class only but the movement of objects in this object is subjugated to more occlusions and random movements compared to the KITTI dataset.

The improvement MOTA of MOT16 dataset indicates signs that our system handles occlusions better.

Since we have only intermediate results for panoptic segmentation for a small test class it can only be inferred that this system has potential to improve panoptic segmentation, but more training and large-scale evaluation is required to state with more certainty.

4.2 Problems and Exceptions to the Generalizations

The results shows that MOTP of our tracker is considerably low in MOT16 dataset in comparison to other systems. This indicates that the LSTM network is unable to handle rapid variations of the bounding box parameters. This is to be expected as the bounding box variations in datasets such as MOT16 is much more chaotic in cases where the pedestrian is rotating while walking and walking in general.

4.3 Agreements/Disagreements with previously published work

The results agree with recently published systems such as Deep Sort []. It is expected that as ML decreases the MOTA to increase as it reduces the number of false negatives considerably. This correlation is depicted in our results.

References

[1] A. Kirillov, K. He, R. B. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *CVPR*, 2019.

[2] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realLong Short-term Memorytime tracking with a deep association metric," in *IEEE International Conference on Image Processing (ICIP)*, 2017.

[3] Z. Qin, J. Wang, and Y. Lu, "Triangulation Learning Network: from Monocular to Stereo 3D Object Detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[4] T. Bruls, HoriaPorav, L. Kunze, and P. Newman, "The Right (Angled) Perspective: Improving the Understanding of Road Scenes Using Boosted Inverse Perspective Mapping," in *IEEE Intelligent Vehicles Symposium*, 2019.

[5] H. S. and S. J., "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE TPAMI*, 2017.

[7] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs," *PAMI*, 2018.

[8] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *ICCV*, 2017.

[9] F. Yu and V. Koltun, "Multi-Scale Context Aggregation by Dilated Convolutions," in *ICLR*, 2016.

[10] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[11] R. B. Girshick, "Fast R-CNN," in *International Conference on Computer Vision (ICCV)*, 2015.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, and P. H. S. Torr, "Fully-convolutional siamese networks for object tracking," in *European Conference on Computer Vision (ECCV)*, 2016.

[14] L. Z. Qiang Wang, L. Bertinetto, W. Hu, and P. H. S. Torr, "Fast online object tracking and segmentation: A unifying approach," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[15] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 FPS with deep regression networks," in *European Conference on Computer Vision (ECCV)*, 2016.

[16] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Detect to track and track to detect," in *International Conference on Computer Vision (ICCV)*, 2017.

[17] M. Liu and M. Zhu, "Mobile video object detection with temporally-aware feature maps," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

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

[19] A. Milan, S. H. Rezatofighi, A. Dick, I. Reid, and K. Schindler, "Online multi-target tracking using recurrent neural networks," in *Conference on Artificial Intelligence, Association for the Advancement of Artificial Intelligence (AAAI)*, 2016.

[20] H. Morimitsu, "Multiple Context Features in Siamese Networks for Visual Object Tracking," in *European Conference on Computer Vision (ECCV)*, 2018.

[21] W. Luo, B. Yang, and R. Urtasun, "Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[22] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation," in *European Conference on Computer Vision (ECCV)*, 2006.

[23] C. Liu, J. Yuen, and A. Torralba, "Sift flow: Dense correspondence across scenes and its applications." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, pp. 978–994, 2011.

[24] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille, "The role of context for object detection and semantic segmentation in the wild," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14, 2014, pp. 891–898.

[25] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, pp. 98–136, 2014.

[26] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," in *CVPR*, 2015.

[27] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 2481–2495, 2017.

[28] L.-C. Chen, Y. Yang, J. Wang, W. Xu, and A. Yuille, "Attention to scale: Scale-aware semantic image segmentation," in *CVPR*, 06 2016, pp. 3640–3649.

[29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6230–6239, 2016.

[30] F. Yu, V. Koltun, and T. A. Funkhouser, "Dilated residual networks," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 636–644, 2017.

[31] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, "Conditional Random Fields as Recurrent Neural Networks," in *ICCV*, 2015.

[32] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected crfs," *CoRR*, vol. abs/1412.7062, 2014.

[33] A. Arnab, S. Jayasumana, S. Zheng, and P. Torr, "Higher Order Conditional Random Fields in Deep Neural Networks," in *ECCV*, 2016.

[34] Z. Liu, X. Li, P. Luo, C.-C. Loy, and X. Tang, "Semantic image segmentation via deep parsing network," in *The IEEE International Conference on Computer Vision (ICCV)*, December 2015.

[35] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *ArXiv*, vol. abs/1706.05587, 2017.

[36] G. Lin, A. Milan, C. Shen, and I. Reid, "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation," in *CVPR*, 07 2017, pp. 5168–5177.

[37] 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 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 91–99.

[38] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *ECCV*, 2016.

[39] Y. Yang, S. Hallman, D. Ramanan, and C. C. Fowlkes, "Layered object models for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, pp. 1731–1743, 2012.

[40] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[41] X. He and S. Gould, "An exemplar-based crf for multi-instance object segmentation," in *CVPR*, 2014, pp. 296–303.

[42] J. Tighe, M. Niethammer, and S. Lazebnik, "Scene parsing with object instances and occlusion ordering," in *IEEE Conference on Computer Vision and Pattern Recognition*, 09 2014, pp. 3748–3755.

[43] J. Dai, K. He, and J. Sun, "Convolutional feature masking for joint object and stuff segmentation," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3992–4000, 2015.

[44] J. R. R. Uijlings, K. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *International Journal of Computer Vision*, vol. 104, pp. 154–171, 2013.

[45] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, "Simultaneous detection and segmentation," in *ECCV*, 2014.

[46] Y. Li, H. Qi, J. Dai, X. Ji, and Y. Wei, "Fully convolutional instance-aware semantic segmentation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 4438–4446.

[47] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia, "Path aggregation network for instance segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8759–8768, 2018.

[48] A. Arnab and P. H. Torr, "Pixelwise Instance Segmentation with a Dynamically Instantiated Network," in *CVPR*, 2017.

[49] H. Liu, C. Peng, C. Yu, J. Wang, X. Liu, G. Yu, and W. Jiang, "An end-to-end network for panoptic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[50] T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L.-C. Chen, "Deeperlab: Single-shot image parser," *ArXiv*, vol. abs/1902.05093, 2019.

[51] N. Gao, Y. Shan, Y. Wang, X. Zhao, Y. Yu, M. Yang, and K. Huang, "Ssap: Single-shot instance segmentation with affinity pyramid," *ArXiv*, 09 2019.

[52] Y. Li, X. Chen, Z. Zhu, L. Xie, G. Huang, D. Du, and X. Wang, "Attention-guided unified network for panoptic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[53] L. Porzi, S. Rota Bulò, A. Colovic, and P. Kontschieder, "Seamless scene segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[54] Y. Xiong, R. Liao, H. Zhao, R. Hu, M. Bai, E. Yumer, and R. Urtasun, "Upsnet: A unified panoptic segmentation network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[55] A. Kirillov, R. Girshick, K. He, and P. Dollar, "Panoptic feature pyramid networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[56] A. Kirillov, R. B. Girshick, K. He, and P. Dollár, "Panoptic feature pyramid networks," in *CVPR*, 2019.

[57] P. Krähenbühl and V. Koltun, "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials," in *NIPS*, 2011.

[58] A. Arnab, S. Jayasumana, S. Zheng, and P. H. S. Torr, "Higher order conditional random fields in deep neural networks," in *European Conference on Computer Vision (ECCV)*, 2016.

[59] Q. Li, A. Arnab, and P. H. S. Torr, "Weakly- and semi-supervised panoptic segmentation," in *European Conference on Computer Vision (ECCV)*, 2018.

[60] A. Milan, L. Leal-Taixé, I. D. Reid, S. Roth, and K. Schindler, "MOT16: A benchmark for multi-object tracking," *CoRR*, vol. abs/1603.00831, 2016. [Online]. Available: http://arxiv.org/abs/1603.00831

[61] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[62] J. Valmadre, L. Bertinetto, J. Henriques, A. Vedaldi, and P. H. S. Torr, "End-to-end representation learning for correlation filter based tracking," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, p. 5000–5008.

[63] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval Research Logistics (NRL)*, 1955.

[64] J. F. Arsalan Mousavian, Dragomir Anguelov, "3D Bounding Box Estimation Using Deep Learning and Geometry," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[65] Koller, Daphne and Friedman, Nir, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. The MIT Press, 2009.

[66] T.-J. Yang, M. D. Collins, Y. Zhu, J.-J. Hwang, T. Liu, X. Zhang, V. Sze, G. Papandreou, and L.-C. Chen, "Deeperlab: Single-shot image parser," *ArXiv*, vol. abs/1902.05093, 2019.

[67] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *International Conference on Computer Vision (ICCV)*, 2015, p. 4705–4713.

[68] J. Zhu, H. Yang, N. Liu, M. Kim, W. Zhang, and M.-H. Yang, "Online multi-object tracking with dual matching attention networks," *Computer Vision – ECCV*, vol. 11209, p. 379–396, 2018.

[69] J. Kuck and P. Zhuang, *Target tracking with kalman filtering*, 2016.

[70] M. Kim, S. Alletto, and L. Rigazio, "Similarity mapping with enhanced siamese network for multi-object tracking," in *Machine Learning for Intelligent Transportation Systems (MLITS)*, 2016.

[71] P. H. T. Vibhav Vineet, Jonathan Warrell, "Filter-based Mean-Field Inference for Random Fields with Higher-Order Terms and Product Label-Spaces," in *ECCV*, 2012.

[72] P. Krähenbühl and V. Koltun, "Efficient Inference in Fully Connected CRFs with Gaussian Edge Potentials - Supplementary Material," in *NIPS*, 2011.

[73] J. Huang, V. Rathod, C. Sun, M. Zhu, A. K. Balan, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, "Speed/accuracy trade-offs for modern convolutional object detectors," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3296–3297, 2016.