2023 IEEE International Conference on Mobility, Operations, Services and Technologies (MOST) | 979-8-3503-1995-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/MOST57249.2023.00009

# Sniffer Faster R-CNN: A Joint Camera-LiDAR Object Detection Framework with Proposal Refinement

Sudip Dhakal, Qi Chen, Deyuan Qu, Dominic Carillo, Qing Yang, Song Fu
*Department of Computer Science and Engineering*
*University of North Texas*
Denton, TX, USA
{sudipdhakal, deyuanqu, dominiccarillo}@my.unt.edu, qi.chen@toyota.com, {qing.yang, song.fu}@unt.edu

***Abstract*—In this paper we present Sniffer Faster R-CNN (SFR-CNN), a novel camera-LiDAR sensor fusion framework for fast and accurate object detection in autonomous driving scenarios. The proposed detection framework architecture uses both LiDAR point clouds and Camera RGB images to generate region proposals. Current implementation of the regional proposal network (RPN) requires the generation of a large number of region proposals, majority of which are unproductive. As such, we devise a novel proposal refinement algorithm, to jointly optimize and filter a number of proposals in RPN through the combined application of both sets of LiDAR and image-based proposals thereby accelerating the LiDAR-Camera fusion algorithm without sacrificing detection precision and accuracy. Our experiments show that number of proposals is a complementary factor in determining the computational overhead in a detection network. Our proposed architecture is shown to produce state of art results on the KITTI joint object detection benchmark with the comparison being based on the execution time. While maintaining efficient detection accuracy we decrease the computational overhead by more than 20 % on the KITTI dataset.**

***Index Terms*—faster r-cnn, mmdetection, execution-time, region proposal network, regression, classification, axis-aligned bounding boxes, connected component labeling, proposal refinement algorithm.**

## I. Introduction

Object detection is a fundamental task in autonomous driving technology or in computer vision domain in general that deals with generating bounding boxes for specified object categories along with assigning class such as car, pedestrian, aeroplane, chair, etc to those categories in digital images or videos. Apart from autonomous driving, it has many practical application in video/image indexing [1], surveillance [2], object tracking, face detection and recognition, medical imaging, sports and so on. In autonomous driving object detection can be used for localization, obstacle avoidance, vehicle control, mapping, perception and planning[3]. Current object detection approaches are mainly divided into two types, i.e. single-stage approaches and two-stage appraches. As illustrated in Fig. 1, one stage detectors such as RetinaNet [4], FCOS [5], YOLO [6], SSD [7] etc. treat object detection as a simple regression problem by taking an input image and learning object classification and bounding-box regression

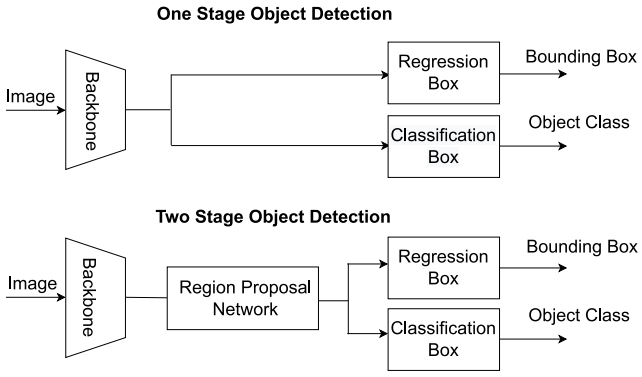



Fig. 1: Overview of One Stage and Two Stage Object Detection.

[8]. Since they are independent of additional CNN's, they are much faster then two-stage object detectors but generally reach lower accuracy rates. On the other hand, two stage detectors such as Mask R-CNN(Region-based Convolutional Neural Networks) [9] and Faster R-CNN [10] use a Region Proposal Network in the first stage to generate region of interest or simply proposals followed by sending the region proposals down the pipeline for learning the class probabilities and bounding box regression. These models typically reach the highest accuracy, but are slower due to overhead caused by the complex architecture of the multiple neural networks. In this context, finding a model that provides the optimal trade-off between accuracy and speed is not an easy task.

Although there are quite a few approaches that have been successfully implemented for accelerating the detection process in two-stage detectors there are very few methods that have addressed this problem from the perspective of region proposal network(RPN). Current implementation of Region Proposal Network requires generation of massive number of region proposals, majority of which are unproductive. Based on this principle, we hypothesize that pruning the proposal numbers in RPN, will provide an optimal trade-off between accuracy and speed. In order to improve execution

979-8-3503-1995-8/23/$31.00 ©2023 IEEE
DOI 10.1109/MOST57249.2023.00009

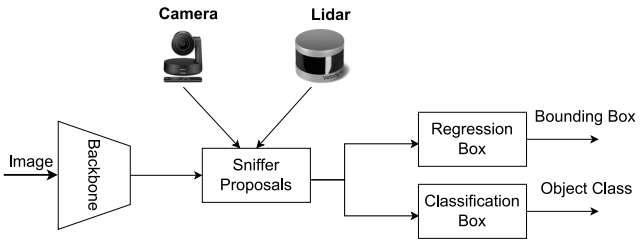



Fig. 2: Overview of Two Stage Object Detection with Camera and LiDAR.

time, we optimize the Faster R-CNN model, by replacing the proposals in regional proposal network with our own proposals named as sniffer proposals. We generate sniffer proposals through the joint application of LiDAR point cloud and camera data. We optimize the original Faster R-CNN method especially the region proposal network of the model and build our own architecture based on our hypothesis.The proposed architecture delivers the following contributions:

• a Proposal Generation approach for transforming 3D proposals obtained from LiDAR Point Cloud to 2D proposals.

• a novel Proposal Refinement Algorithm for refining and pruning proposals in the RPN network, through the combined application of both sets of LiDAR and image-based proposals.

• evaluation on the KITTI dataset shows we outperform state-of-the-art image-based methods and LiDAR-based method in terms of execution time and even manage to get higher accuracy at some instances of IoU.

## II. Related Works

In the present context, there are plenty of methods that have been implemented for the object detection task in autonomous vehicles. If we consider both accuracy and inference time, then it is an arduous task to pick only one model as the best one. Some models such as [4][5][6][7] are much faster, but provide less accurate results, while others [8][9] are more accurate, but are very fast in terms of execution speed. Therefore, finding the optimal trade off between accuracy and inference speed is a very complicated task in object detection domain. Specially, in two-stage methods that has established convolutions neural network as the state-of-the-art for detection in images [11]. In the paper [12] by Huang et al., the author studied the trade-off between accuracy and execution time for object detection model with modern convolutional neural network concept and argued that by changing various parameters such as varying meta-architecture, feature extractor, image resolution and deployment on mobile devices, the trade off can be minimized. Different from their approach, we treat Region Proposal Network as one of the principle factor for decreasing the execution time. Instead of manually changing the number of proposal without any underlying basis as proposed in [12], we showcase that through joint application of LiDAR proposals and camera image 2D proposals, we can prune the number of proposals, without removing the important ones, that will eventually impact the speed and precision.

On the other hand, [13] proposes anchor pruning methods for object detection to reduce the computational cost of the convolutional neural network. They also tend to focus on optimizing the backbone network for improving execution time. Similarly, in [14][15], the authors discuss about pruning the deep convolutional neural network to minimize the trade-off between speed and accuracy. Furthermore, [16][17] discusses about configuration of hardware component for speeding up the execution time. While these approaches have advantages on their own ways, we view the trade-off problem from a different perspective. We argue that there are far too many region proposals being generated in the region proposal layer that eventually lead to a overload in computation. We propose, by carefully removing the unproductive proposals, we can decrease the computation overhead.

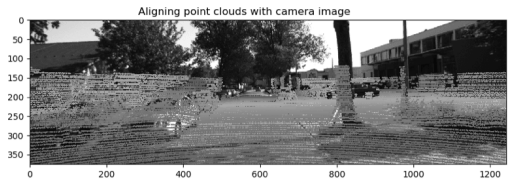



Fig. 3: Illustration on projecting LiDAR point clouds on a camera image.

## III. Proposed Framework Architecture

In this section, we develop an efficient structure-aware two-stage joint detection network for object detection and classification in autonomous vehicles.The proposed method, depicted in Figure 4, uses a subset of proposals from both LiDAR point cloud (3D proposals) and original Faster R-CNN (2D proposals). Section A introduces the mechanism we used for obtaining 2D proposals from LiDAR point cloud. Section B introduces our novel proposal refinement algorithm for filtering proposals. Similarly in Section C we will introduce our network architecture including the backbone network, feature pyramid network, RoI head and Soft Non-maximum Suppression algorithm we used for improving the detection accuracy. We aim to reduce the total number of proposals used in RPN network of Faster R-CNN or any two-stage methods and eventually decrease the execution time through the application of our architecture.

### *A. Multi-Sensor Fusion*

In the proposed two-stage multi-sensor detector we take two frames, one from a LiDAR point clouds and another from an RGB image as input. These two sensors go through an alignment process for fusion in order to compensate for each other. The basic camera-lidar cordinate transformation and calibration have been provided by the KITTI dataset [26]

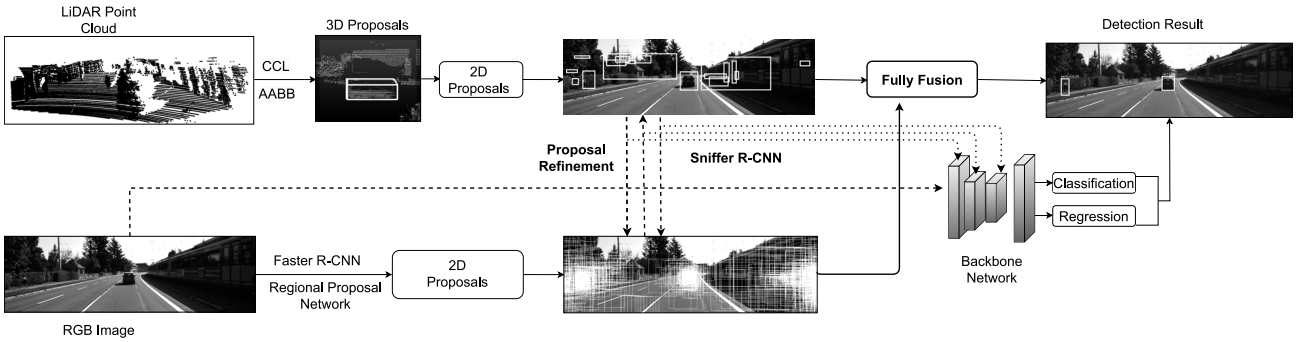



Fig. 4: The overall architecture of our proposed joint 2D and 3D object detection framework with proposal refinement.

and we use these existing tools for projecting image captured from cameras onto the LiDAR data as shown in Fig. 3.

### *B. LiDAR Proposal Generation*

To showcase possible objects in the sensing data, connected components are extracted from the point clouds as shown in Fig. 5. We obtain 3D proposals with the application of the Axis-Aligned Bounding Boxes (AABB) method from the LiDAR data [19]. As shown in the Fig. 5 we project 3D proposals onto a front-view camera image. We then select front, middle and rear cross sections on 3D proposals. These vertexes are transformed to image coordinates to draw 2D boxes (coloured in 3 levels of black color as shown in Fig. 5).

In a connected component environment where there are multiple points cloud segmented together we obtain connected components from LiDAR data. In point clouds data there are multiple points that are segmented into smaller parts which are separated by a minimum distance. Each part is a set of connected points. We have derived this approached from the classic image processing algorithm, called Connected Component Labeling (CCL) [20]. Given a set of point cloud this algorithm is used to detect connected regions. Once the first point of a connected component is identified, it is easy to find and gather all the other points in that particular connected component and this is the primary reason why this method is effective as well as efficient. The next step is to generate 3D proposals and for that we use a 3D grid to extract these connected components. We collect this grid from the octree structure which divides a given 3D space into at most eight part in order to store points. We can control how small the minimum gap can be between the adjacent connected component by selecting a different octree level. Finally, to speed up the generation of 3D proposals, components with points less than a specific number are ignored. Next we perform segmentation of LiDAR point cloud into disconnected components, where each box indicates a candidate object in the given sensing data.

Once the 3D segmentation is completed, we use our alignment approach discussed in Section A, B, and C to produce 2D bounding boxes from LiDAR point clouds, these are termed as LiDAR Proposals denoted by L.

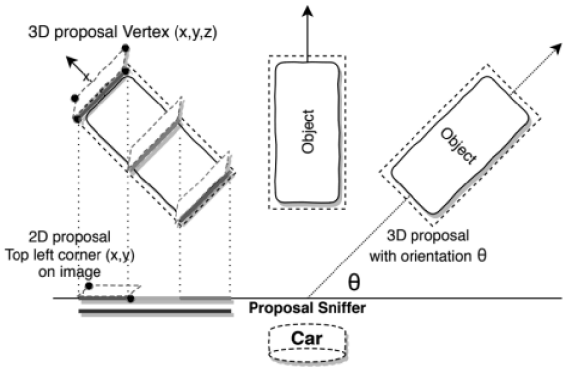



Fig. 5: Relations between 3D and 2D proposals illustrated in the LiDAR Proposal Generation module.

### *C. Proposal Refinement Algorithm*

In this section we showcase our novel proposal refinement algorithm. The proposal refinement algorithm as shown in Algorithm 1 inputs two types of proposals. First, set if 2D LiDAR proposals obtained from the Section D "LiDAR Proposal generation". There are less than 100 LiDAR proposals per images. Second, original 2D proposals obtained via Regional Proposal Network of Faster R-CNN model( 1000 proposals per image) as R. Clearly the number of proposal in original RPN is substantial. On the other hand, 2D LiDAR proposals generated from the proposal generation part are inadequate as seen in Fig. 7 where white color proposal are the missing proposal that eventually leads to those objects being undetected if only the 2D LiDAR proposals are used in the detection network. Hence our novel proposal refinement algorithm prudently chooses only the best proposals from both sets of proposals. For a given image, we compare each proposal from LiDAR proposal(L) with each proposal from original proposal(R) i.e for each bbox( proposal) with coordinates $(r_1, r_2, r_3, r_4)$ in R, we compare each bbox (proposal) with coordinates $(l_1, l_2, l_3, l_4)$in L for a given image. And if the difference between the all the adjacent bbox coordinate meets the given threshold then we keep both the proposals. However if it fails to meets the given

**Algorithm 1:** Main Algorithm of Proposal Refinement

**Input:** set of RGB proposal $r^i \in R$, $0 < i \leq 1000$.
set of LiDAR proposal $\ell^j \in L$, $0 < j \leq 100$.
**Output:** Final sniffer proposals $S^*$.

$r_1, ..., r_4$ are vertex coordinates in $r$, RGB proposal.
$\ell_1, ..., \ell_4$ are vertex coordinates in $\ell$, LiDAR proposal.
**for** *each i and j* **do**
  **if** $|r_1^i - \ell_1^j| and |r_2^i - \ell_2^j| and |r_3^i - \ell_3^j| and |r_4^i - \ell_4^j| < \lambda$ **then**
    Keep $r$ and $\ell$ as $s_k$;
  **else**
    Filter out $r$ and $\ell$ ;
  **end**
**end**
get sniffer proposals $S = \{s_1, s_2, ..., s_k\}$, $k < 600$.
remove repeated proposals from $S$.
**return** sniffer proposals $S^* = \{s_1, s_2, ..., s_n\}$, $n \leq k$.

threshold value $\lambda$ we filter out or simply remove the proposal set from both sniffer and original proposal. We evaluate this algorithm for two different cases. In first case we keep the threshold to a less then or equal to value to address the problem related with the small LiDAR proposals. For instance, for a very small LiDAR proposals which is unable to contain any object within itself will meet the threshold with the larger original proposals which will be kept in the final proposal set thus producing good detection result. Similarly, In second case we keep the threshold to a greater then or equal to value to address the problem related with the large LiDAR proposals. For instance, for a very large LiDAR proposals which is able to contain object but will be eventually disregarded due to IoU threshold will meet the threshold with the smaller original proposals which will be kept in the final proposal set thus again contributing to good detection result. Both the cases can be achieved through the application of absolute subtraction of the bounding boxes. From both cases, we obtain certain number of proposals for a given image but there are multiple repetition due to the fact that same proposal meets the threshold for multiple other proposals and as result of this, it will be output multiple times. We remove the repeated proposals by manually checking the repeated appearance of each proposals to finally obtain final sniffer proposals as illustrated in Algorithm 1. We conduct this experiment for different threshold value to eventually obtain the best threshold for our algorithm. The number of final sniffer proposals (∼600) as shown in algorithm is based on the best threshold value which we have discussed in Section E.

### *D. Simplified Network Architecture*

We emulate the Faster R-CNN model as our base model and make necessary changes in both test configuration especially regional proposal network(RPN), and regional convolutional neural network (RCNN).In Fig. 6 we illustrate the simplified architecture of our entire framework.

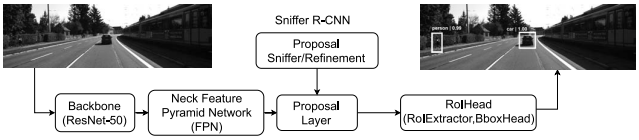



Fig. 6: Network architecture of our proposed solution.

**Backbone** Convolutional Neural Network or simply CNN are normally composed of multiple convolutional network layers and pooling layers. While convolutional layers are used to extract an over-complete representation of the input feature, pooling layers on the other hand are used to downsample the feature map size to minimize the execution and assist in creating more robust representation. Backbone is the part that transforms an image to feature maps, such as a ResNet-50 without the last fully connected layer [20] . We are using ResNet-50 as our backbone network as it is proven to have achieved the best performance in classes car and pedestrian [22].Our backbone network contains four convolutional blocks with depth 50.Each convolution is followed by a batch normalization. As a result, the backbone network produces multi-stage feature maps at different spatial resolutions[23].

**Feature Pyramid Network** Neck is the part that connects the backbone and heads. It performs some refinements or reconfiguration on the raw feature maps produced by the backbone. We adopt Feature Pyramid Network "FPN" with 256 output channels in our base model as it can be trained end-to-end with all scales and is used consistently a train/test time. FPN are able to achieve higher accuracy than all the existing state-of-art methods for feature refinement [23].

**Proposal Layer** The goal of the Proposal Layer is to prune the list of proposals produced by the LiDAR and refine them with original 2D proposals again with the application of proposal refinement algorithm and produce class more accurate bounding boxes to be used to test the classification layer to produce good classification and regression results. Leveraging proposals generated from proposal layer, ROI on corresponding images can be effectively identified, hence accelerating the detection process with a low computational cost.

**Region of Interest** RoIExtractor is the part that extracts RoIwise features from a single or multiple feature maps with RoIPooling-like operators. Similarly, RoIHead is the part that takes RoI features as input and make RoI-wise task specific predictions, such as bounding box classification/regression, mask prediction[20]. We adopt SingleRoIExtractor for this purpose.

**Soft Non-Maximum Suppression** One indispensable component of object detection is non-maximum suppression (NMS), a post-processing algorithm responsible for merging all detections that belong to the same object [23]. In NMS detection, box with the maximum score is selected and all the other detection boxes with a significant overlap are suppressed for a given threshold value. However, there is one problem with this approach. As per the design of the algorithm, if an object

lies within the predefined overlap threshold, it leads to a miss. The simple yet efficient way to deal with this case is to use Soft-NMS which instead of completely removing proposals with low score, assigns a reduced confidence score to these proposals so that they are still under consideration for the presence of other object belonging to a different class. Soft-NMS obtains consistent improvements for the coco-style mAP metric on standard datasets like PASCAL VOC 2007 (1.7 % for both RFCN and Faster-RCNN) and MS-COCO (1.3 % for R-FCN and 1.1 % for Faster-RCNN) by just changing the NMS algorithm without any additional hyper-parameters[25].

## IV. Experiments and Results

To prove experimentally the viability of our proposed framework we evaluate the performance of our model on KITTI dataset [26]. In this section we will mainly discuss about our experimental setting, implementation details and then analyze the effect and performance of our framework in the real world environment and compare it with the existing state-of-the-art methods.

### *A. Dataset*

KITTI Dataset [26] is one of the most popular dataset of 2D as well as 3D detection for autonomous driving. In KITTI dataset there are usually 7, 481 training samples and 7, 518 test samples.For experiment studies we split the official training set into three different sets;a training set of 4600 samples, a validation set of 1481 samples and a mini-test set of 200 samples. The KITTI benchmark requires detections of cars, pedestrians and cyclist. However, for the ease of experiments and easy accessibility of groundtruth for cars and pedestrians, we trained our model based on these two classes only.

### *B. Training and Inference Details.*

We use MMDetection toolbox for training, validation and testing purposes.This is an object detection toolbox that contains a rich set of object detection and instance segmentation methods as well as related components and modules [21].Our joint 2d and 3d framework is trained emulating the Faster R-CNN method in an end-to-end manner with the SGD optimizer. For the KITTI dataset, we train the entire network with the batch size 24, learning rate 0.0025 for 4 epochs on Intel Core i9 10th Gen Processor, which takes around 7 hrs.Similarly, under the same environment we test 200 samples of images in different scenarios.

### *C. Evaluation Metric.*

We adopt standard evaluation metrics for PASCAL VOC dataset evaluating the results using two widely used metrics namely mean average precision(mAP) and recall. The mAP value is calculated with a rotated IoU threshold 0.5 for cars and pedestrian. The mean average precision on the test set is calculated with 40 recall positions on the official KITTI test server. Similarly, in addition to the evaluation metrics we also assess the execution of our Proposal Refinement Algorithm and visualize the impact of the algorithm by providing the graphical representation of number of proposal kept and removed from both original and LiDAR sets of proposals. We will thoroughly discuss about this evaluation in Section E " Result of Proposal Refinement".

### *D. Comparison with state-of-the-art methods.*

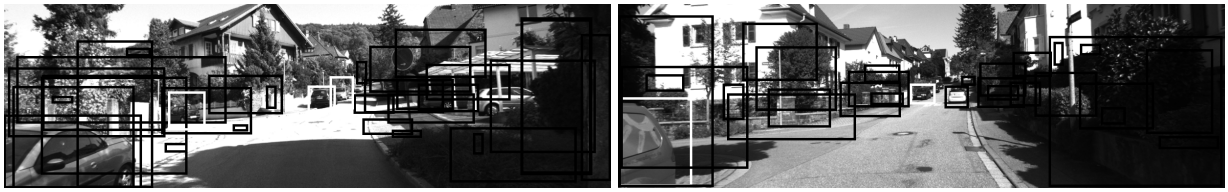
We report the performance on our KITTI test set and compare it with other state-of-art methods. Table 1 shows the performance of our framework on the KITTI test set that we extracted from the original test set. As expected the performance is very good in original Faster R-CNN method. The region proposal network of the original Faster R-CNN method consists of 2D proposals only. On the other hand, for Faster R-CNN (LiDAR Only) method we use proposals only from the LiDAR. The inference result based on LiDAR only method clearly shows that the 3D proposals are not sufficient for accurate object detection results. Although the latency is low in comparison to the original Faster R-CNN, the mAP at different instances of IoU is significantly lower. This is due to the very small number of proposals used in LiDAR only detection method. Like mentioned earlier in the section "LiDAR Proposal Generation", we used Axis-aligned Bounding Boxes followed by Connected Component Labeling method to convert 3D LiDAR proposals to 2D LiDAR proposals. In comparison to the original proposals, the number of proposals generated through this method is significantly lower as a result of which they fail to encapsulate the entirety of the image. In Fig. 8, the gray bounding boxes are the projection of LiDAR proposals and the white boxes are the ground-truth bounding boxes. We can clearly see that the there are no LiDAR proposals for some of the object in each of the test images. Hence, we can conclude that only LiDAR proposals are not enough for accurate object detection as they have missed out on several ground truth objects. On the other hand 2D proposals from the original method yields a fairly accurate results with around 1000 proposals per image, majority of which are ineffectual. Our Sniffer Faster R-CNN framework, utilizes only the most useful sets of proposals from both sets of proposals and even manages to outperform Original Faster-RCNN in some instances of IoU. It also outperforms Faster R-CNN (LiDAR only) method by a significant margin as shown in Table 1. Similarly, as indicated by our results in Table 1, Sniffer Faster R-CNN represents a significant improvement in terms of inference runtime. The latency for Sniffer Faster-RCNN is 0.1737 which is 20 % improvement over the original Faster-RCNN, which has latency of 0.2149 seconds per frame.The inference time is based on our KITTI input test samples consisting of 200 images.Similarly, we also evaluate the performance of our model for other two-stage methods and we can clearly see that the approach can be extended to any two-stage methods such as Cascade R-CNN as seen in the Table I.

### *E. Result of Proposal Refinement*

In this section we showcase the impact of our novel Proposal Refinement algorithm for filtering proposals. The original

| Method | Car | | | | Pedestrian | | | | Latency(s/f) | GFLOPs |
|---|---|---|---|---|---|---|---|---|---|---|
| | @0.4 | @0.5 | @0.6 | @0.7 | @0.4 | @0.5 | @0.6 | @0.7 | | |
| Faster R-CNN | 0.92 | 0.91 | 0.892 | 0.87 | 0.815 | 0.795 | 0.689 | 0.571 | 0.2149 | 206.76 |
| Cascade R-CNN | 0.928 | 0.92 | 0.903 | 0.864 | 0.788 | 0.761 | 0.724 | 0.517 | 0.3147 | 234.47 |
| Sniffer Cascade R-CNN | 0.92 | 0.914 | 0.898 | 0.859 | 0.764 | 0.747 | 0.715 | 0.514 | 0.1989 | 234.27 |
| Sniffer Faster R-CNN | 0.912 | 0.903 | 0.894 | 0.837 | 0.817 | 0.798 | 0.708 | 0.55 | 0.1737 | 206.67 |

TABLE I: A comparision of the performance of Sniffer Faster R-CNN with the state of the art object detectors evaluated on the KITTI test set. The results are evaluated by the mean Average Precision with 40 recall positions.



(a) (b)

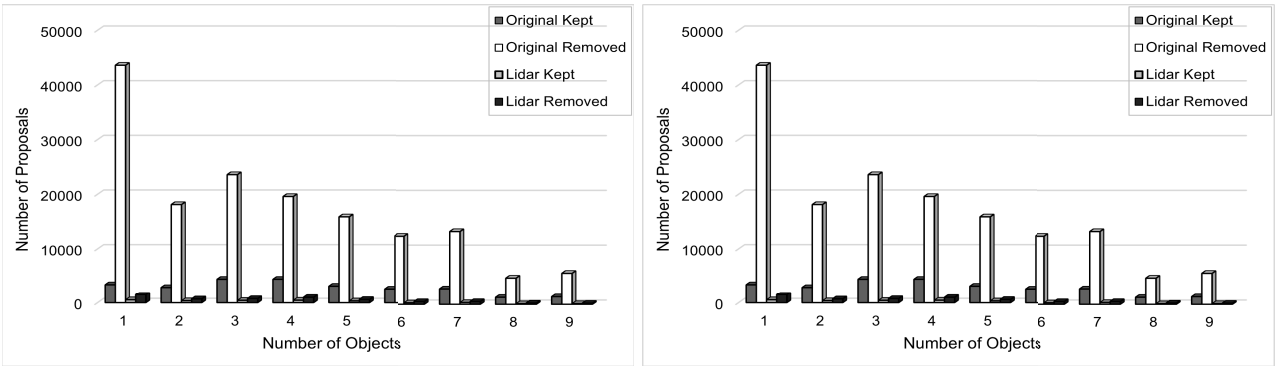
Fig. 7: Projection of 2D LiDAR proposals on KITTI test set ( white bounding boxes are missing candidate boxes/proposals in the image whereas black bounding boxes are the LiDAR proposals.

Faster R-CNN accommodated large number of proposal, majority of which are ineffectual. On the other hand, Faster R-CNN with LiDAR proposals in RPN has inadequate number of proposals. Our Proposal Refinement algorithm sorts out and utilizes only the most useful proposals from both sets of proposals. Fig. 9 and Fig. 10 demonstrate total number of proposal kept and removed for a given number of objects in test images after the application of our algorithm. It can be observed that for test images consisting of few number of objects, the number of proposals removed from both original and LiDAR proposals is significantly high. This is mainly due to the fact that less number of proposals can be considered for presence of small number of objects. For instance, let's consider a scenario where there is only one car in the vicinity, that means we do not have to draw proposals in every single part of the image. The proposals can be concentrated only to the part of the image that contains the car and this can be achieved by fewer proposals in comparison to the scenario of having large number of cars or pedestrian in the vicinity, where we will definitely need more proposals to encompass the entirety of the image so that no objects are left undetected. This is again visible in the graph given below. For instance, in images having number of object equal to 2, significant number of proposals are removed and very few proposals are kept. Similarly, in images having number of object equal to 10, few proposals are removed and large number of original as well as LiDAR proposals are kept. This is consistent with our expectation that neither an excessive, overly populated proposals nor a deficient proposals is required for good object detection result. Our algorithm yields only the useful sets of proposals that are required for object detection.This evaluation is based on threshold lambda value equal to 20. We gradually increase the value of lambda as seen in Fig. 9 to find the optimal value and as visible in the Fig. 9 the accuracy seems to be stable beginning at lambda value equal to 50. We will further discuss about the varying value of lambda in section below.

### *F. Ablation Studies*

To further validate the effectiveness of our proposed framework, we evaluate the performance of our model on varying value of lambda($\lambda$). As mentioned earlier, we have defined a lambda threshold value in our proposal refinement algorithm. We conducted separate set of test experiments for $\lambda_0$ to $\lambda_{100}$. As seen in Fig. 10. the mAP is very low at the initial stage but as the value of lambda increases, mAP also increases and becomes stable at value $\lambda_{45}$. With increasing value of $\lambda$ the total number of our final sniffer proposals in RPN also increases and the execution time also increases as seen in Fig. 11. This meets the expectation of our proposal refinement algorithm because as the threshold value increases additional or more number of proposals will the criteria and this results in more number of proposals being kept or selected from original and LiDAR proposal set. While there are 32636 proposal selected for $\lambda_{20}$ value there are around 60794 proposals selected for $\lambda_{30}$ value. From the above numbers we can conclude that latency is directly proportional to the value of $\lambda$ which is directly proportional to the number of proposals. This means the latency is also directly proportional to the number of proposals. The latency for $\lambda_{20}$ is 0.164 second per frame which is lower in comparison to the latency for $\lambda_{30}$ which stands at 0.17 second per frame. While number of proposal is not the most significant factor for decreasing the latency it is very clear that it is one of the useful factor for a accelerated execution result. This is also visible in the Fig. 11 that show the runtime analysis. TABLE II on the other hand illustrates the performance of our model when compared to the base network in terms of number of parameter used and GFLOPs.

Apart from the changes in the number of proposals in RPN and some finetuning there are no major changes in the overall architecture of our model as compared to the original Faster-




(a) (b)

Fig. 8: Impact of proposal refinement algorithm on number of proposal being kept and removed for given number of objects in an image.

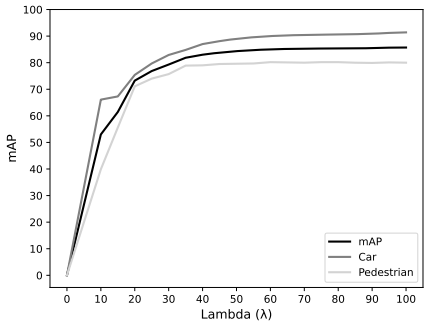



Fig. 9: The effect of variation of lambda values to the Sniffer Faster R-CNN network.

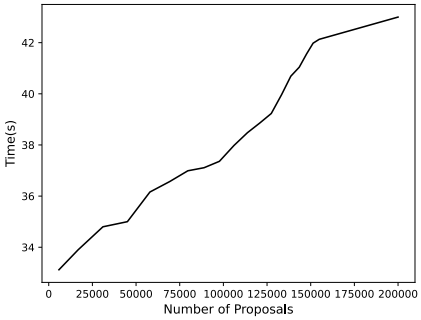



Fig. 10: Runtime analysis to demonstrate number of proposals as a useful factor for accelerating the execution time of object detection.

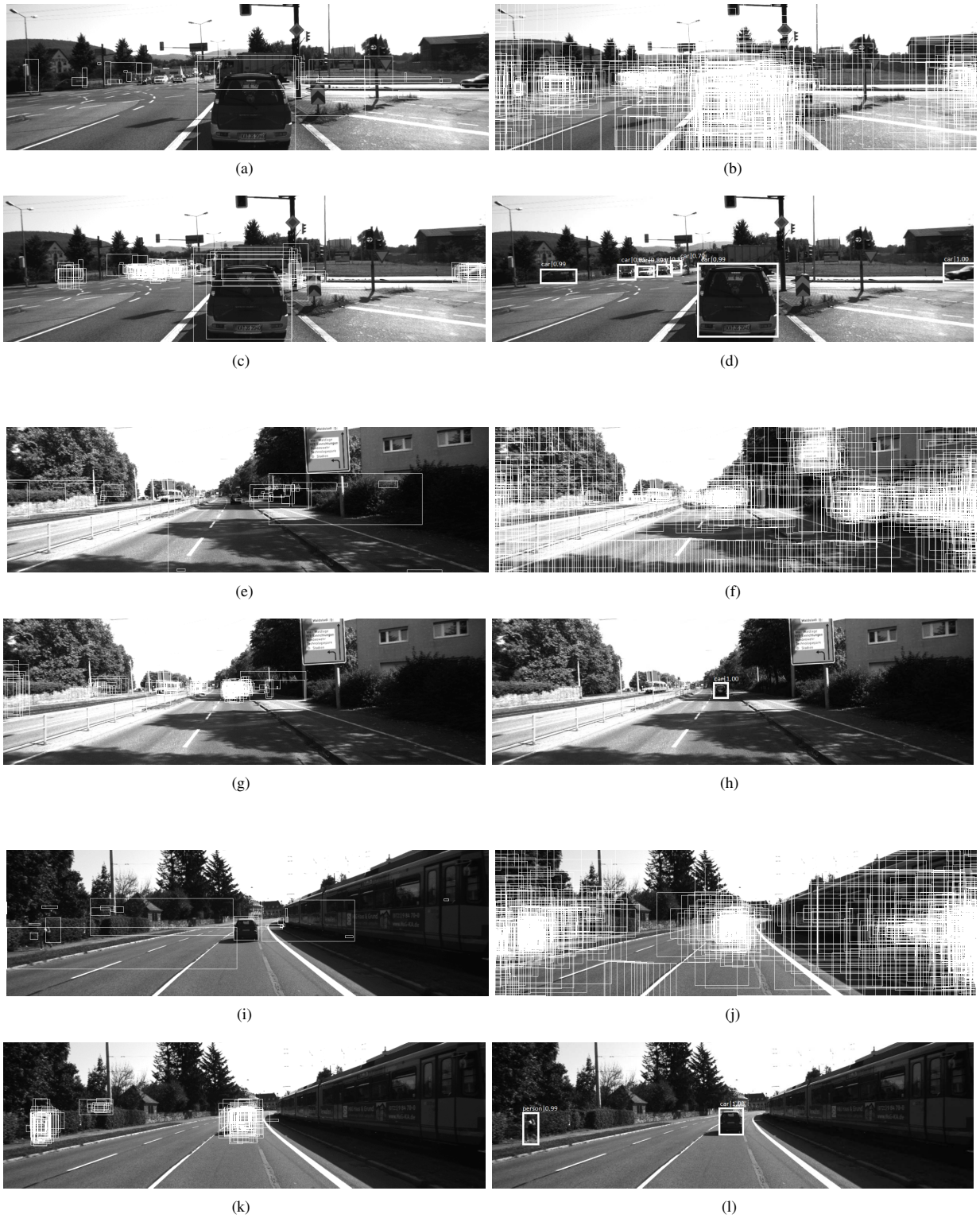
| Method | Car AP@0.5 | Pedestrian AP@0.5 | Number of Parameters | GFLOPs |
|---|---|---|---|---|
| Base Network (Faster R-CNN) | 0.91 | 0.795 | 41.22m | 206.76 |
| Base Network (LiDAR Proposals) | 0.92 | 0.761 | 41.22m | 206.76 |
| Sniffer Faster R-CNN ( 40 ) | 0.87 | 0.79 | 41.13m | 206.67 |
| Sniffer Faster R-CNN ( 50 ) | 0.903 | 0.798 | 41.13m | 206.67 |

TABLE II: A comparison of the performance of different variations of hyper parameters, evaluated on the KITTI test set. The effect of variation of hyper parameters on the GFLOPs and number of parameters ( in million) are measured relative to the base network.

RCNN method and this is clearly visible in TABLE II. As shown in the table there is no significant difference in the the number of parameter and GFLOPs value in all the cases.

### *G. Real World Object Detection Performance*

Fig. 12 shows the three selected examples of the object detection results, with the first image showing the projection of LiDAR only 2D proposals, second image showing the projection of original 2D proposals, third image showing the projection of sniffer proposals obtained after proposal refinement and the final image showing the final detection result based on the sniffer proposal. These figures demonstrate that while the proposals generated by LiDAR only are very few in number they are not sufficient for accurate object detection. On the other hand, as seen in second figure, the original method contains far too many proposals. Some of these proposals are repeated and some are nowhere near the object in the image. Therefore, they are unproductive and hence can be disregarded. This is achieved through our Proposal Refinement algorithm and the projection of sniffer proposals in the third figure demonstrate the usefulness of our algorithm. We can clearly see, the counterproductive proposals have now been disregarded and only the most useful proposal are kept and we can observe the accurate detection result in the final image.




(a) (b)

(c) (d)

(e) (f)

(g) (h)

(i) (j)

(k) (l)

Fig. 11: Projection of proposals on the KITTI test set and real world object detection results.( Fig a,e,i. shows the projection of LiDAR proposals only, Fig b,f,j. shows the projection of original proposals form base network of Faster R-CNN. And, Fig c,g,k. shows the projection of our sniffer proposals and Fig d,h,l. shows the detection results with our Sniffer Faster R-CNN.

According to these figures, our proposed framework Sniffer R-CNN has been very successful in proposing accurate bounding boxes. In our experiments, Proposal Refinement algorithm was able to generate proposals with numbers significantly lower in comparison to the original number of proposals. As mentioned earlier, total number of proposals in region proposal network is one of the contributing factor in execution time of the object detection framework.We maintained the accuracy of the detection and reduced the execution time by reducing the number of proposal as seen in the real word object detection results. Some proposals missed by the LiDAR proposal are incorporated from original set of proposals to maintain the accuracy.

## V. Conclusion

In this paper, we presented a novel Region Proposal Network for object detection that inherently performs as a sensor fusion algorithm, combining the data obtained from LiDAR with vision data to obtain faster and accurate object detection results. Our Proposal Refinement algorithm is able to produce accurate region proposals while significantly decreasing the number of proposal at the same time and maintain the accuracy of the detection. Experiments on the KITTI dataset show the superiority of our proposed architecture over the state of art in terms of inference time by decreasing the execution time by 20 %. Similarly, testing results of the real trace dataset from KITTI show that, number of proposals in RPN is one of the important factor in execution delay. We studied that current state-of-art RPN contains and depend on substantial number of proposal for detection, majority of which are unavailing. Through joint application of proposals from both 2D and 3D approach we decrease the number of proposals in RPN by more then 40 % in comparision to original methods and when the value of IoU is high, our method is able to achieve a much better average precision as well.

## Acknowledgement

The work is supported by National Science Foundation grants CNS-1852134, OAC-2017564, and ECCS-2010332.

## References

[1] Snoek, C.G., Worring, M. Multimodal Video Indexing: A Review of the State-of-the-art. Multimedia Tools and Applications 25, 5–35 (2005).https://doi.org/10.1023/B:MTAP.0000046380.27575.a5

[2] F. F. Chamasemani, L. S. Affendey, F. Khalid and N. Mustapha, "Object detection and representation method for surveillance video indexing," 2015 IEEE 3rd International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA), 2015, pp. 1-5, doi: 10.1109/ICSIMA.2015.7559038.

[3] S. Dhakal, D. Qu, D. Carrillo, Q. Yang and S. Fu, "OASD: An Open Approach to Self-Driving Vehicle," 2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD), 2021, pp. 54-61, doi: 10.1109/MetroCAD51599.2021.00017.

[4] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal Loss for Dense Object Detection," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 2, pp. 318-327, 1 Feb. 2020, doi: 10.1109/TPAMI.2018.2858826.

[5] Z. Tian, C. Shen, H. Chen and T. He, "FCOS: A Simple and Strong Anchor-free Object Detector," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2020.3032166.

[6] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[7] Liu, Wei, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. "Ssd: Single shot multibox detector." In European conference on computer vision, pp. 21-37. Springer, Cham, 2016.

[8] Soviany and R. T. Ionescu, "Optimizing the Trade-Off between Single-Stage and Two-Stage Deep Object Detectors using Image Difficulty Prediction," 2018 20th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), 2018, pp. 209-214, doi: 10.1109/SYNASC.2018.00041.

[9] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 2980-2988, doi: 10.1109/ICCV.2017.322.

[10] S. Ren, K. He, R. 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, vol. 39, no. 6, pp. 1137-1149, 1 June 2017, doi: 10.1109/TPAMI.2016.2577031.

[11] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang and O. Beijbom, "PointPillars: Fast Encoders for Object Detection From Point Clouds," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 12689-12697, doi: 10.1109/CVPR.2019.01298.

[12] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama et al., "Speed/accuracy trade-offs for modern convolutional object detectors", Proceedings of CVPR, pp. 7310-7319, 2017.

[13] Bonnaerens, Maxim, Matthias Freiberger and Joni Dambre. "Anchor Pruning for Object Detection." ArXiv abs/2104.00432 (2021): n. pag.

[14] WANG Sheng-sheng WANG Meng WANG Guang-yao. Deep Neural Network Pruning Based Two-Stage Remote Sensing Image Object Detection[J]. Journal of Northeastern University Natural Science, 2019, 40(2): 174-179.

[15] C. Zhao, B. Ni, J. Zhang, Q. Zhao, W. Zhang and Q. Tian, "Variational Convolutional Neural Network Pruning," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 2775-2784, doi: 10.1109/CVPR.2019.00289.

[16] Wang, Robert Li, Xiang Ao, Shuang Ling, Charles. (2018). Pelee: A Real-Time Object Detection System on Mobile Devices.

[17] Zhang, Xiaofan, Haoming Lu, Cong Hao, Jiachen Li, Bowen Cheng, Yuhong Li, Kyle Rupnow, Jinjun Xiong, Thomas E. Huang, Humphrey Shi, Wen-mei W. Hwu and Deming Chen. "SkyNet: a Hardware-Efficient Method for Object Detection and Tracking on Embedded Systems." ArXiv abs/1909.09709 (2020): n. pag.

[18] A. Geiger, F. Moosmann, Ö. Car and B. Schuster, "Automatic camera and range sensor calibration using a single shot," 2012 IEEE International Conference on Robotics and Automation, 2012, pp. 3936-3943, doi: 10.1109/ICRA.2012.6224570.

[19] Martin Stich, Heiko Friedrich, and Andreas Dietrich, Spatial splits in bounding volume hierarchies, Proceedings of the Conference on High Performance Graphics 2009, 2009, pp. 7–13.

[20] Alexander JB Trevor, Suat Gedikli, Radu B Rusu, and Henrik I Christensen, Efficient organized point cloud segmentation with connected components, Semantic Perception Mapping and Exploration (SPME) (2013).

[21] L. Chen et al., "Deep Neural Network Based Vehicle and Pedestrian Detection for Autonomous Driving: A Survey," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 6, pp. 3234-3246, June 2021, doi: 10.1109/TITS.2020.2993926.

[22] Chen, Kai Wang, Jiaqi Pang, Jiangmiao Cao, Yuhang Xiong, Yu Li, Xiaoxiao Sun, Shuyang Feng, Wansen Liu, Ziwei Xu, Jiarui Zhang, Zheng Cheng, Dazhi Zhu, Chenchen Cheng, Tianheng Zhao, Qijie Li, Buyu Lu, Xin Zhu, Rui Wu, Yue Lin, Dahua. (2019). MMDetection: Open MMLab Detection Toolbox and Benchmark.

[23] C. He, H. Zeng, J. Huang, X. -S. Hua and L. Zhang, "Structure Aware Single-Stage 3D Object Detection From Point Cloud," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, pp. 11870-11879, doi: 10.1109/CVPR42600.2020.01189.

[24] T. Lin, et al., "Feature Pyramid Networks for Object Detection," in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017 pp. 936-944.doi: 10.1109/CVPR.2017.106

[25] Bodla, Navaneeth Singh, Bharat Chellappa, Rama Davis, Larry. (2017). Improving Object Detection With One Line of Code.

[26] A. Geiger, P. Lenz and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 3354-3361, doi: 10.1109/CVPR.2012.6248074.