Detection of Imaged Objects with Estimated Scales

Xuesong Li, Ngaiming Kwok, Jose E. Guivant, Karan Narula, Ruowei Li and Hongkun Wu
School of Mechanical and Manufacturing Engineering,
The University of New South Wales, NSW 2052, Australia

Keywords: Computer Vision, Object Detection, Convolutional Neural Network.

Abstract: Dealing with multiple sizes of the object in the image has always been a challenge in object detection. Pre-defined multi-size anchors are usually adopted to address this issue, but they can only accommodate a limited number of object scales and aspect ratios. To cover a wider multi-size variation, we propose a detection method that utilizes depth information to estimate the size of anchors. To be more specific, a general 3D shape is selected, for each class of objects, that represents different sizes of 2D bounding boxes in the image according to the corresponding object depths. Given these 2D bounding boxes, a neural network is used to classify them into different categories and do the regression to obtain more accurate 2D bounding boxes. The KITTI benchmark dataset is used to validate the proposed approach. Compared with the detection method using pre-defined anchors, the proposed method has achieved a significant improvement in detection accuracy.

1 INTRODUCTION

Object detection is an important topic in computer vision community. Classical methods are usually developed based on handcrafted image features, such as histograms of oriented gradient (HOG) (Dalal and Triggs, 2005) and the deformable part model (DPM) (Felzenszwalb et al., 2008). Recently, convolutional neural network (CNN) has attracted much research attention due to its astonishing performance for object detection task(Ren et al., 2015; Liu et al., 2015; He et al., 2017), whereby the image features are automatically learned rather than handcrafted. These frameworks (Ren et al., 2015; Liu et al., 2015; Redmon et al., 2015) are widely used in image-based object detection problems.

CNN-based object detection methods consist of two main components: classification and localization. Classification mainly relies on features of the target object while localization regression depends on both features and object sizes. CNN is able to find scale-invariant and object-specific features to identify the objects; however, object sizes in the image are not know a priori which makes detection tasks much more difficult than classification tasks. In real-world applications, objects captured in an image usually have a large scale variation such as shown in Fig.1; that makes object detection even more challenging. In order to detect objects of different sizes, strategies including image pyramid and multiple anchors were proposed. Multi-size anchors are more commonly utilized to fit their corresponding objects, due to computation efficiency and cheap memory cost. For example, in the Faster-RCNN (Ren et al., 2015), 9 discrete anchors with 3 scales and 3 aspect ratios are adopted to handle all object sizes. Then the most promising anchors are the ones with similar sizes to the objects in the image, and they are selected as inputs to the next stage classifier and regressor.

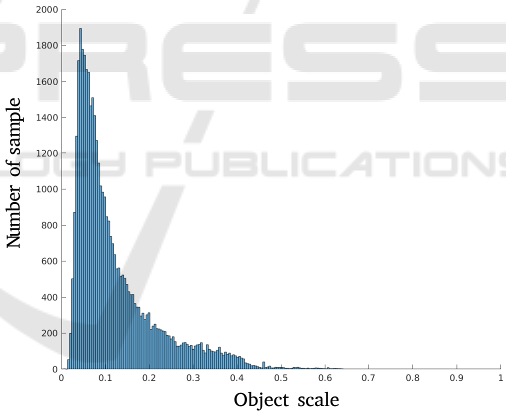

Figure 1: Object scale histogram of KITTI, object scales are depicted as the root square of object area divided by the image size. The histogram bin size is 0.005.

Those selected anchors are the discretized samples from the continuous space of box scale and aspect ra-

Li, X., Kwok, N., Guivant, J., Narula, K., Li, R. and Wu, H.
Detection of Imaged Objects with Estimated Scales.
DOI: 10.5220/0007353600390047
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 39-47
ISBN: 978-989-758-354-4

tio with a set of fixed bins. Although using a larger number of anchors can better represent the continuous space, the corresponding increases in complexity make regressors difficult to learn scale-invariant features and deep learning models expensive to train. On the contrary, it is difficult to find a fit for objects with only a few anchors. Given size variations in the dataset, the number of anchor scales and their aspect ratios are important hyperparameters. The detection performance is usually susceptible to improper settings of these hyperparameters. In order to find out how these hyper-parameters affect final performance and how to select the optimal ones, we designed multiple controlled experiments on KITTI benchmark dataset (Geiger et al., 2012). We found that the set of designed anchors for object detection should adequately cover the continuous space of object scales and aspect ratios, and simultaneously keeps a minimal number of anchors. Subject to such a contradictory criterion, the selection of the optimal hyperparameters is a challenging task.

In order to satisfy both the requirements for designing anchors, we iteratively explore whether it is possible to estimate the continuous object scale that can cover the whole scale space instead of pre-defining multiple discrete anchors with fixed scales and aspect ratios. To acquire continuous scales, we utilize the distance of the object with respect to the sensor to estimate the coarse scale of the detected objects. A corresponding detection framework based on CNN is also proposed to validate how estimated scales can improve detection performance. To validate our method, we conducted extensive evaluations on the KITTI benchmark with a fine-grained analysis. Our proposed method can outperform state-of-the-art with predefined anchors while using the same CNN backbone, especially on detecting difficult objects. The proposed method can also be assembled into multi-object detection algorithms with complex detection frameworks (Ren et al., 2017) (Dai et al., 2017). Our code is open-source and freely available on github[1].

There are three main works in our paper as following.

- Designed controlled experiments to answer the question of how the number of predefined anchors affects the detection performance.
- Proposed a detection method based on estimated size of objects.
- Conducted massive experiments on the KITTI benchmark to validate our method.

The rest of the paper is organized as follows. Section 2 is an introduction to the related work, followed by Section 3 which illustrates how multiple anchors affect detection accuracy. Our proposed detection method is presented in Section 4. Experiments of the proposed method are given in Section 5. Section 6 concludes our main work and summarizes the contributions of this paper.

2 RELATED WORK

2.1 Mutli-size Detectors

Scale-space theory (Lindeberg, 1990) is a vital and fundamental theory in signal processing, and significant research has been devoted to this field. Multi-size detectors (Ren et al., 2015) are usually utilized to address multi-scale of objects. Multi-size detectors take one-size input and apply multi-size detectors to detect their corresponding objects (Ren et al., 2015; Lin et al., 2016; He et al., 2016). The Faster-RCNN (Ren et al., 2015) implements detection on the final feature map using 9 different anchors with 3 different sizes and 3 different aspect ratios. Each anchor can represent one size detector that finds objects with similar sizes. However, the final feature map is usually at a low resolution with high-level semantic information, which makes small objects detection very challenging. To improve small objects detection, the feature pyramid network is proposed to propagate high-level semantic information in deeper layers back to shallower layers with high-resolution maps; Small objects are mainly detected from fused shallower feature maps (Lin et al., 2016). The recurrent rolling network extends feature pyramid network by using a recurrent neural network to fuse feature maps from different layers and integrate context information (Ren et al., 2017). However, even pyramid feature maps may not be useful for detecting small-size object since high-level information does not contain the semantic feature on small objects. Therefore, to increase the final feature map resolution, by upsampling the image, has become the most common practical technique to detect small objects instead of building an image pyramid (He et al., 2016).

Some methods are proposed to change the size of the receptive field to accommodate multi-scale objects, which includes the dilated and deformable convolutional network (Dai et al., 2017; Yu and Koltun, 2015). Transformation parameters are learned by a network, similar to STN (Spatial Transformer Networks) (Jaderberg et al., 2015), by building the STN to perform an affine transformation on input fe-

[1] Main open-source code can be found: https://github.com/Benzlxs/Object_detection_estimated_sclales

atures for the classification task. These methods require the CNN backbone in the detection model to learn scale-invariant features and utilize one detection head to do classification and regression on all scale objects together. Contrary to this, two sub-networks are used to predict multi-scale objects independently (Li et al., 2015), and every sub-network handles object within its scale range. Such kind of independent prediction is made at layers of different resolutions. MSC-CNN (Cai et al., 2016) claims that receptive filed of CNN should be consistent with the size of objects, and design algorithms to detect different scale objects on feature maps of multiple resolutions. Every resolution represents one receptive field size on which objects within certain scale ranges are found. Deep learning models can manage to learn scale-invariant representations when the scales variation is not large but still suffers from extremely small and large scale variations of objects. On the other hand, scale-specific methods with an image pyramid can handle all scales well. To combine the advantages of scale-variant and scale-invariant methods, coarse image pyramids are built (Singh and Davis, 2017; Hu and Ramanan, 2016) from 3 levels as input. Every level in the pyramid corresponds to one detection model which is just required to deal with a fixed range of scales instead of all object scales in the training dataset. Such methods usually achieve the state-of-the-art performance where each object detector focuses only on objects within certain scale ranges.

All multi-scale detection methods mentioned in this section rely on manually predefining anchors. Every anchor is designed to cover one size of target objects and its region in the image will be pooled into a fixed shape as scale-invariant feature representations, so hyperparameters of these anchors are vital for detection performance. Instead of heavily relying on tuning the hyperparameters of anchors, we proposed a detection method in this work that uses estimated anchors to conver all continuous scales and aspect ratios of detected objects.

2.2 Depth based Detection Methods

Depth information is already available in many applications. To take advantage of the depth data for object detection, common methods compress 3D information into 2D information on which the CNN is applied directly. 3D point cloud is encoded into a cylindrical-like image where every valid element carries 2-channel data so that the 2D image can inherit all the information from the 3D point cloud (Li et al., 2016). The 2D CNN is then used to process such kind of 2D image to detect the object. However, such kind of data representation fails to achieve decent detection performance. To improve accuracy further, The work in (Chen et al., 2016) convert 3D point cloud into birdview representation which includes many height maps, a density map, and an intensity map; based on this representations, the CNN is then applied to propose 3D object candidates. These methods actually regard point clouds as one extra image channel and are similar with ideas used to deal with RGBD data (Gupta et al., 2014). In this work, we employ raw depth data to estimate coarse scale in an efficient and simple manner instead of encoding depth information into some feature maps. Estimated scales can remove the need for manually predefining anchors and are able to cover all possible continuous scales.

3 MULTIPLE ANCHORS

Predefined multi-scale anchors are crucial to the performance of the detection method. This section describes how these anchors affect detection results and how to select optimal predefined anchors.

3.1 Scales and Aspect Ratios

Predefined anchors are a set of boxes with different sizes $\{(w_i, h_i) : 1 \leqslant i \leqslant N\}$ in which w_i and h_i are width and height of the box respectively, and N is the number of predefined anchors. The size of an anchor is defined by its scale and aspect ratio that denote the anchor area and the ratio between box height and width. At each sliding-window location (x_j, y_j), N proposal boxes $\{(x_j, y_j, w_i, h_i) : 1 \leqslant i \leqslant N, 1 \leqslant j \leqslant W \times H\}$, where W and H are the size of feature map will be predicted by the network. Therefore, $H \times W \times N$ proposals will be generated by the region proposal network. Each anchor size, (w_i, h_i), can be calculated according to equation. 1 and 2.

$$w_i = \frac{S_k}{\sqrt{R_j}} \tag{1}$$

$$h_i = S_k \times \sqrt{R_j} \tag{2}$$

Scales $\{S_k : 1 \leqslant k \leqslant K\}$ are another set of predefined parameters. Every S_k represents one scale of the anchor and can be calculated as the root square of the anchor region in the image coordinate. $\{R_j : 1 \leqslant j \leqslant J\}$ is the set of aspect ratios, and J is the number of predefined aspect ratio for every scale. As shown in Equ. 1 and 2, every scale will be expanded to J anchors, which have the same area of different ratios between height and width, in order to handle various poses of the same object in the image. The number

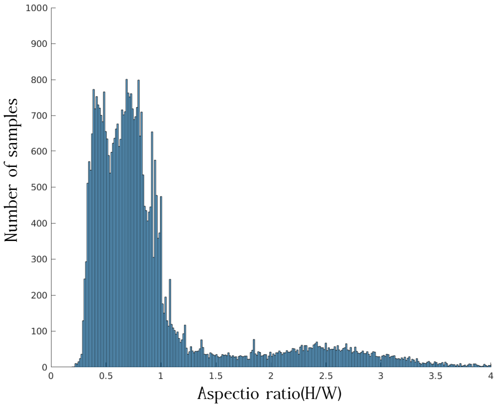

Figure 2: Object aspect ratio histogram of KITTI dataset. Aspect ratios are calculated by dividing H by W of the box size. Histogram bin size is 0.001.

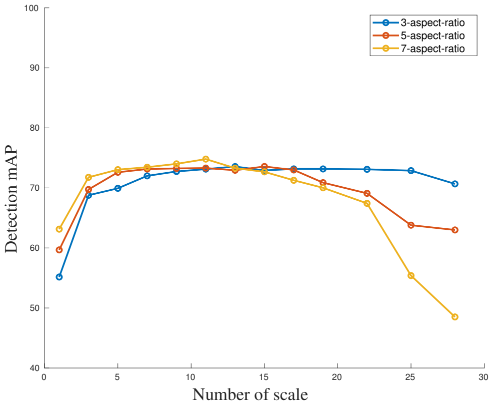

Figure 3: Comparison of detection mAP against different combinations of scales and aspect ratios. The Detection mAP is the mean average detection precision of car, cyclist and pedestrian of all degree of difficulties.

of anchors N is $K \times J$, and $H \times W \times K \times J$ proposal boxes will be generated.

3.2 Anchor Selection

Variables S_k and R_j are very important user-defined hyperparameters. To understand how these hyperparameters affect the final detection performance, we conduct controlled experiments on KITTI dataset (Geiger et al., 2012). The object scale and aspect ratio histograms can be found in Fig. 1 and 2. The domain of scale and aspect ratio variation are $[0.0, 0.64]$ and $[0.25, 10.56]$ respectively. In our experiments, we uniformly sample different scales in $[0.02, 0.54]$ instead of the full range, because the 99.5% of the objects in the training data reside in this range and the full range with extremely large and small scale will bring very heavy experimental workload. Similarly, the chosen sampling range of aspect ratios is between $[0.25, 4.0]$, in which 99.5% of all the objects can be covered. The Faster-RCNN (Ren et al., 2015) with VGG16 network backbone (Simonyan and Zisserman, 2014) is employed in our experiments[1]. Results are plotted in Fig. 3.

From the Fig. 3, we find that three anchors settings, $(K = 3, R = 25)$, $(K = 5, R = 15)$ and $(K = 7, R = 11)$, can respectively achieve the best performance among their anchor sets with the same aspect ratios (K) but different scales (R). The interesting phenomenon is that the total number of anchors in these three sets is very similar and close to 75. When the anchor number is small, less than 75, the detection accuracy is proportional to the anchor number. It is because more anchors can help to better cover the continuous domain of all object sizes, and every anchor just needs to handle a small range of size variation which reduces learning difficulties of the region proposal network. If the number of anchors is too large, the detection accuracy will be inversely proportional to the anchor number. The reason is that a larger number of possible predefined boxes will lead to a significant imbalance between the positive and negative examples, and easy negative examples can overwhelm the training of region proposal network which may in turn lead to degenerate models. Another reason is that the number of convolution prediction filters increases linearly with the number of anchors; together with the fixed number of training labels, a large proportion of the prediction filters are unable to get enough training. Besides, as the number of predefined anchors increases, the computational and memory cost will also increase and more training time will be required. It is concluded that predefined anchors should be able to cover all possible scales of objects, but their numbers should be as small as possible.

It is difficult to predefine anchors that satisfy both the requirements mentioned above. The common way is to conduct trial-and-error experiments to find the optimal number for one dataset, which can be computationally demanding and time-consuming. Therefore, we introduce the depth information into the network in order to reduce the required number of anchors. The depth information is utilized to estimate the coarse size of the objects that ca cover the continuous scale and aspect ratio ranges at the same time. For every location, only a few anchor boxes are generated according to the number of target object types and real size variations.

[1] Experimental code and results can be found https://github.com/Benzlxs/TFFRCNN

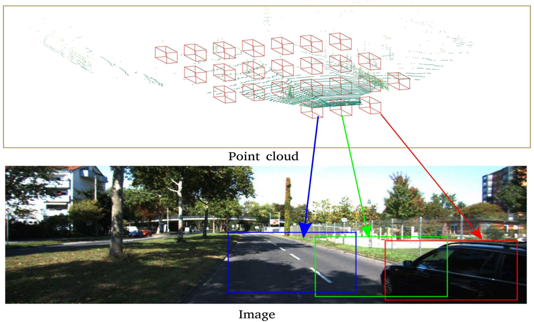

Figure 4: Examples of projection from P_{3D} to P_{2D}. The top image is 3D bounding boxes of P_{3D} in point cloud coordinate, and the bottom one is 2D bounding boxes of their corresponding P_{2D}.

4 DETECTION WITH ESTIMATED ANCHOR

In many real-world applications, like the driverless car and indoor service robot, depth information is there for other purposes and can be obtained easily by depth sensors such as stereo camera or Lidar. The depth information can decide the object location in 3D space, and coarse 3D size of detected objects can be predefined, for example, $4.2 \times 1.8 \times 1.6$ m^3 can coarsely represent the 3D bounding box of a sedan. Given calibration parameters between the depth sensor and camera, the 3D bounding box can be projected into a 2D region in image coordinate, and the 2D region can be used to represent a coarse scale of the detected object. Thousands of projected regions will be generated and used to crop object features from light feature map compressed from *conv5_3* of VGG16. RoI (Region of Interest) pooling is implemented to convert every region feature into fixed shape as scale-invariant feature representations. The scale-invariant network consisting of two fully connected layers is used to do coarse classification and regression to find valuable proposals. It has the similar function with the region proposal network. Selected regions are then used to do cropping and RoI pooling on the thick feature map *conv5_3* for the second-stage classification and regression which includes three-layers of the fully connected network.

Based on the observation of the process described in the previous section, the proposed detection method will estimate a coarse region size of the object, (w_i, h_i), and applies it to enhance the performance of object detection. In this section, inference about object size and how to integrate scale into the CNN framework for object detection will be presented.

4.1 Anchor Estimation

The distance between the observer and objects can almost tell their scales in the image, so scale for different objects can be coarsely estimated from their depth map. Without predefining a set of possible anchors, we utilize depth to estimate the coarse size of target objects. Given image coordinates and depth data, 3D point cloud can be generated using simple affine transformation. Some noisy points representing the sky region will be removed. Thousands of 3D bounding boxes that can represent the coarse 3D size of detected objects are employed to wrap all selected points. These 3D bounding boxes are projected back into image coordinate to find the coarse sizes of the anchors.

We validate our method using KITTI dataset where point cloud data is available. The 3D space range for sampling 3D bounding boxes are $[0, 70]$ m along x axis and $[-40, 40]$ m along y axis under Lidar coordinate convention. We slide these 3D bounding boxes along x and y axes with 0.4 m intervals on the road plane which can easily be estimated by random sample consensus. 3D boxes are filtered out by removing those in which the number of points is less than 4. Then, 8 corner points P_{3D} of remaining boxes will be projected back to points P_{2D} in image coordinates according to Equ.3, where R is the rotation matrix and T is the transformation matrix. Calibration parameters are provided in the KITTI dataset.

$$P_{2D} = R \times P_{3D} + T \tag{3}$$

Projected boxes in the image coordinate can be employed as the coarse size of the anchor, such as in Fig. 4. Instead of sliding a fixed set of anchors over every point in the feature map as in Faster-RCNN, we just generate a smaller number of region boxes on interesting points, about $40,000$ of them, while Faster-RCNN with 15 anchors will generate about $200,000$ region boxes.

4.2 Detection Network

CNN has been widely and successfully used to extract features from an image for object detection and classification, and it is adopted to generate feature maps in our detection method. Based on feature maps, fully connected layers are employed to do classification and regression. The method pipeline is depicted in Fig. 5.

The VGG16 (Simonyan and Zisserman, 2014) are used to generate feature maps from the input image. Numerous 2D boxes N_{roi} will be estimated from depth information according to the method described in the

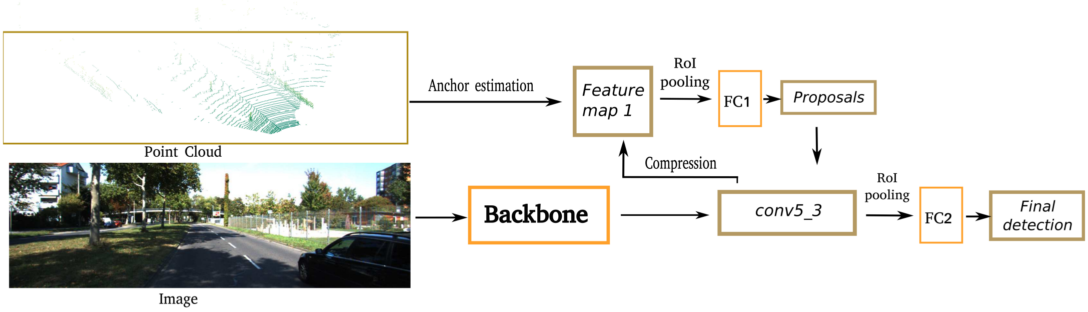

Figure 5: The proposed detection framework. Backbone is pretrained VGG16.

previous section. Usually, the last layer of feature maps has a large number of channels (512 in VGG16). If we crop features from the estimated boxes directly on a wide feature map, the large number of scale boxes will be memory-intensive and computationally expensive. Therefore, by using 1×1 convolutional layers, we compress the last layer of feature maps, the *conv5_3* of VGG16, into a thin feature map, *Feature map 1* in the Fig.5, with 32 channels of the same width and height. Compressed feature maps are designed to select primary features for region proposal network. Every estimated region box will be used to crop its corresponding feature on *Feature map 1* into $5 \times 5 \times 32$ scale-invariant features, also called ROI pooling operation. Then, features are connected with light-weight $FC1$ which consists of two fully connected layers with 512 neurons per layer. The $FC1$ plays the same role as the region proposal network in the Faster-RCNN (Ren et al., 2015); it will perform coarse classification with output $N_{roi} \times 2(foreground/background)$ and regression of boxes with output $N_{roi} \times 4$. N_{rpn} will be selected from N_{roi} as a mini-batch to train the $FC1$ network. All predicted boxes are post-processed by non-maximum suppression (NMS) to select the top N_{prop} proposals. Since the number of proposals is small, selected proposals are used to do RoI pooling directly on the last layer of the backbone *conv5_3*. Then, $7 \times 7 \times N_{prop}$ features are produced and connected to $FC2$ which contains three layers of the fully connected layers with 2048 neurons each. The $FC2$ will generate final detection results, $N_{prop} \times C$ (the number of category) for classification and $N_{prop} \times 4$ for bounding box regression.

The loss function is defined in Equ. 4, 5 and 6, which consists of two terms, L_{rpn} and L_{2s}. L_{rpn} is used to train the region proposal network, $FC1$, while L_{2s} is for the second-stage refineing network, $FC2$. Loss of the whole detection network is the sum of these two term with weighting parameter α. L_{cls} is the cross entropy between predicted categories and corresponding labels, and L_{reg} is the $smooth_{L1}$ loss function (Ren et al., 2015). p^i and t^i are the outputs of classification and regression network; p^{i*} and t^{i*} are their corresponding ground truths. A set of proposals N_{rpn}, 512 in our configuration, is selected from N_{roi}. When calculating the regression loss, only positive labels will be counted ($\mathbb{1}$ indicates that there is a positive label). N^{pos}_{rpn} represents the number of positive labels, and N_{prop}, 1024 in our configuration, is the number of samples we selected for the second-stage network $FC2$. These hyperparameters are selected according to numerous experiments in the next section. The AdamOptimizer (Kingma and Ba, 2014) is employed to train our network end-to-end.

$$L_{all} = L_{rpn} + \alpha L_{2s} \tag{4}$$

$$\begin{aligned} L_{rpn} = & \frac{1}{N_{rpn}} \sum_i L_{cls}(p^i_{rpn}, p^{i*}_{rpn}) \\ & + \frac{1}{N^{pos}_{rpn}} \sum_i \mathbb{1}^{obj}_i L_{reg}(t^i_{rpn}, t^{i*}_{rpn}) \end{aligned} \tag{5}$$

$$\begin{aligned} L_{2s} = & \frac{1}{N_{prop}} \sum_j L_{cls}(p^j_{2s}, p^{j*}_{2s}) \\ & + \frac{1}{N^{pos}_{prop}} \sum_j \mathbb{1}^{obj}_i L_{reg}(t^j_{2s}, t^{j*}_{2s}) \end{aligned} \tag{6}$$

5 EXPERIMENTS

The proposed method is evaluated on the KITTI benchmark, which includes 7481 training and 7518 testing sets of high-resolution images. LIDAR laser data is also available. Since the ground truth of the testing dataset is not publicly available, we split the training dataset in the 3:1 ratio for training and validation respectively. These are then used to conduct comparative experiments for the purpose of deducing how the hyperparameters affect the detection performance. The 0.7 IoU threshold for the car and 0.5 IoU threshold both for pedestrian and cyclist are used to calculate mean Average Precision (mAP). Lastly, the adequately-trained network model with optimal hyperparameters is deployed to process the testing dataset and the detection result is submitted to the bench-

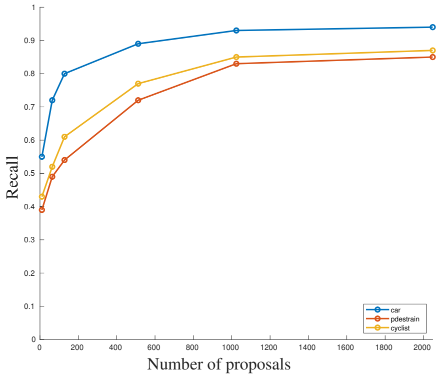

Figure 6: Proposal recall on the KITTI validation set for three classes.

mark site to compare against the Faster-RCNN method.

Table 1: Experiments on how to select predefined 3D bounding boxes, the detection performance is mAP(%) of easy, moderate and hard categories. The definition of easy, moderate and hard objects can be found in the benchmark (Geiger et al., 2012).

K	orientation	$\sim N_{roi}$	car	pedestrian	cyclist
1	[0]	5,000	76.48	68.12	60.78
1	$[0, \frac{\pi}{2}]$	10,000	85.64	71.29	70.36
1	$[0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$	20,000	89.12	72.01	74.51
2	[0]	10,000	80.65	71.85	63.62
2	$[0, \frac{\pi}{2}]$	20,000	88.23	73.96	74.82
2	$[0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$	40,000	**91.68**	74.00	76.45
3	[0]	15,000	81.87	72.93	63.80
3	$[0, \frac{\pi}{2}]$	30,000	90.03	74.03	74.64
3	$[0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$	60,000	91.51	**74.09**	**76.51**

For data augmentation, horizontally flipping and color jittering are used to increase the number of training dataset. The data is augmented by a factor of three such that its size is three times as big as the size of the original dataset. The whole detection network is trained in an end-to-end manner. The AdamOptimizer(Kingma and Ba, 2014) is configured with an initial learning rate of 0.0005 and exponential decay factor of 0.6 for every 40,000 iterations.

5.1 Hyperparamters Selection on the Validation Set

The number of estimated anchors N_{roi} and the number of proposals N_{prop}, are important hyperparameters for our detection framework. For the number of estimated anchors N_{roi}, it consists of valid non-road point cloud and predefined 3D template bounding boxes of different sizes and orientations. As for box sizes, we use the k-means clustering method to do clustering on 3D bounding box size of all training labels and calculate average box size of every cluster. For example, if we just use one size of the 3D bounding box for the car and setting k=1, the clustered size result is $[3.884(length), 1.629(width), 1.526(height)]$ m. While if we use two sizes of 3D bounding box for car and setting k=2, the clustered sizes are $[3.539, 1.599, 1.506]$ m and $[4.229, 1.658, 1.546]$ m. Experimental results are shown in Table 1, from which we can find that increasing the orientation and size (k) number will improve detection performance until N_{roi} is large, like over $\sim 30,000$, while large N_{roi} will make training much slower and consume more memory. Considering the tradeoff between efficiency and accuracy, we finally select size clusters k=2 and orientations $[0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$ for car; k=1 and $[0, \frac{\pi}{2}]$ for pedestrian; and k=2 and $[0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}]$ for cyclist in our framework.

For the number of proposals N_{prop}, we output all the proposals from region proposal network and do NMS with the IoU threshold of 0.8. The top N_{prop} proposals are saved to calculate the recall. The experimental results are shown in Fig 6, from which we can find that when N_{prop} the is over 1024, the recall barely increases. Since smaller N_{prop} consumes less memory and speeds up the training procedure, N_{prop} of 1024 is chosen for the experiments.

5.2 Evaluation on the Test Set

Results[2] are shown in Table 2, Fig 7 and 8. We conduct comparisons with the Faster-RCNN, for both of our method and Faster-RCNN employ the same VGG16 backbone (Simonyan and Zisserman, 2014), but different methods to determine the object scale and aspect ratio. On the KITTI benchmark, results of Faster-RCNN with 70 anchors are available for all objects, while Faster-RCNN with 9 anchors is only evaluated on car detection. For car detection, results of Faster-RCNN with 70 and 9 anchors are compared with our method. We can find that increasing the number of anchors can significantly enhance the detection accuracy. The detection performance of the Faster-RCNN in detecting the car objects peaks when the anchor number reaches 70. Our proposed method with the estimated scales, on the other hand, can gain similar detection accuracy on the easy category. However, on moderate and difficult categories, estimated scales can help to find the anchor size closest to the detected object, which reduces the difficulties of regressing locations of moderate and difficult objects. For pedestrian and cyclist detection, only

[2] Testing results can be found in the KITTI object detection leaderboard http://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=2d

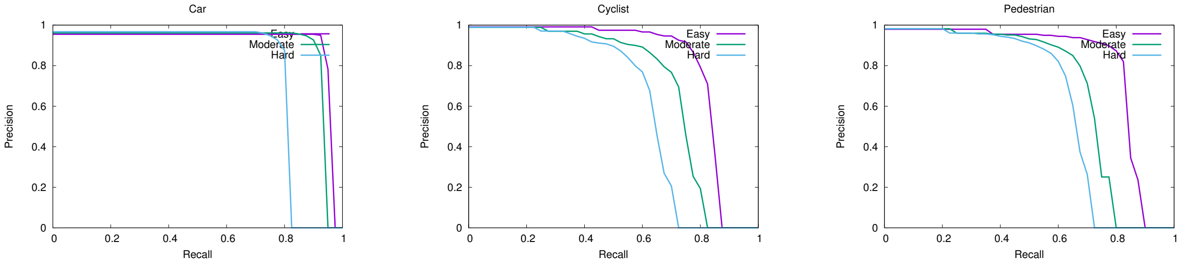

Figure 7: Precision recall curves for object detection on KITTI test set.

Table 2: Detailed result for object detection on KITTI test set, given in terms of average precision (AP).

Object	Method	mAP	Easy	Moderate	Hard
Car	Proposed method	**84.08 %**	86.82%	**87.10%**	**78.32%**
	Faster-RCNN (70 anchors)	79.01 %	**87.9 %**	79.11 %	70.19%
	Faster-RCNN (9 anchors)	54.72%	62.31%	56.58 %	45.27 %
Cyclist	Proposed method	**69.88 %**	**78.51 %**	**69.80 %**	**61.32 %**
	Faster-RCNN (70 anchors)	63.22 %	71.41 %	62.81 %	55.44 %
Pedestrian	Proposed method	**69.16 %**	77.95 %	**67.25 %**	**62.28%**
	Faster-RCNN (70 anchors)	68.48 %	**78.35 %**	65.91 %	61.19 %

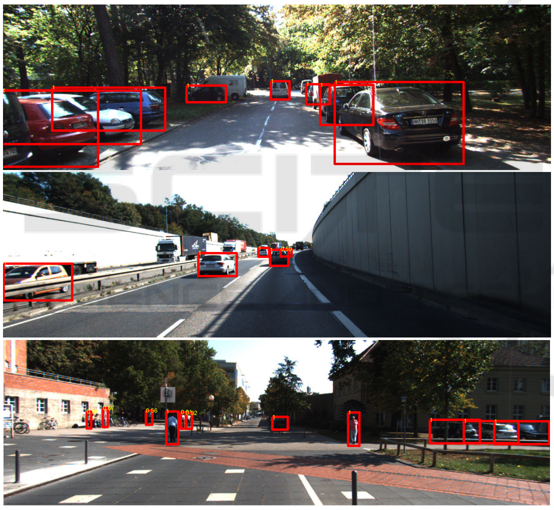

Figure 8: Some samples of detection results on KITTI test set.

Faster-RCNN with 70 anchors is evaluated on the testing dataset and compared with our method. There are 6.66 % and 0.68 % accuracy gains in the mAP are achieved for the cyclist and the pedestrian category respectively. One explanation is that cyclists have much larger variation in scale and aspect ratio than the pedestrians, therefore, estimated anchors brings more advantages than predefined anchors for the cyclists.

6 CONCLUSION

Inexact multiple scales and aspect ratios are obstacles in the detection of all objects in the image. Classical methods use predefined anchors to solve this problem, but the discrete and sparse anchor samples are not able to cover all scales and aspect ratios.The simple solution of increasing the number of anchors is infeasible since a large number of anchors hinders the training of CNN. To find the optimal size of anchors, we proposed to use depth information to estimate a coarse scale of objects and designed the corresponding detection framework. The proposed method can achieve a significant improvement compared with the method using a similar backbone and predefined anchors. However, using the depth information to estimate coarse anchor sizes makes it unfeasible for many applications where the depth information is not available. Currently, there are researches indicating the neural network has been successfully used to predict object depths from a single image. Our future research will endeavor to estimate estimate coarse and continuous anchor sizes from a single image.

REFERENCES

Cai, Z., Fan, Q., Feris, R. S., and Vasconcelos, N. (2016). A unified multi-scale deep convolutional neural network for fast object detection. *CoRR*, abs/1607.07155.

Chen, X., Ma, H., Wan, J., Li, B., and Xia, T. (2016). Multi-view 3d object detection network for autonomous driving. *arXiv preprint arXiv:1611.07759*.

Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., and Wei, Y. (2017). Deformable convolutional networks. *CoRR*, abs/1703.06211.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and*

Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 886–893. IEEE.

Felzenszwalb, P., McAllester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE.

Geiger, A., Lenz, P., and Urtasun, R. (2012). Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE.

Gupta, S., Girshick, R. B., Arbelaez, P., and Malik, J. (2014). Learning rich features from RGB-D images for object detection and segmentation. *CoRR*, abs/1407.5736.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. B. (2017). Mask R-CNN. *CoRR*, abs/1703.06870.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hu, P. and Ramanan, D. (2016). Finding tiny faces. *CoRR*, abs/1612.04402.

Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. (2015). Spatial transformer networks. *CoRR*, abs/1506.02025.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Li, B., Zhang, T., and Xia, T. (2016). Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*.

Li, J., Liang, X., Shen, S., Xu, T., and Yan, S. (2015). Scale-aware fast R-CNN for pedestrian detection. *CVRR*, abs/1510.08160.

Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. J. (2016). Feature pyramid networks for object detection. *CoRR*, abs/1612.03144.

Lindeberg, T. (1990). Scale-space for discrete signals. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(3):234–254.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., and Berg, A. C. (2015). SSD: single shot multibox detector. *CoRR*, abs/1512.02325.

Redmon, J., Divvala, S. K., Girshick, R. B., and Farhadi, A. (2015). You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640.

Ren, J., Chen, X., Liu, J., Sun, W., Pang, J., Yan, Q., Tai, Y.-W., and Xu, L. (2017). Accurate single stage detector using recurrent rolling convolution. *arXiv preprint arXiv:1704.05776*.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556.

Singh, B. and Davis, L. S. (2017). An analysis of scale invariance in object detection - SNIP. *CoRR*, abs/1711.08189.

Yu, F. and Koltun, V. (2015). Multi-scale context aggregation by dilated convolutions. *CoRR*, abs/1511.07122.