RefineNet: Refining Object Detectors for Autonomous Driving

Eshed Ohn-Bar, Rakesh Nattoji Rajaram, and Mohan Manubhai Trivedi
Laboratory for Intelligent and Safe Automobiles
University of California San Diego
{eohnbar, rnattoji, mtrivedi}@ucsd.edu

***Abstract*—Highly accurate, camera-based object detection is an essential component of autonomous navigation and assistive technologies. In particular for on-road applications, localization quality of objects in the image plane is important for accurate distance estimation, safe trajectory prediction, and motion planning. In this paper, we mathematically formulate and study a strategy for improving object localization with a deep convolutional neural network. An iterative region-of-interest pooling framework is proposed for predicting increasingly tight object boxes and addressing limitations in current state-of-the-art deep detection models. The method is shown to significantly improve performance on a variety of datasets, scene settings, and camera perspectives, producing high quality object boxes at a minor additional computational expense. Specifically, the architecture achieves impressive gains in performance (up to 6% improvement in detection accuracy) at fast run-time speed (0.22 seconds per frame on 1242×375 sized images). The iterative refinement is shown to impact subsequent vision tasks, such as object tracking in the image plane and in ground plane.**

***Index Terms*—Object detection, convolutional networks, proposal refinement, fast detection, autonomous driving, vehicle detection and tracking, surround behavior analysis, multi-perspective vision.**

I. INTRODUCTION

Object detection from a camera is a long studied problem in computer vision and intelligent vehicles [1], [2]. For on-road, safety-critical applications, accurate localization is key as it allows understanding of the surround for planning around obstacles. Recent progress in vision-based object detection technologies have significantly advanced state-of-the-art, but several issues are still left unresolved [3]. Specifically for on-road settings, there is a need to not just robustly detect objects under a diversity of settings, including variable occlusion, size, truncation, illumination, orientation and scene complexity, but also *accurately localize* them with a high degree of accuracy. Furthermore, computational resources and run-time speed play a critical role for many applications, including autonomous driving. To that end, we propose and analyze the role of a refinement module for region-based object detection models. The module results in significantly better object localization without any modification to the training of the detector, and little impact on the computational cost during testing.

Consider a scenario where an autonomous vehicle has to maneuver around other on-road occupants (i.e. vehicles, pedestrians, cyclists, etc.), as in Fig. 1. This task involves understanding of the 3D world around the vehicle, detecting the boundaries of objects (to avoid collision), and predicting surround agent behavior. Hence, given an image of the scene, a critical vision task is to detect and accurately localize objects.

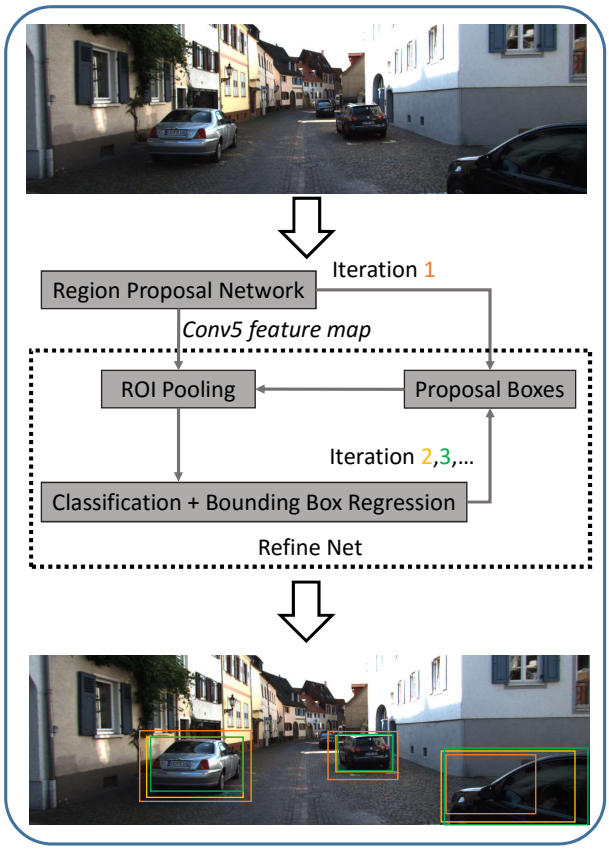


Fig. 1. This paper studies an iterative refinement process (**RefineNet**) in order to improve the quality of a deep learning-based object detector. The technique makes use of the already extracted CNN features and improves the localization accuracy of the detection boxes at a marginal increase in computation cost. In the image, orange, yellow and green color represents bounding boxes at iterations 1, 2 and 3 respectively.

The reason for this is twofold. First, missing a part of a pedestrian or a vehicle could be a matter of life and death. Second, non-tight boundaries could result in sub-optimal performance in subsequent tasks, including object segmentation, classification, 3D localization, orientation estimation, object tracking, surround behavior recognition, and up to planning and decision making. Therefore, the task of localization is of high importance, in particular for the intelligent vehicle. Furthermore, achieving it at low computational cost is desirable. We formulate localization within an iterative framework, such that detection boxes become increasingly until convergence. By using the proposed method, we are able to show significant detection and localization improvement for a vehicle detection task in a variety of driving settings. Because the developed

approach is general, it is applicable to many state-of-the-art object detectors. Furthermore, it comes at no additional training time cost or memory cost, and minimal impact on test-time speed.

Visual image analysis often incorporates a deep convolution neural network (CNN), whether for classification [4], [5], [6], object detection [7], [8], [9], [10], and semantic segmentation [11], [12], [13], [14]. Specifically for object detection, state-of-the-art approaches employ an attention mechanism in the form of a region proposal step, as proposed in R-CNN [7] (Regions CNN). The features within the regions are then classified into an object class, and regressed for encompassing bounding box parameters. Subsequent innovations [8], [9] to the R-CNN framework worked to join its independent modules into a joint, end-to-end framework, and decreasing its run-time speed. In this work, we employ a fast detection network and achieve significant improvement by performing iterative bounding box refinement (as shown in Fig. 1). The studied approach is mathematically formulated on top of R-CNN, and is complementary to most improvements introduced into R-CNN in state-of-the-art detectors (including improved proposals [15], deeper networks [5], or better multi-scale handling [16]). Each iteration in the proposed iterative localization framework provides the region of interest (ROI) pooling layer a region closer to the ground truth object to pool features from. This improves classification and localization accuracy, while also providing an interesting framework in which to analyze the R-CNN technique and its shortfalls. We also analyze the iterative refinement framework which we term **RefineNet** extensively, from hyper-parameter settings and convergence and up to generalization across datasets. Specifically, the contributions presented in this paper are as follows.

A. Contributions

- **Localization refinement framework**: We develop a general detection framework which provides iterative refinement of the output of a deep detection network. The general insight that better localized regions leads to better bounding box regression (preliminarily presented by us in [17]) is analyzed on two types of driving settings, urban European (KITTI [18]) and a multi-view highway dataset [19], [20], showing significant impact on performance.
- **Mathematical motivation**: The mathematics of region proposal-based detection methods naturally motivates an iterative refinement module which is utilized in this work. This general idea can be incorporated into any detection network, with Faster R-CNN [9] used in this work. Under an iterative framework viewpoint, analysis of convergence is interesting (shown to occur within 3 iterations). While most state-of-the-art deep object detection frameworks employ Fast R-CNN [8] or Faster R-CNN [9] with either better proposals [15], deeper network designs [5], or novel loss functions [21], the idea of iterative refinement using a fixed network structure for better performance has not been studied in related research.
- **Experimental analysis**: A set of novel experiments not performed in [17] demonstrates generalization across settings, datasets, and camera perspectives. Furthermore, we analyze for perspective sensitivity, impact of hyper-parameters (such as number of object proposals) on performance and run-time speed, and 2D/3D tracking and localization. Up to 6% improvement in detection accuracy is observed on the challenging KITTI benchmark [18] (German/Urban driving settings) and a multi-perspective US highway dataset captured by our lab [19], [20]. As accurate localization is essential for better understanding of surround activities [22], [19] and safe trajectory planning in on-road settings [23], refinement is especially crucial for camera-based on-road, autonomous driving settings.
- **Run-time speed**: RefineNet allows employing a network with less parameters but still achieve high accuracy. For instance, on the KITTI object detection benchmark [18], RefineNet with a smaller network (ZF Net [24]) achieves comparable results in terms of detection performance to using the Faster R-CNN baseline, but with a bigger network (VGG16 [5]), while running nearly an *order of magnitude faster* than Faster R-CNN with VGG16 making it one of the fastest detectors on the benchmark.

II. Related Research Studies

Recent progress in object detection can be attributed to the ability of deep convolutional neural network to learn discriminative features across wide variation in object appearance. In [25], bounding box for object is treated as a regression problem on top of prefixed object masks. R-CNN [7] first minimizes the search space from millions of windows to a few thousand probable windows (using selective search [26]) and then extracts CNN features from each window using a model that is fine tuned on a particular dataset. This high dimensional feature is then passed on to a support vector machine for classification and regression to correct the bounding box. Fast R-CNN [8] builds on top of R-CNN to improve the computational efficiency by introducing ROI pooling layer to extract features by sampling from this layer. Faster R-CNN [9] further improves the computational efficiency by introducing a proposal regression layer to perform object detection with a single pass. 3DOP [15] provides a new proposal generation mechanism using depth information which when used along with Fast-RCNN produces state-of-the art results. Depth can also be used as a cue in the detection model, as in [27]. MSS [28] employs an improved localization model over multi-scale conv5 features. Detection models are learned for each image scale to better capture object variation due to scale. SDP [16] extends the Fast R-CNN idea by introducing ROI pooling layer at multiple *conv* layers to improve detection of small objects and also apply the cascaded rejection classifier technique to quickly reject proposals with low confidence.

Prior to CNN making headway into object detection, Deformable Parts Model (DPM) [29] was the gold standard for years. The key idea introduced in DPM was to formulate an object as a root template with a fixed number of associated parts whose position is flexible relative to the root template. Similarly, Regionlets [30] introduces appearance flexibility but in the feature space. It operates by minimizing the search

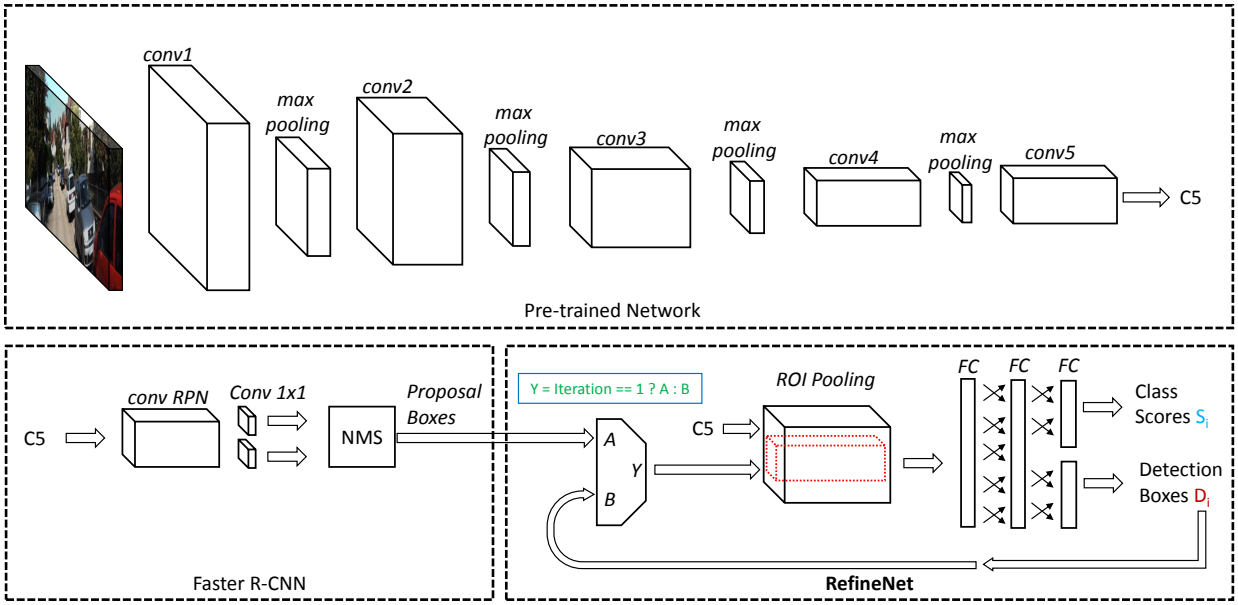


Fig. 2. An overview of the approach studied in this paper. A pretrained network which is fine-tuned on a object detection dataset is used to extract convolutional feature map (C_5). Using these features, proposal boxes are generated with Faster R-CNN framework, followed by classification and bounding box regression (iteration 1, generating detection boxes D_1 and detection scores S_1). Successive iterations i involves refining the detection boxes D_i by using detections from previous iterations i.e. D_{i-1} as proposal boxes for constructing the ROI pooled features.

space to a few thousand windows (using selective search [26]), extracting features from a fixed number of regions inside these windows, and then pooling them to establish invariance to localization, scale and aspect ratio. Next, the detected objects are re-localized using a localization model. SubCat [31] introduces modifications on top of the detector. Here, objects are sub-categorized into a fixed number of clusters based on geometric features such as height, width, aspect ratio, occlusion etc. and other image features. Then, a separate model is trained for each of these clusters. Along with improving detector accuracy, SubCat also estimated vehicle orientation.

It is possible to draw some similarities between RefineNet and recurrent neural network (RNN) or auto-context work such as [32], [33], [34]. While RefineNet generalizes R-CNN as a first stage in an iterative framework without recurrence and for localization purposes, the aforementioned approaches do not iterate pooling of the CNN features in each ROI and do not study such as framework for improved object localization for on-road data.

III. RefineNet

Most related research studies improve upon R-CNN by optimizing one of its modules, from region proposals, to the type of network used. On the other hand, the proposed iterative framework *generalizes R-CNN* into a framework which reveals more about the lacking of the components in R-CNN. Specifically, the ROI pooling layer, and the bounding box classification and regression modules (shown in Fig. 2) are sensitive to the original proposed ROI. By iterating over ROI pooling and box regression, we provide the system with a mechanism to increasingly correct itself and any shortcomings in the sub-modules. Hence, the approach is named RefineNet. Although the idea is general, in this work we employ the current state-of-the-art object detector of Faster R-CNN [9] to demonstrate performance gains.

In supervised CNN frameworks, the objective is to train a network F that predicts an output $\hat{y}$, given an input x (i.e. an image),

$$\hat{y} = F(\mathbf{x}) \tag{1}$$

In this notation, F is an embedding of all of the parameters and operations of the network layers.

Most modern detectors employ Fast R-CNN [8]. In this framework, convolutional feature maps are first extracted from a given image. If the image dimensions are $H \times W$, the method employes a CNN (ZF [24], AlexNet [4], or VGG16 [5]) to extract convolutional features (*conv5*) of dimensions $\lfloor \frac{H}{16} \rfloor \times \lfloor \frac{W}{16} \rfloor \times 256$. Next, proposal boxes are generated and projected to this convolutional feature space for re-sampling to a fixed size ($6 \times 6 \times 256$ for ZF network). This is followed by further pooling, 3 cascaded fully connected layers, and final bounding box regression and class score. Faster R-CNN [9] introduces a region proposal network (RPN) for a unified, end-to-end network for object detection. Before the introduction of the RPN, the region proposal mechanism was kept as a separate module during training and testing. The RPN in [9] employs the *conv5* features and applies a filter of size 3×3 followed by two 1×1 convolutional filters for generating proposal boxes and objectness scores at each spatial location. At each spatial location, multiple proposals can be generated using *anchor* boxes. These anchor boxes can be setup at multiple scales and aspect ratio and serves as reference for regression.

Regardless of the exact region proposal mechanism, a key insight in the R-CNN-based frameworks is the additional ROI parameters, D, so that the prediction function becomes

$$\hat{y} = F(\mathbf{x}, D) \tag{2}$$

We note that the label space is now both a class prediction and a $4D$ bounding box, $\hat{y} = \{\hat{y}^c, \hat{y}^D\}$. The parameter $D \in R^{M\times 4}$ specifies M boxes in the image plane and is introduced for detection and localization applications (as opposed to the classification only case in Eqn. 1). Although current state-of-the-art object detectors all employ a region proposal and pooling mechanism, several potential questions have not been well studied in literature, in particular the impact of a poorly localized D on the output of the fully connected layers and output quality. Furthermore, y^D is obtained using a bounding-box regression module, and its ability to recover from poorly localized regions D also requires analysis.

Motivated by such potential issues, we introduce a generalization of R-CNN with an iterative framework. Since $\hat{y}^D$ has undergone bounding-box regression, y^D is generally better localized then the input proposal ROIs, D. We then re-feed y^D to analyze for further gains, and define $\hat{y}_2 = F(\mathbf{x}, y^D) := F(\mathbf{x}, D_1)$. In general, the process can be applied iteratively,

$$\hat{y}_{N+1} = F(\mathbf{x}, D_N) \tag{3}$$

where we note that the $N = 1$ case is the baseline R-CNN technique. Throughout the iterations, the D parameter changes from the RPN output in $N = 1$, and previous R-CNN regression outputs at $N > 1$, until the regression module provides no additional refinement benefit. This formulation allows us to analyze the properties of the bounding box regression module in F.

A benefit of the proposed approach is that its general nature allows us to study it with any region-based object detection method. In this work we utilize the state-of-the-art Faster R-CNN detection scheme. RefineNet follows the training scheme of the underlying detection scheme, but iterates over the ROI-pooling, fully-connected, and output layers in test time. First, a pass-forward through the network generates the *conv5* feature activations which are stored in memory. The RPN in Faster R-CNN generates detection boxes, D_1. Successive iterations i use the same *conv5* features, but detection boxes from previous iterations i.e D_{i-1} as proposal boxes input to the ROI pooling stage of Fast R-CNN. Throughout this process, we find that the overlap with the ground truth target boxes increases, so that features obtained by the ROI-pooling layer and fully connected layers become more representative of the true object class and its location. Hence, not just localization gets improved, but also the class scoring. RefineNet allows for recursively improving the classification score and also the localization accuracy. As will be shown in the next section, this process results in significant improvement on a variety of dataset settings and camera perspectives, and is crucial for applications requiring high localization accuracy of objects (as opposed to generic object detection which is often the settings in which these networks are tested). Furthermore, the formulation provides insights into possible shortfalls in the R-CNN architecture, and propose to resolve it by iterative refinement. The refinement can also be though of as an attention mechanism which allows the network to better handle challenging cases.

During training, we follow [9] to first train the RPN network initialized with model pre-trained on ImageNet [35] dataset. On KITTI we employ ignore regions during the training of the networks. Specifically, an anchor box generated using the RPN network is ignored if it overlaps (intersection over union-IoU) by more than 0.6 with an ignore region. Foreground boxes are required to have atleast 0.5 IoU overlap with a ground truth box, and an IoU of less than 0.3 for background boxes. As the main modification is the test time refinement procedure, we employ the standard multi-task loss function defined over a classification loss (of the object classes or background) and regression loss. The tasks are learned jointly, as in Fast R-CNN [8].

IV. Experimental Setup

Dataset: RefineNet and its parameter settings are evaluated on two datasets, the KITTI object detection benchmark [18] and a US highway dataset collected using a four perspective setup collected in our lab [19], [20]. On KITTI, we follow the training/validation split of [36], resulting in 3682/3799 images respectively. As augmentation, instances are horizontally flipped which leads to a small improvement in performance. KITTI object detection benchmark evaluates the detector performance at 3 different difficulty settings, varying by object properties. Specifically, "easy" test settings employ objects of height greater than 40 pixels, no occlusion, and small truncation (up to 15%). "Moderate" difficulty employs a height of 25 pixels, partial occlusion, and up to 30% truncation. "Hard" difficulty adds upon to "moderate" to include objects with high occlusion and high truncation (up to 50%). For analysis on KITTI, we employ the 'car' object class, but detection of other object types (e.g. pedestrians) is also expected to benefit from the proposed approach [37]. All models are trained on moderate difficult settings, as suggested by the KITTI benchmark [38]. A similar experimental setup is created on the US highway dataset, captured using four synchronized GoPro cameras (synchronized captured at a resolution of 2704×1440, at 12 Hz). The main objective is to analyze generalization and potential overfitting to the settings or perspective [39]. The panoramic camera array dataset is designed to analyze surround vehicles and their behavior. Hence, on this video dataset, we will demonstrate performance improvement on a variety of tasks related to camera-based object recognition, tracking, and behavior analysis. The dataset contains 400 frames in each view (a total of 1600 frames) and over 4000 vehicles. The instances have also been annotated with occlusion and truncation state to analyze the performance gains of the RefineNet method against the baseline in a new camera and scene settings. For the detection tasks, a precision-recall curve is obtained and the area under the curve (AUC) is used as a performance measure. As we are concerned with localization quality, we will be varying the IoU overlap threshold, o_{th}, required for a true positive detection. This type of analysis quantifies localization improvement due to different choices in the RefineNet approach (hyper parameters and number of iterations).

Training details: There are a few modifications needed in order to achieve good performance on KITTI with state-of-

the-art detection networks. The most important one involves training and testing in multiple scales, as scale variation is a frequent challenge in on-road settings. As a result, most state-of-the-art CNN detectors employ an image pyramid. For instance, in [21], input image is up-sampled by $4\times$ and in [15] by $3\times$. Upsampling of the input image helps deal with network architectures which have a stride of more than 1, thereby losing fine-grained or small object detail, as well as handle the reduction in resolution due to pooling layers. These modifications are necessary when going from a general classification or detection task on ImageNet [35], to on-road settings such as KITTI or highway driving. The following parameters are used for the 4-step alternate training using stochastic gradient descent with momentum. For RPN training, we set the batch size to 256 instances, and train for $80,000$ iterations, with a base learning rate of 0.001, step size of $60,000$, learning rate scale factor of 0.1, momentum of 0.9, and weight decay of 0.0001. For Fast R-CNN training, most of the parameters are kept fixed, besides that the batch size is set to 128 instances, and training is done for 40,000 total iterations.

Analysis parameters: Throughout the experiments the main parameters we will vary are detailed below. As most networks are trained with a fixed scale of 224×224 on ImageNet, it is unable to deal with representing objects with large scale variation as they appear on the road. We refer to the scale of the shortest side of the image as s, and show results of RefineNet using different settings of s. During training, each ground truth is assigned to the closest scale. During testing, only the top K_2 proposals are selected after passing the top K_1 proposals through a non maximal suppression (NMS) unit (IoU threshold: 0.7). We note that multi-scale detection occurs at each scale independently, and the results are later joined across scales with another NMS (IoU threshold: 0.3) to remove duplicate detection boxes. To analyze localization, we also vary the overlap threshold required for a true positive detection, o_{th}. We note that iterative refinement and its impact with these parameters have not been studied in related research studies. In order to further understand the sensitivity of the model settings to iterative refinement, two types of models are compared, M_1 and M_2. The models vary in terms of the analysis parameter settings, as detailed in the next section.

V. Experimental Analysis

Our initial experiment employs a RefineNet model which we refer to as M_1. It is trained with a ZF network on KITTI with the scales parameter $s = \{375, 750\}$ for multi-scale training and testing. This implies training and testing in the original image scale, as well as twice the original image scale. We follow Faster R-CNN with 9 anchors at 3 different scales (8,16 and 32) and 3 different aspect ratios (1:1, 1:2 and 2:1). For the first iteration, we use the $K_1 = 6000$ and $K_2 = 300$ boxes into the Fast R-CNN network. In Table I, we report AUC as a function of overlap threshold o_{th} and number of refinement iterations. As the o_{th} increases, we observe more significant improvement due to the refinement iterations. For instance, when o_{th} is set to 0.7 (the KITTI default), we observe an improvement from 78.78% up to 81.26% in only one additional refinement iteration. This significant improvement of 2.48% especially at this high overlap requirement demonstrates the usefulness of iterative refinement. This aspect of performance improvement is highly critical to camera-based vision for autonomous driving, as it will impact subsequent tasks (including 3D tracking, as will be discussed later). Adding another refinement iteration often further improves performance, especially when the overlap requirement is high, but often convergence occurs at $N = 2$ or $N = 3$. The performance are impressive considering that with iterative refinement the performance nearly matches detection with the much bigger and more computationally intense network, VGG16. At the same time, RefineNet with ZF runs significantly faster than VGG16 by nearly an order of magnitude.

RefineNet provides highly localized detection, with no additional cost during training. Using a smaller network provides noisier prediction output, yet this is often shown to be resolved by iterative refinement. Next, we would like to analyze the limits of RefineNet by varying the number of proposals (a main determining factors in the run-time of R-CNN-based frameworks), and see how well can it still correctly resolve all ground truth objects. The results are shown in Table II, where we set $K_1 = 1000$ and $o_{th} = 0.7$. We can see how the improvement of the iterative refinement step is consistent regardless of the number of proposals used, K_2. Comparing to the results in Table I, we observe very small reduction in AUC due to using smaller values of K_2, but the impact on run-time is large. Specifically at $N = 3$, run-time reduces from 0.29 seconds per image to 0.22 seconds per image with $K_2 = 200$, while nearly achieving the same performance.

Another main parameter to study when performing localization experiments is the number of anchor boxes in the RPN. Specifically, we train a RefineNet model (M_2) with just one anchor box (a square with sides of length 67 pixels and centered at 0,0). The experiment is meant to measure how well can RefineNet resolve boxes which are poorly localized (for further gains in speed). In this experiment, training and testing is carried out at scales $s = \{375, 750\}$ as before. In Table III, we report accuracy and runtime as a function of K_2. $K_1 = 1000$ as in previous experiments. At $K_2 = 200$, runtime reduces to **0.20** seconds with less than 0.9% decrease in AUC. Although, the decrease in runtime is not significant, the improvement in AUC from 74.54% to 80.69% is more than **6%**. An important observation here is that decreasing the number of anchor boxes from 9 in M_1 to just 1 in M_2 reduces the number of model parameters, making the model more light-weight. Computational efficiency and memory are important aspects of on-road vision-based techniques. While the first iteration (the baseline Faster R-CNN model) significantly suffers from this reduction (AUC drops from 78.79% vs 74.54%), RefineNet is able to regain most of the loss in performance within one or two refinement iterations.

Comparing VGG16 and ZF: The ZF Net [24]) offers

[1]Using Nvidia GTX Titan X

TABLE I
AUC AT DIFFERENT OVERLAP THRESHOLD (o_{th}) AND NUMBER OF REFINEMENT ITERATIONS (N). METRICS GENERATED ON KITTI VALIDATION SET USING THE REFINENET MODEL M_1.

o_{th}	N=1	N=2	N=3	N=4	N=5
0.60	90.92	**91.97**	91.52	91.30	91.23
0.65	86.48	**88.34**	87.76	87.67	87.61
0.70	78.78	81.26	**81.58**	81.15	80.73
0.75	65.10	**69.63**	69.28	69.59	69.20
0.80	43.63	50.06	**51.61**	51.21	50.86
Runtime[1] (sec)	0.20	0.24	0.29	0.34	0.38

TABLE II
AUC AT DIFFERENT NUMBER OF INPUT PROPOSALS (K_2) AND NUMBER OF REFINEMENT ITERATIONS (N). METRICS GENERATED ON KITTI VALIDATION SET USING THE REFINENET MODEL M_1.

K_2	N=1	N=2	N=3	N=4	N=5	Runtime(sec) @N=3
200	78.79	81.07	**81.25**	80.86	80.54	0.22
100	78.83	81.13	81.15	81.02	80.44	0.20
50	78.16	80.66	80.79	80.46	80.00	0.16
25	76.77	78.99	79.06	78.97	78.35	0.15

TABLE III
AUC AT DIFFERENT NUMBER OF INPUT PROPOSALS (K_2) AND NUMBER OF REFINEMENT ITERATIONS (N). METRICS GENERATED ON KITTI VALIDATION SET USING THE REFINENET MODEL M_2.

K_2	N=1	N=2	N=3	N=4	N=5	Runtime(sec) @N=3
200	74.54	80.03	**80.69**	80.13	78.83	0.20
100	74.79	80.02	80.37	79.41	78.28	0.18

fast training and testing, and so we prefer it for intelligent vehicles applications and prototyping of new ideas. Nonetheless, we would like to compare results with the state-of-the-art VGG16 [5] network which is larger and significantly more computationally intensive. Furthermore, we would like to see if RefineNet can generalize to other network architectures beyond ZF Net. For the VGG16 network with the same settings as the previous experiments and an overlap threshold of $o_{th} = 0.70$, the AUC for $N = 1, 2, 3, 4$ is $82.20, 83.87, 83.27, 83.35$, respectively. Hence, we observe iterative refinement to improve VGG16 output as well. More surprisingly, RefineNet with a much smaller ZF network nearly matches the VGG16 baseline in performance (81.58% vs. 82.20). To emphasize, the larger VGG16 network runs at a run-time of nearly an order of magnitude slower (a factor of $\times 9$). Further gains in detection performance of up to 83.87 result from employing RefineNet.

Evaluation on KITTI test set: Our main emphasis in this paper has been analyzing a novel iterative framework when deep CNN object detectors are concerned. In the process, we underlined some of the limitations in current state-of-the-art R-CNN-based detectors, mostly with sensitivity to the proposal boxes. We also highlight fast run-time which is more appropriate to the intelligent vehicles domain. As a final experiment on KITTI, we perform a comparative discussion by submitting results to the KITTI evaluation server. We train a RefineNet model with parameters taken from M_1. This model achieves

TABLE IV
AUC ACHIEVED BY STATE-OF-THE-ART DETECTORS ON THE KITTI OBJECT DETECTION BENCHMARK. ASTERISK (*)- METHODS EMPLOY THE VGG [5] NETWORK AS OPPOSED TO THE ZF NETWORK USED IN THIS WORK.

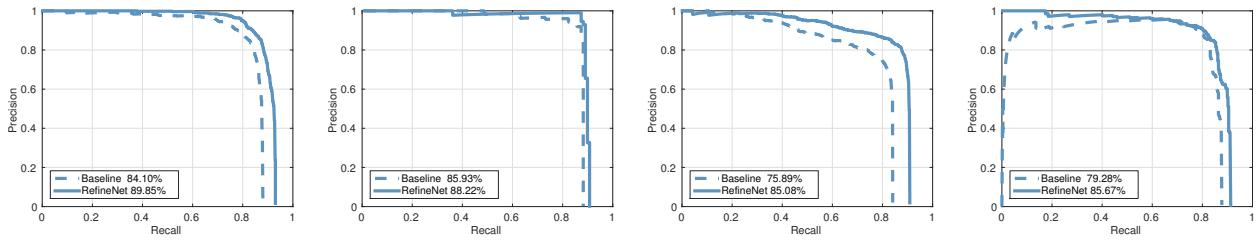
Detector	AUC			Runtime(sec)
	Easy	Moderate	Hard	
3DOP* [15]	**93.04**	88.64	79.10	3
SubCNN* [21]	90.81	**89.04**	**79.27**	2
SDP* [16]	90.14	88.85	78.38	0.40
RefineNet (ours)	89.88	79.17	66.38	**0.22**
Faster R-CNN* [9]	86.71	81.84	71.12	2
3DVP [36]	87.46	75.77	65.38	40
Regionlets [30]	84.75	76.45	59.70	1
SubCat [31]	84.14	75.46	59.71	0.7
OC-DPM [40]	74.94	65.95	53.86	10

89.88%, 79.17%, and 66.38% on the easy, moderate, and hard settings KITTI benchmark [18], respectively. In Table IV, we compare AUC at different different difficulty settings. First, we note that all state-of-the-art detectors (including Faster R-CNN) employ the more powerful but more computationally expensive VGG16 [5] network. On the other hand, our model is light weight in memory and run-time, yet reaching state-of-the-art on 'easy' settings. Furthermore, the RefineNet model performs nearly at the same performance level on moderate settings as the Faster R-CNN model which serves as the closest baseline (not employing other complementary modifications as in SubCNN [21] and 3DOP [15]). As moderate settings include challenging cases of higher truncation, occlusion, and small sized objects, the results are encouraging as RefineNet is an order of magnitude faster in run-time, and is also significantly faster to train.

Fig. 3 demonstrates the improvement due to the iterative refinement on a variety of scenes and visual challenges. In particular, large pose variation, occlusion, truncated, and small instances are all shown to be better handled by RefineNet compared to the baseline. The figure demonstrates how RefineNet better captures the true geometry of objects, seen by tighter boxes and correct object boundary identification, even when an object is occluded by another object. The improved localization is crucial for driver assistance applications, where 3D distance is often estimated using the object location and size in the scene. Fig. 3 demonstrates cases which still need to be resolved in future work. These include severe occlusion by other objects or truncation, where RefineNet may improve but not fully recover the correct location of an object. Further modifications to the proposed framework are needed in order to handle such challenging cases.

Evaluation on US highway settings: For additional analysis, we test the model trained on KITTI on a multi-perspective highway video dataset captured in our lab. As KITTI has only front-view camera in European Urban scenes, this allows evaluating the generalization of the RefineNet framework. As will be demonstrated, the iterative refinement will show a benefit across drastic scene variations and camera perspectives. This is crucial for a robust vehicle detection system for autonomous driving. For evaluation, we follow KITTI with

Fig. 3. Improvement of RefineNet is shown under occlusion, pose variation, and variation in size. Sample detection boxes generated using RefineNet model M_2 on KITTI validation set. In the image, orange, yellow and green color represents bounding boxes at iterations 1, 2 and 3 respectively. Confidence scores are shown next to the detection boxes.



(a) Front (b) Right (c) Rear (d) Left

Fig. 4. Performance curves with and without the proposed refinement on the multi-perspective highway dataset. Area under the curve improvement is shown for each of the perspectives. Evaluation includes partial occlusion and partial truncation instances.

TABLE V
COMPARING DIFFERENT DETECTORS, THE DEFORMABLE PARTS MODEL [29], SUBCAT [31], AND FASTER R-CNN [9], AGAINST THE PROPOSED REFINENET MODEL FOR TRACKING WITHIN EACH INDIVIDUAL PERSPECTIVE AND OVERALL ON THE HIGHWAY DATASET.

Methods	**MOTA↑**	**MOTP↑**	**IDS↓**	**Frag↓**	**MT↑**	**ML↓**	**Recall↑**	**Precision↑**
Front Camera								
DPM	0.71	0.78	0	0	0.80	0.10	0.81	0.89
SubCat	0.82	0.83	1	1	0.80	0.00	0.83	1.00
Faster R-CNN	0.74	0.83	0	0	0.60	0.10	0.74	1.00
RefineNet (proposed)	0.77	0.84	0	1	0.80	0.10	0.80	0.97
Rear Camera								
DPM	0.87	0.80	1	4	0.75	0.00	0.87	1.00
SubCat	0.82	0.85	0	9	0.75	0.00	0.87	0.94
Faster R-CNN	0.87	0.84	3	8	0.75	0.00	0.87	1.00
RefineNet (proposed)	0.88	0.86	0	4	0.75	0.00	0.90	0.98
Left Camera								
DPM	0.77	0.80	0	1	0.40	0.40	0.77	1.00
SubCat	0.76	0.77	0	1	0.40	0.20	0.76	1.00
Faster R-CNN	0.87	0.79	0	1	0.80	0.20	0.88	0.99
RefineNet (proposed)	0.82	0.81	0	1	0.80	0.20	0.84	0.98
Right Camera								
DPM	0.62	0.82	0	0	0.67	0.33	0.62	1.00
SubCat	0.55	0.83	0	0	0.33	0.33	0.55	1.00
Faster R-CNN	0.48	0.85	0	0	0.00	0.33	0.48	1.00
RefineNet (proposed)	0.62	0.85	0	0	0.00	0.00	0.62	1.00
Overall								
DPM	0.79	0.79	1	**5**	**0.69**	0.15	0.83	0.95
SubCat	0.81	0.84	1	11	0.65	**0.08**	0.83	0.97
Faster R-CNN	0.81	0.83	3	9	0.62	0.12	0.81	**1.00**
RefineNet (proposed)	**0.83**	**0.85**	**0**	6	**0.69**	**0.08**	**0.84**	0.98

a 70% overlap threshold and evaluation on three truncation and occlusion levels.

Fig. 4 demonstrates a consistent improvement across the four camera perspectives. As shown in the figure, perspectives with similar views to KITTI (front and rear) particularly benefit the refinement with RefineNet, by up to $5-6\%$ AUC increase. Side views on the other hand contain appearance variations leading to aspect ratios which are not found in KITTI. Also, as the highway dataset is captured with a wide angled settings, there is more distortion introduced into the appearance of objects. This is one reason for why the detection performance gains are smaller on the side perspectives (but still significant). As aspect ratio statistics are very different, the RefineNet model often improves localization in one dimensions of the bounding box while somewhat reducing localization in another. Generally, as side views often contain distortion due to the perspective of the camera, further improvements are required for handling generalization over such cases. This insight provides an interesting future work to pursue. Example cases are shown in Fig. 7.

Impact on 2D/3D tracking: We employ the US highway dataset in order to evaluate impact of detection performance on a state-of-the-art tracker (MDP [41]). The purpose here is to demonstrate the usefulness of the proposed RefineNet approach in generating boxes which are tighter, and are therefore prone to less errors when tracked. Fixing the tracker, we choose optimal settings for four detectors, the Deformable Parts Model [29], SubCat [31], Faster R-CNN [9], and the proposed RefineNet. We note that Faster R-CNN corresponds to no refinement, and hence it is the main comparative baseline.

TABLE VI
COMPARING DIFFERENT DETECTORS FOR A MULTI-PERSPECTIVE 3D TRACKING TASK.

Methods	MOTA↑	MOTEP↓	IDS↓	Frag↓	MT↑	ML↓	Recall↑	Precision↑
DPM [29]	0.42	1.23	**3**	83	0.38	0.15	0.74	0.70
SubCat [31]	0.64	1.23	5	93	0.46	0.15	0.79	**0.85**
Faster R-CNN [9]	0.61	1.09	19	76	**0.54**	0.08	0.80	0.81
RefineNet (proposed)	**0.65**	**1.05**	**3**	**66**	**0.54**	**0.00**	**0.83**	0.82

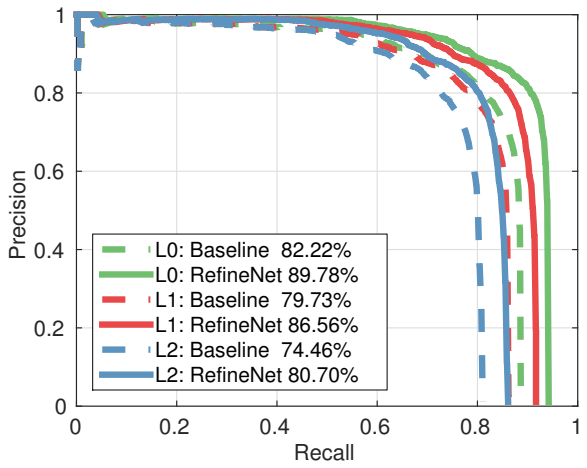


Fig. 5. Improvement due to the proposed refinement framework for different evaluation settings of 'L0' - no occlusion or truncation, 'L1' - partial occlusion and truncation, 'L2' - all instances, including heavy occlusion and truncation. The precision-recall curves for each evaluation setting are computed over all of the four perspectives.

As shown in Table V, RefineNet outperforms all the baselines by a large margin when tracking the boxes in 2D. The methods are sorted by the MOTA metric [42]. The results demonstrate how improved localization results greatly impacts ID switches (improving over all baselines), a low number of fragmented trajectories, high mostly tracked and low mostly lost [43], and highest recall and precision. This experiment quantifies an important element of RefineNet, in which subsequent vision tasks benefit significantly from the tighter and re-scored boxes.

Autonomous driving involves accurate 3D localization of surrounding objects [44], [45]. Hence, in addition to the improvement in image-plane tracking, we also analyze impact on 3D tracking and localization. In monocular settings, this can be done using a projection to a ground plane. The highway video dataset has been calibrated accordingly so that we can measure the quality of 3D tracks obtained by different image-based object detectors. Objects are first tracked in each perspective in 2D using MDP, and consequently tracked in 3D (ground plane) using a Kalman filter. Some of the tracking metrics need to be revised to use a Euclidean distance with the ground truth projections instead of the 2D overlap. Specifically, the MOTEP metric [19] reflects quality of 3D localization. Table VI shows the significant improvement of iterative refinement on 3D tracking, and the results are visualized in Fig. 6. The large performance gains due to refinement demonstrate how much subsequent vision tasks, such as behavior analysis of surrounding vehicles, can also benefit from the improvements proposed in this work. Specifically, the MOTEP metric is reduced from 1.09 to 1.05 due to refinement, and ID switches are reduced from 19 to just 3. The results significantly outperform the DPM and SubCat results for this task, addressing the question of whether lower detection quality can be tolerated with a tracker.

Fig. 6 visualizes all of the trajectories in the highway dataset. Comparing among trackers, we observe longer trajectories which are more accurately localized in the ground plane. Difficult scenarios of large movement are shown to be better handled as well. Tracking cases which are entirely missed by the baseline detector re-appear with RefineNet. When considering a situation where activity of surrounding agents needs to be recognized or predicted, these performance gains are crucial.

VI. CONCLUDING REMARKS

In this paper, we proposed and analyzed an iterative refinement framework for deep object detectors. The method is shown to significantly improve localization accuracy of vehicle detection in a variety of settings, datasets, and camera perspectives. The analysis demonstrated good performance with fast run-time speed. Specifically, RefineNet runs in about 0.22 seconds per image on images of size 1242×375, while allowing for smaller convolutional neural networks to operate on similar performance level to very deep and large networks. The improvement in localization was shown to significantly impact subsequent vision tasks, including 2D/3D object tracking.

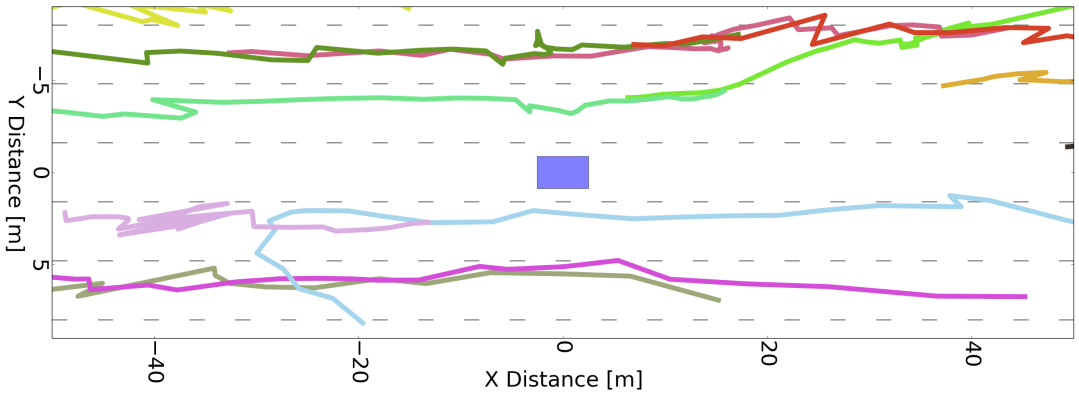
In the future, utilization of scene information [12], [46] in generating or pruning proposals can further provide increased run-time speed without sacrificing detection and localization quality. A refinement module with multi-resolution analysis [47], [48] can benefit detection of small and challenging instances. The general idea of iterative refinement can be employed to improve a variety of vision tasks, from orientation and landmark estimation to activity recognition.

ACKNOWLEDGMENT

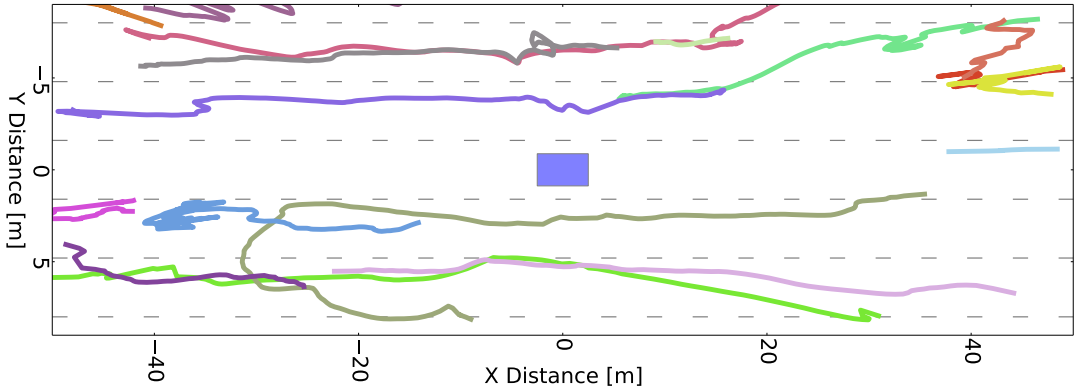
We would like to thank reviewers for their careful reading and constructive suggestions for improving the clarity and quality of this paper. We thank our associated industry sponsors, in particular Toyota-CSRC and Dr. Pujitha Gunaratne, for supporting this research.

REFERENCES

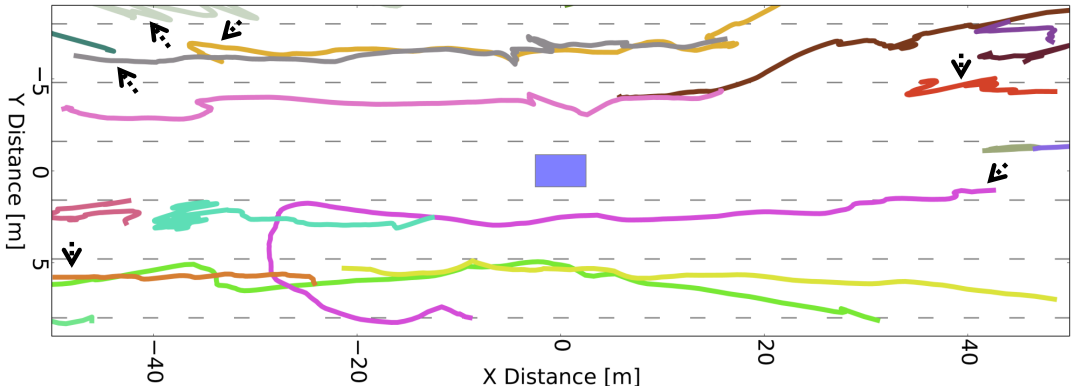
[1] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, 2013.



(a) Ground-truth



(b) Faster R-CNN baseline



(c) RefineNet (proposed)

Fig. 6. RefineNet helps 3D tracking. Comparing ground-plane projections, we observe how RefineNet boxes results in less ID switches, longer tracks, and more accurate localization in the ground plane. Each trajectory is color coded using a random color across the experiments (as track IDs vary), but arrows in (c) are shown to guide the comparison.

[2] S. Sivaraman, B. Morris, and M. M. Trivedi, "Observing on-road vehicle behavior: Issues, approaches, and perspectives," in *IEEE Conf. Intelligent Transportation Systems*, 2013.

[3] B. Ranft and C. Stiller, "The role of machine vision for intelligent vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 8–19, 2016.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 1097–1105.

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.

[6] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," *CoRR*, vol. abs/1409.4842, 2014.

[7] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Computer Vision and Pattern Recognition*, 2014.

[8] R. Girshick, "Fast r-cnn," in *International Conference on Computer Vision*, 2015.

[9] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015.

[10] L. Huang, Y. Yang, Y. Deng, and Y. Yu, "Densebox: Unifying landmark localization with end to end object detection," *CoRR*, vol. abs/1509.04874, 2015.

[11] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[12] E. Romera, L. M. Bergasa, and R. Arroyo, "Can we unify monocular

(a)

(b)

Fig. 7. RefineNet results on a four-perspective US highway dataset with training on KITTI. In general, vehicles in the front view are better detected with RefineNet over the baseline, as shown in scenes (a) and (b), while side view vehicles are challenging due to distortion and aspect-ratio variation not found in KITTI.

detectors for autonomous driving by using the pixel-wise semantic segmentation of cnns?" in *IEEE Intelligent Vehicles Symposium*, 2016.

[13] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *CoRR*, vol. abs/1511.07122, 2015.

[14] G. Lin, C. Shen, I. D. Reid, and A. van den Hengel, "Efficient piecewise training of deep structured models for semantic segmentation," *CoRR*, vol. abs/1504.01013, 2015.

[15] X. Chen, K. Kundu, Y. Zhu, A. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems*, 2015.

[16] F. Yang, W. Choi, and Y. Lin, "Exploit all the layers: Fast and accurate cnn object detector with scale dependent pooling and cascaded rejection classifiers," in *Conference on Computer Vision and Pattern Recognition*, 2016.

[17] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi, "RefineNet: Iterative refinement for accurate object localization," in *IEEE Conference on Intelligent Transportation Systems*, 2016.

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

[19] J. Dueholm, M. Kristoffersen, R. Satszoda, T. Moeslund, and M. M. Trivedi, "Trajectories and behaviors of surrounding vehicles using panoramic camera arrays," *IEEE Transactions on Intelligent Vehicles*, 2016.

[20] J. V. Dueholm, M. S. Kristoffersen, R. Satzoda, E. Ohn-Bar, T. B. Moeslund, and M. M. Trivedi, "Multi-perspective vehicle detection and tracking: Challenges, dataset, and metrics," in *IEEE Conf. Intelligent Transportation Systems*. IEEE, 2016, pp. 959–964.

[21] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Subcategory-aware convolutional neural networks for object proposals and detection," in *arXiv:1604.04693*, 2016.

[22] A. Doshi and M. M. Trivedi, "Tactical driver behavior prediction and intent inference: A review," in *IEEE Conf. Intelligent Transportation Systems*, 2011.

[23] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 90–104, 2016.

[24] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *CoRR*, vol. abs/1311.2901, 2013.

[25] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in Neural Information Processing Systems*, 2013, pp. 2553–2561.

[26] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *International journal of computer vision*, vol. 104, no. 2, pp. 154–171, 2013.

[27] J. J. Yebes, L. M. Bergasa, and M. García-Garrido, "Visual object recognition with 3d-aware features in kitti urban scenes," *Sensors*, vol. 15, no. 4, pp. 9228–9250, 2015.

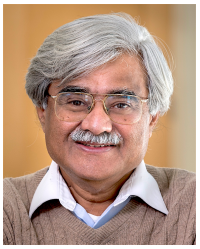
[28] E. Ohn-Bar and M. M. Trivedi, "Multi-scale volumes for deep object detection and localization," *Pattern Recognition*, 2016.

[29] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE*

Transactions on Pattern Analysis and Machine Intelligence, vol. 32, no. 9, pp. 1627–1645, 2010.
[30] X. Wang, M. Yang, S. Zhu, and Y. Lin, "Regionlets for generic object detection," in *International Conference on Computer Vision*, 2013.
[31] E. Ohn-Bar and M. M. Trivedi, "Learning to detect vehicles by clustering appearance patterns," *IEEE Transactions on Intelligent Transportation Systems*, 2015.
[32] Z. Tu, "Auto-context and its application to high-level vision tasks," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
[33] J. Ba, V. Mnih, and K. Kavukcuoglu, "Multiple object recognition with visual attention," *CoRR*, vol. abs/1412.7755, 2014.
[34] K. Gregor, I. Danihelka, A. Graves, and D. Wierstra, "DRAW: A recurrent neural network for image generation," *CoRR*, vol. abs/1502.04623, 2015.
[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.
[36] Y. Xiang, W. Choi, Y. Lin, and S. Savarese, "Data-driven 3D voxel patterns for object category recognition," in *Computer Vision and Pattern Recognition*, 2015.
[37] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi, "An exploration of why and when pedestrian detection fails," in *IEEE Conf. Intelligent Transportation Systems*, 2015.
[38] J. J. Yebes, L. M. Bergasa, R. Arroyo, and A. Lázaro, "Supervised learning and evaluation of KITTI cars detector with DPM," in *IEEE Intelligent Vehicles Symposium*, 2014.
[39] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi, "A study of vehicle detector generalization on U.S. highway," in *IEEE Conference on Intelligent Transportation Systems*, 2016.
[40] B. Pepik, M. Stark, P. Gehler, and B. Schiele, "Occlusion patterns for object class detection," in *Conference on Computer Vision and Pattern Recognition*, 2013.
[41] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *International Conference on Computer Vision*, 2015.
[42] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *EURASIP Journal on Image and Video Processing*, vol. 2008, no. 1, pp. 1–10, 2008.
[43] Y. Li, C. Huang, and R. Nevatia, "Learning to associate: Hybridboosted multi-target tracker for crowded scene," in *Computer Vision and Pattern Recognition*, 2009.
[44] B. Okumura, M. R. James, Y. Kanzawa, M. Derry, K. Sakai, T. Nishi, and D. Prokhorov, "Challenges in perception and decision making for intelligent automotive vehicles: A case study," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 20–32, 2016.
[45] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.
[46] A. D. Costea and S. Nedevschi, "Multi-class segmentation for traffic scenarios at over 50 fps," in *IEEE Intelligent Vehicles Symposium*, 2014.
[47] R. N. Rajaram, E. Ohn-Bar, and M. M. Trivedi, "Looking at pedestrians at different scales: A multiresolution approach and evaluations," *IEEE Transactions on Intelligent Transportation Systems*, 2016.
[48] A. D. Costea, A. V. Vesa, and S. Nedevschi, "Fast pedestrian detection for mobile devices," in *IEEE Conference on Intelligent Transportation Systems*, 2015.

Eshed Ohn-Bar received the B.S. degree in mathematics from the University of California Los Angeles, and the M.S. degree in electrical engineering from the University of California San Diego. He is currently pursuing a Ph.D. degree at UCSD with a focus on signal and image processing in the Laboratory for Safe and Intelligent Automobiles. His research interests include vision for intelligent vehicles, driver assistance and safety systems, computer vision, object detection and tracking, multi-modal behavior recognition, and human-robot interactivity.

Rakesh Nattoji Rajaram received the bachelor's degree in electrical engineering from International Institute of Information Technology, Hyderabad, India, and the master's degree in intelligent systems, robotics and control at University of California San Diego. His research interests include computer vision, machine learning, intelligent vehicles, and autonomous robots.



Mohan Manubhai Trivedi is a distinguished Professor of Electrical and Computer Engineering and founding director of the Computer Vision and Robotics Research Laboratory and LISA: Laboratory for Intelligent and Safe Automobiles at UCSD. LISA was awarded the IEEE Intelligent Transportation Systems "LEAD Institution" award in 2015. Trivedi received B.E (with honors, 1974) from BITS, India and PhD (1979) from the Utah State University. Currently, LISA team members are pursuing research in intelligent/highly automated vehicles, machine perception, machine learning, human-robot interactivity, driver assistance, active safety and intelligent transportation systems. LISA team has played a key role in several major research collaborative initiatives. These include human-centered vehicle collision avoidance systems, vision based passenger protection system for "smart" airbags, predictive driver intent analysis and distributed video arrays for transportation and homeland security applications.

LISA members have won over a dozen "Best" paper awards and two "Best Dissertation Awards" by the IEEE ITS Society (Dr. Shinko Cheng 2008 and Prof. Brendan Morris 2010). Prof. Trivedi has given over 100 Keynote/Plenary talks and has received IEEE ITS Society's "Outstanding Research Award" and a number of other major awards. He is a Fellow of the IEEE, IAPR, and SPIE. Trivedi serves regularly as a consultant to industry and government agencies in the USA and abroad.