Paper Lists

Dapeng Feng

Contents

1. ARXIV **3**
- 1.1. 2019 . . . 3
  - 1.1.1 End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds [29] . . . 3
  - 1.1.2 Objects as points [28] . . . 3
  - 1.1.3 PU-GCN: Point Cloud Upsampling using Graph Convolutional Networks [17] . . . 4
  - 1.1.4 StarNet: Targeted Computation for Object Detection in Point Clouds [14] 4
- 1.2. 2020 . . . 6
  - 1.2.1 AutoML-Zero: Evolving Machine Learning Algorithms From Scratch [19] . . . 6
  - 1.2.2 Are Labels Necessary for Neural Architecture Search [13] . . . 7
  - 1.2.3 Inferring Spatial Uncertainty in Object Detection [27] . . . 7
  - 1.2.4 Improved Baselines with Momentum Contrastive Learning [3] . . . 9
  - 1.2.5 Range Conditioned Dilated Convolutions for Scale Invariant 3D Object Detection [1] . . . 9

2. CVPR **9**
- 2.1. 2012 . . . 9
  - 2.1.1 Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite [4] . . . 9
- 2.2. 2016 . . . 10
  - 2.2.1 Deep Residual Learning for Image Recognition [8] . . . 10
- 2.3. 2017 . . . 10
  - 2.3.1 Feature Pyramid Networks for Object Detection [11] . . . 10
  - 2.3.2 PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation [15] . . . 11
- 2.4. 2019 . . . 13
  - 2.4.1 PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud [23] . . . 13
  - 2.4.2 PointPillars: Fast Encoders for Object Detection From Point Clouds [10] 14
- 2.5. 2020 . . . 14
  - 2.5.1 Designing Network Design Spaces [18] . . . 14
  - 2.5.2 Momentum Contrast for Unsupervised Visual Representation Learning [7] . . . 15
  - 2.5.3 Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud [24] . . . 16
  - 2.5.4 PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection [22] . . . 16
  - 2.5.5 RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds [9] . . . 17
  - 2.5.6 Scalability in Perception for Autonomous Driving: Waymo Open Dataset [25] . . . 18

3. ECCV **19**
- 3.1. 2014 . . . 19
  - 3.1.1 Microsoft COCO: Common Objects in Context [12] . . . 19

4. ICCV **19**
- 4.1. 2015 . . . 19
  - 4.1.1 Fast R-CNN [5] . . . 19
- 4.2. 2019 . . . 20
  - 4.2.1 M3D-RPN: Monocular 3D Region Proposal Network for Object Detection [2] . . . 20

5. IJCV **20**
- 5.1. 2015 . . . 20
  - 5.1.1 ImageNet Large Scale Visual Recognition Challenge [21] . . . 20

6. NIPS **21**
- 6.1. 2017 . . . 21

6.1.1 PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space [16] 21

7. T-PAMI **22**
7.1. 2016 22
7.1.1 Region-Based Convolutional Networks for Accurate Object Detection and Segmentation [6] 22
7.2. 2017 22
7.2.1 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [20] 22
7.3. 2019 23
7.3.1 The ApolloScape Open Dataset for Autonomous Driving and its Application [26] 23

1. ARXIV

1.1. 2019

1.1.1 End-to-End Multi-View Fusion for 3D Object Detection in LiDAR Point Clouds [29]

Abract: Recent work on 3D object detection advocates point cloud voxelization in birds-eye view, where objects preserve their physical dimensions and are naturally separable. When represented in this view, however, point clouds are sparse and have highly variable point density, which may cause detectors difficulties in detecting distant or small objects (pedestrians, traffic signs, etc.). On the other hand, perspective view provides dense observations, which could allow more favorable feature encoding for such cases. In this paper, we aim to synergize the birds-eye view and the perspective view and propose a novel end-to-end multi-view fusion (MVF) algorithm, which can effectively learn to utilize the complementary information from both. Specifically, we introduce dynamic voxelization, which has four merits compared to existing voxelization methods, i) removing the need of pre-allocating a tensor with fixed size; ii) overcoming the information loss due to stochastic point/voxel dropout; iii) yielding deterministic voxel embeddings and more stable detection outcomes; iv) establishing the bi-directional relationship between points and voxels, which potentially lays a natural foundation for cross-view feature fusion. By employing dynamic voxelization, the proposed feature fusion architecture enables each point to learn to fuse context information from different views. MVF operates on points and can be naturally extended to other approaches using LiDAR point clouds. We evaluate our MVF model extensively on the newly released Waymo Open Dataset [25] and on the KITTI [4] dataset and demonstrate that it significantly improves detection accuracy over the comparable single-view PointPillars [10] baseline.

Contribution:

1. Synergize the bird's-eye view and the perspective view to learn the effective complementary information from both.
2. Introduce dynamic voxelization, which has four advantages:
   (a) Eliminate the need to sample a predefined number of points per voxel and pad voxel to predefined size.
   (b) Reduce the information loss due to stochastic point/voxel dropout.
   (c) Yield deterministic voxel embedding
   (d) Establish the bi-directional relationship between points and voxels, which serves as a natural foundation for cross-view feature fusion.

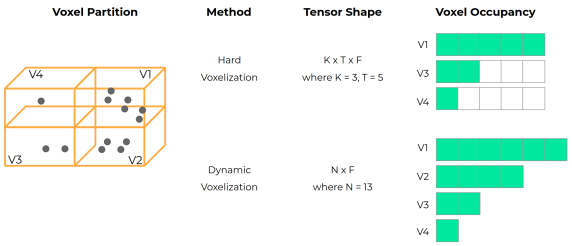


Figure 1: Illustration of the differences between *hard voxelization* and *dynamic voxelization*. The space is devided into four voxels, indexed as $\mathbf{v}_1$, $\mathbf{v}_2$, $\mathbf{v}_3$, $\mathbf{v}_4$, which contain 6, 4, 2 and 1 points respectively. *hard voxelization* drops one point in $\mathbf{v}_1$ and misses $\mathbf{v}_2$, with $15F$ memory usage, whereas *dynamic voxelization* captures all four voxels with optimal memory usage $13F$.

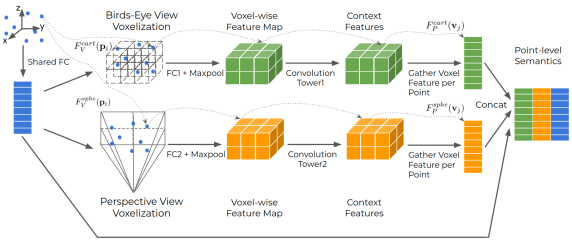
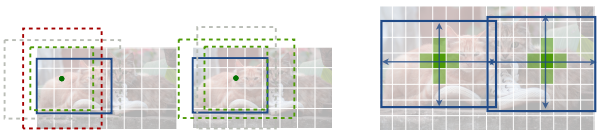


Figure 2: Multi-View Fusion (MVF) Network Architecture. Given a raw LiDAR point cloud as input, the proposed MVF first embeds each point into a high dimensional feature space via one fully connected (FC) layer, which is shared for different views. Then, it applies *dynamic voxelization* in the birds-eye view and the perspective view respectively and establishes the bi-directional mapping ($F_V^*(\mathbf{p}_i)$ and $F_P^*(\mathbf{v}_j)$) between points and voxels therein, where $* \in \{cart, sphe\}$. Next, in each view, it employs one additional FC layer to learn view-dependent features, and by referencing $F_V^*(\mathbf{p}_i)$ it aggregates voxel information via Max Pooling. Over the voxel-wise feature map, it uses a convolution tower to further process context information within an enlarged receptive field, while still maintaining the same spatial resolution. Finally, based on $F_P^*(\mathbf{v}_j)$, it fuses features from three different sources for each point, *i.e.,* the corresponding voxel features from the birds-eye view and the perspective view as well as the corresponding point feature obtained via the shared FC.

Figures: Figure 1, Figure 2

1.1.2 Objects as points [28]

Abract: Detection identifies objects as axis-aligned boxes in an image. Most successful object detectors enumerate a nearly exhaustive list of potential object locations and classify each. This is wasteful, inefficient, and requires addi-



(a) Standard anchor based detection. Anchors count as positive with an overlap $IoU > 0.7$ to any object, negative with an overlap $IoU < 0.3$, or are ignored otherwise.

(b) Center point based detection. The center pixel is assigned to the object. Nearby points have a reduced negative loss. Object size is regressed.

Figure 3: Different between anchor-based detectors (a) and our center point detector (b). Best viewed on screen.

tional post-processing. In this paper, we take a different approach. We model an object as a single point — the center point of its bounding box. Our detector uses keypoint estimation to find center points and regresses to all other object properties, such as size, 3D location, orientation, and even pose. Our center point based approach, CenterNet, is end-to-end differentiable, simpler, faster, and more accurate than corresponding bounding box based detectors. CenterNet achieves the best speed-accuracy trade-off on the MS COCO [12] dataset, with 28.1% AP at 142 FPS, 37.4% AP at 52 FPS, and 45.1% AP with multi-scale testing at 1.4 FPS. We use the same approach to estimate 3D bounding box in the KITTI [4] benchmark and human pose on the COCO keypoint dataset. Our method performs competitively with sophisticated multi-stage methods and runs in real-time.

URL: https://github.com/xingyizhou/CenterNet

Figures: Figure 3

1.1.3 PU-GCN: Point Cloud Upsampling using Graph Convolutional Networks [17]

Abract: Upsampling sparse, noisy, and non-uniform point clouds is a challenging task. In this paper, we propose 3 novel point upsampling modules: Multi-branch GCN, Clone GCN, and NodeShuffle. Our modules use Graph Convolutional Networks (GCNs) to better encode local point information. Our upsampling modules are versatile and can be incorporated into any point cloud upsampling pipeline. We show how our 3 modules consistently improve state-of-the-art methods in all point upsampling metrics. We also propose a new multi-scale point feature extractor, called Inception DenseGCN. We modify current Inception GCN algorithms by introducing DenseGCN blocks. By aggregating data at multiple scales, our new feature extractor is more resilient to density changes along point cloud surfaces. We combine Inception DenseGCN with one of our upsampling modules (NodeShuffle) into a new point upsampling pipeline: PU-GCN. We show both qualitatively and quantitatively the advantages of PU-GCN against the state-of-the-art in terms of fine-grained upsampling quality and point cloud uniformity. The source code of this work is available at https://github.com/guochengqian/PU-GCN.

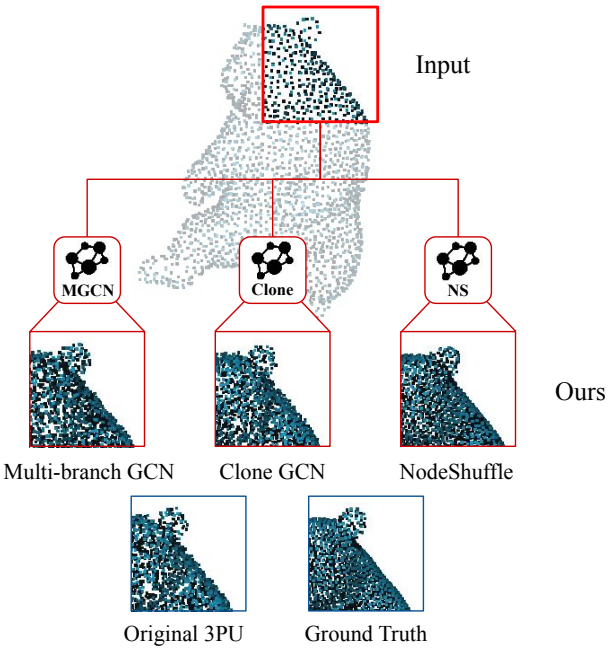


Figure 4: **Novel GCN based point upsampling modules.** We develop three new upsampling modules (*Multi-branch GCN*, *Clone GCN* and *NodeShuffle*) with Graph Convolutional Network (GCN) architecture. When integrated into the 3PU upsampling pipeline, they generate more uniform, dense point clouds with fine-grained details, as compared to state-of-the-art point upsampling techniques.

Contribution:

1. Three novel point cloud upsampling modules using graph convolutions: Multi-branch GCN, Clone GCN, and NodeShuffle.

2. Inception DenseGCN, a feature extraction block that encodes multi-scale information effectively and efficiently.

Figures: Figure 4, Figure 5, Figure 6, Figure 7, Figure 8, Figure 9

1.1.4 StarNet: Targeted Computation for Object Detection in Point Clouds [14]

Abract: LiDAR sensor systems provide high resolution spatial information about the environment for self-driving

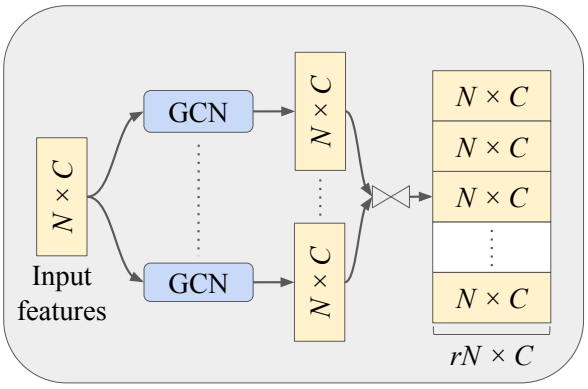


Figure 5: Multi-branch GCN

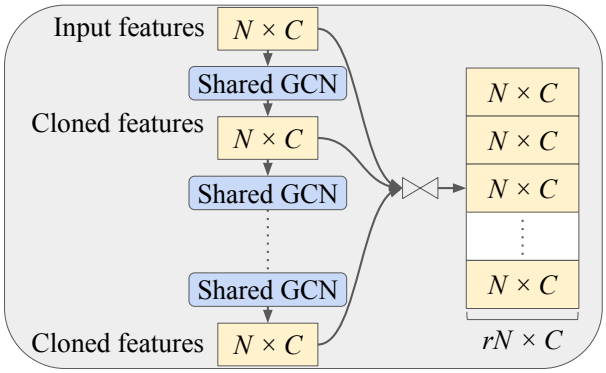


Figure 6: Clone GCN

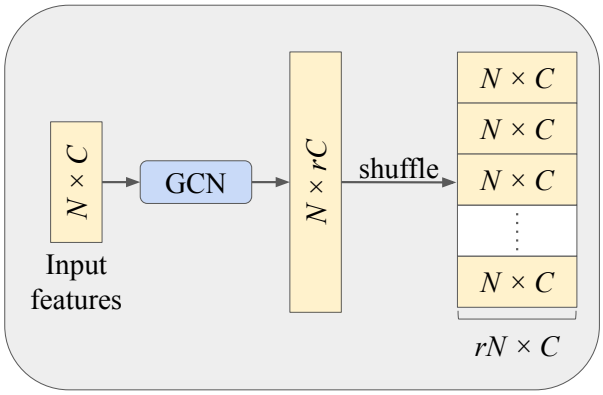


Figure 7: NodeShuffle

cars. Therefore, detecting objects from point clouds derived from LiDAR represents a critical problem. Previous work on object detection from LiDAR has emphasized repurposing convolutional approaches from traditional camera imagery. In this work, we present an object detection system designed specifically for point cloud data blending aspects of one-stage and two-stage systems. We observe that objects in point clouds are quite distinct from traditional camera images: objects are sparse and vary widely in location, but do not exhibit scale distortions observed in single camera perspective. These two observations suggest that simple and cheap data-driven object proposals to maximize spatial coverage or match the observed densities of point cloud data may suffice. This recognition paired with a local, non-convolutional, point-based network permits building an object detector for point clouds that may be trained only once, but adapted to different computational settings – targeted to different predictive priorities or spatial regions. We demonstrate this flexibility and the targeted detection strategies on both the KITTI [4] detection dataset as well as on the large-scale Waymo Open Dataset [25]. Furthermore, we find that a single network is competitive with other point cloud detectors across a range of computational budgets, while being more flexible to adapt to contextual priorities.

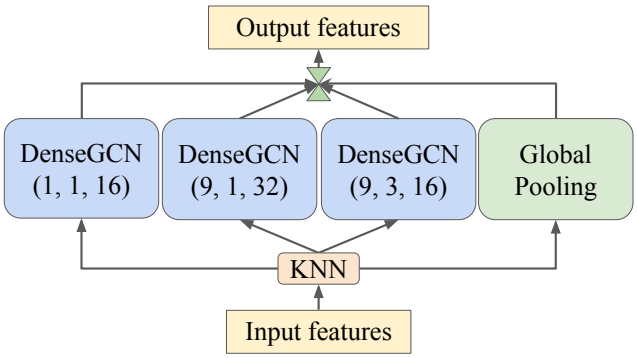


Figure 8: **Inception DenseGCN.** We use the parameters (k, d, c) to define a DenseGCN block. k is the number of neighbors (kernel size), d is the dilation rate, and c is the number of output channels. *KNN* is applied at the first layer to build the graph and the node neighborhoods.

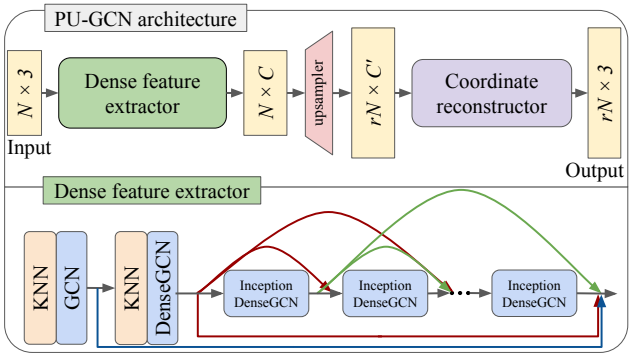


Figure 9: **PU-GCN architecture.** PU-GCN uses a dense feature extractor consisting of 1 or more densely connected Inception DenseGCN blocks, followed by the upsampler and coordinate reconstructor.

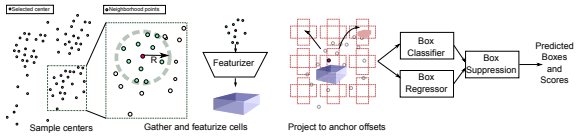


Figure 10: Given a sparse sampling of locations in the point cloud, the model extracts a small (random) subset of neighboring points. The model featurizes the point cloud, classifies the region, and regress bounding box parameters. The object location is predicted relative to the selected location and only uses local information. This setup ensures that each spatial location may be processed by the detector independently.

URL: http://github.com/tensorflow/lingvo
Contribution:

1. Introduce a flexible, local point-based object detector geared towards self-driving cars (SDC) perception. In the process the authors demonstrate that cheap proposals on point clouds, paired with a point-based network, results in a system that is competitive with state-of-the-art performance on self-driving car benchmarks.

2. Demonstrate the computational-flexibility of the proposed model through showing how a single model designed in the fashion may adapt its inference cost. For instance, a single trained pedestrian model may exceed the predictive performance of a baseline convolutional model [10] by $\sim$ 48% at similar computational demand; or, the same model without retraining may achieve similar predictive performance but with $\sim$ 20% of the computational demand.

3. Demonstrate the ability of the model ot selectively target specific locations of interest. The authors show how temporal context (using only the outputs of previous frames) can be used with the model to improve detection mAP scores by $\sim$ 40%.

Figures: Figure 10, Figure 11, Figure 12, Figure 13, Figure 14

1.2. 2020

1.2.1 AutoML-Zero: Evolving Machine Learning Algorithms From Scratch [19]

Abract: Machine learning research has advanced in multiple aspects, including model structures and learning methods. The effort to automate such research, known as AutoML, has also made significant progress. However, this progress has largely focused on the architecture of neural networks, where it has relied on sophisticated expert-designed layers as building blocks—or similarly restrictive

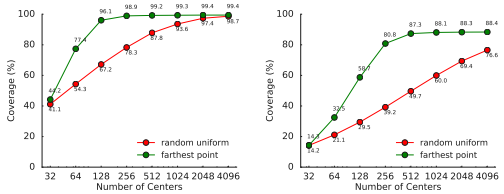


Figure 11: Simple sampling procedures have good coverage over ground truth bounding boxes. The coverage of proposals for cars and vehicles is plotted against the number of samples on KITTI [4] (left) and Waymo [25] (right). Error bars (not shown) range from 0.5%-3.0%.

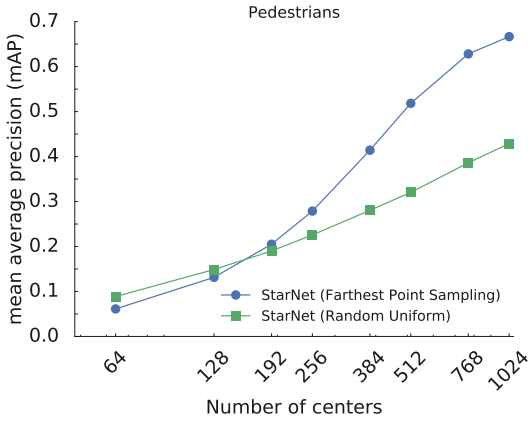


Figure 12: Adaptive computation with a single trained model. Waymo *Validation set* mAP on pedestrians of a single StarNet model trained with 1024 proposals, evaluated with 64 to 1024 proposals.

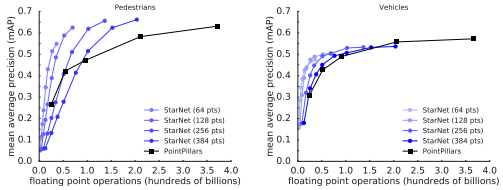


Figure 13: Flexible computational cost of detection for (left) pedestrians and (right) vehicles. Across 5 separately-trained PointPillars models [10], computational cost grows quadratically with increased spatial resolution for the LiDAR pseudo-image. All curves for StarNet arise from a *single* set of saved model weights. Each curve traces out StarNet accuracy on the *Validation set* for a fixed number of point cloud points. Points along on a single curve indicate 64 to 1024 selected centers.

search spaces. Our goal is to show that AutoML can go further: it is possible today to automatically discover complete

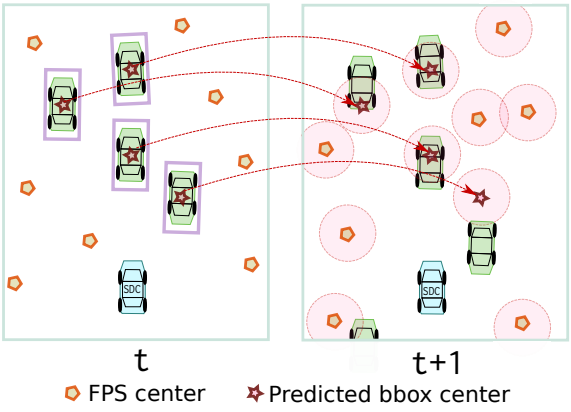


Figure 14: Leveraging previous proposals. Using the highest confidence predicted centers from the previous frames can help improve detection mAP in the next frame.

machine learning algorithms just using basic mathematical operations as building blocks. We demonstrate this by introducing a novel framework that significantly reduces human bias through a generic search space. Despite the vastness of this space, evolutionary search can still discover two-layer neural networks trained by backpropagation. These simple neural networks can then be surpassed by evolving directly on tasks of interest, *e.g.* CIFAR-10 variants, where modern techniques emerge in the top algorithms, such as bilinear interactions, normalized gradients, and weight averaging. Moreover, evolution adapts algorithms to different task types: *e.g.*, dropout-like techniques appear when little data is available. We believe these preliminary successes in discovering machine learning algorithms from scratch indicate a promising new direction for the field.
URL: https://github.com/google-research/google-research/tree/master/automl_zero
Contribution:

1. AutoML-Zero, the proposal to automatically search for ML algorithms from scratch with minimal human design.

2. A novel framework with open-sources code and a search space that combines only basic mathematical operations.

3. Detailed results to show potential through the discovery of nuanced ML algorithms using evolutionary search.

1.2.2 Are Labels Necessary for Neural Architecture Search [13]

Abract: Existing neural network architectures in computer vision — whether designed by humans or by machines — were typically found using both images and their associated labels. In this paper, we ask the question: can we find high-quality neural architectures using only images, but no human-annotated labels? To answer this question, we first define a new setup called Unsupervised Neural Architecture Search (UnNAS). We then conduct two sets of experiments. In sample-based experiments, we train a large number (500) of diverse architectures with either supervised or unsupervised objectives, and find that the architecture rankings produced with and without labels are highly correlated. In search-based experiments, we run a well-established NAS algorithm (DARTS) using various unsupervised objectives, and report that the architectures searched without labels can be competitive to their counterparts searched with labels. Together, these results reveal the potentially surprising finding that labels are not necessary, and the image statistics alone may be sufficient to identify good neural architectures.
URL: https://github.com/facebookresearch/pycls
Contribution:

1. The architecture rankings produced by supervised and self-supervised pretext tasks are *highly correlated*. This finding is consistent across two datasets, two search spaces, and three pretext tasks.

2. The architectures searched without human annotations are *comparable in performance* to their supervised counterparts. This result is consistent across three pretext tasks, three pretext datasets, and two target tasks. There are even cases where unsupervised search outperforms supervised search.

3. Existing NAS approaches typically use *labeled* images from a *smaller* dataset to learn transferable architectures. We present evidence that using *unlabeled* images from a *large* dataset may be a more promising approach.

Figures: Figure 15, Figure 16, Figure 17, Figure 18, Figure 19

1.2.3 Inferring Spatial Uncertainty in Object Detection [27]

Abract: The availability of real-world datasets is the prerequisite to develop object detection methods for autonomous driving. While ambiguity exists in object labels due to errorprone annotation process or sensor observation noises, current object detection datasets only provide deterministic annotations, without considering their uncertainty. This precludes an indepth evaluation among different object detection methods, especially for those that explicitly model predictive probability. In this work, we propose a

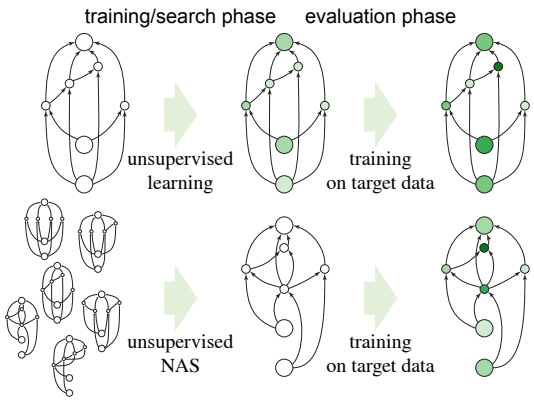


Figure 15: **Unsupervised neural architecture search**, or UnNAS, is a new problem setup that helps answer the question "are labels necessary for neural architecture search?" In traditional unsupervised learning (top panel), the *training phase* learns the weights of a fixed architecture; then the *evaluation phase* measures the quality of the weights by training a classifier (either by fine-tuning the weights or using them as a fixed feature extractor) using supervision from the target dataset. Analogously, in UnNAS (bottom panel), the *search phase* searches for an architecture without using labels; and the *evaluation phase* measures the quality of the architecture found by an UnNAS algorithm by training the architecture's weights using supervision from the target dataset.

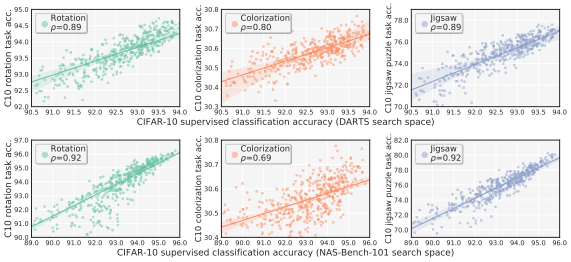


Figure 16: **Correlation between supervised classification accuracy *vs.* pretext task accuracy on CIFAR-10 ("C10")**. Top panel: DARTS search space. Bottom panel: NAS-Bench-101 search space. The straight lines are fit with robust linear regression (same for Figure 17 and Figure 18).

generative model to estimate bounding box label uncertainties from LiDAR point clouds, and define a new representation of the probabilistic bounding box through spatial distribution. Comprehensive experiments show that the proposed model represents uncertainties commonly seen in driving scenarios. Based on the spatial distribution, we further propose an extension of IoU, called the Jaccard IoU (JIoU), as a new evaluation metric that incorporates label uncertainty. The experiments on the KITTI [4] and the Waymo Open

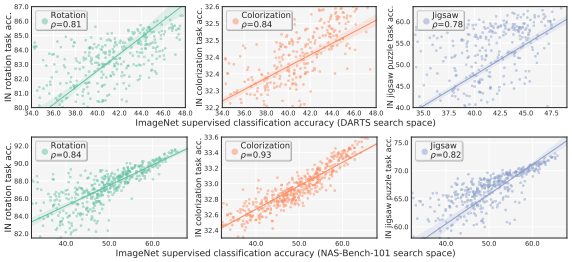


Figure 17: **Correlation between supervised classification accuracy *vs.* pretext task accuracy on ImageNet ("IN") [21]**. Top panel: DARTS search space. Bottom panel: NAS-Bench-101 search space.

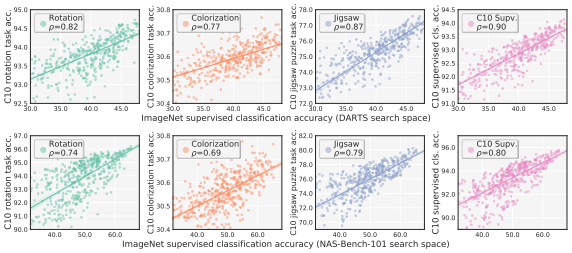


Figure 18: **Correlation between ImageNet [21] supervised classification accuracy *vs.* CIFAR-10 ("C10") pretext task accuracy**. Rankings of architectures are highly correlated between supervised classification and three unsupervised tasks, as measured by Spearman's rank correlation (ρ). We also show rank correlation using CIFAR-10 supervised proxy in the rightmost panel. Top panel: DARTS search space. Bottom panel: NAS-Bench-101 search space.

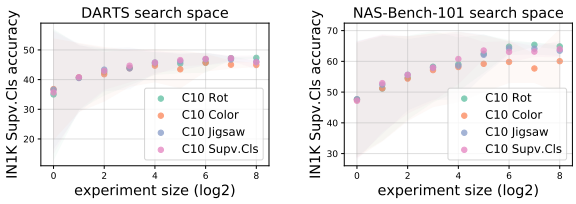


Figure 19: **Random experiment efficiency curves**. Left panel: DARTS search space. Right panel: NAS-Bench-101 search space. We show the range of ImageNet [21] classification accuracies of top architectures identified by the three pretext tasks and the supervised task under various experiment sizes. See text for more details.

Datasets [25] show that JIoU is superior to IoU when evaluating probabilistic object detectors.

Contribution:

1. Infer the inherent uncertainty using a generative model in bounding box labels for object detection, and systematically analyze its parameters

2. Propose a new evaluation metric called JIoU for the object localization task, which considers label uncertainty and provides richer information than IoU when analyzing probabilistic object detectors.

1.2.4 Improved Baselines with Momentum Contrastive Learning [3]

Abract: Contrastive unsupervised learning has recently shown encouraging progress, *e.g.*, in Momentum Contrast (MoCo) [7] and SimCLR. In this note, we verify the effectiveness of two of SimCLR's design improvements by implementing them in the MoCo framework. With simple modifications to MoCo—namely, using an MLP projection head and more data augmentation—we establish stronger baselines that outperform SimCLR and do not require large training batches. We hope this will make state-of-the-art unsupervised learning research more accessible. Code will be made public.
URL: https://github.com/facebookresearch/moco

1.2.5 Range Conditioned Dilated Convolutions for Scale Invariant 3D Object Detection [1]

Abract: This paper presents a novel 3D object detection framework that processes LiDAR data directly on a representation of the sensor's native range images. When operating in the range image view, one faces learning challenges, including occlusion and considerable scale variation, limiting the obtainable accuracy. To address these challenges, a rangeconditioned dilated block (RCD) is proposed to dynamically adjust a continuous dilation rate as a function of the measured range, achieving scale invariance. Furthermore, soft range gating helps mitigate the effect of occlusion. An end-to-end trained box-refinement network brings additional performance improvements in occluded areas, and produces more accurate bounding box predictions. On the challenging Waymo Open Dataset [25], our improved range-based detector outperforms state of the art at long range detection. Our framework is superior to prior multiview, voxel-based methods over all ranges, setting a new baseline for range-based 3D detection on this large scale public dataset.
Contribution:

1. a novel range conditioned dilated (RCD) convolutional operator is introduced that is capable of dynamically adjusting the local receptive field using the measured distance, to provide a consistent scale relative to the convolutional kernel at any distance;

2. a region convolutional neural network (RCNN) based second stage network is investigated in the context of range image-based 3D object detection;

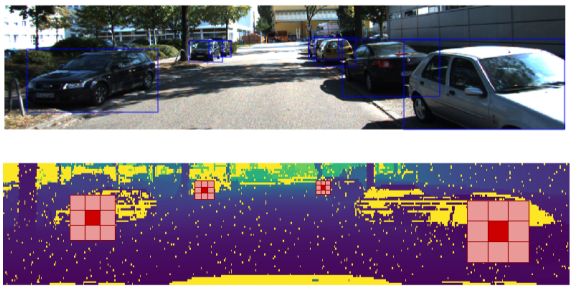


Figure 20: **Top:** A reference RGB image of a scene with object detections. **Bottom:** Our proposed RCD layer dynamically adjusts the receptive field using the measured range to the object. Here a basic 3×3 kernel is used for illustration. *The RGB image is only for illustration purposes.*

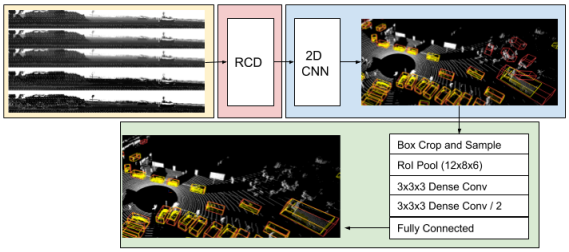


Figure 21: Network architecture overview. The input range image is of size $64 \times 2650 \times 8$. The input is first passed to a range conditioned dilation layer, and a 2D convolution net. This first stage (in blue) generates 3D box proposal for each point with corresponding classification score and likelihood of representing the same box as neighbors. High scoring proposals are processed and then passed to an RoI pooler to divide each proposal to a $12 \times 8 \times 6$ grid. Then the second stage (in green) applies 3D convolution and finally a fully connected layer predicts a 3D box and a score.

3. a new baseline is set for range image-based 3D object detection on a public dataset. The introduced RCD based model performs especially well at long ranges (large distances), where voxel and sparse-point cloud based approaches suffer from point sparsity issues.

Figures: Figure 20, Figure 21, Figure 22, Figure 23, Figure 24

2. CVPR

2.1. 2012

2.1.1 Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite [4]

Abract: Today, visual recognition systems are still rarely employed in robotics applications. Perhaps one of the main reasons for this is the lack of demanding benchmarks that

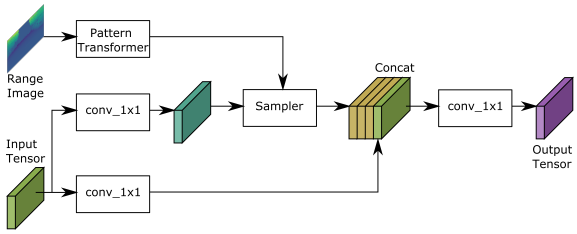


Figure 22: An overview of the range-conditioned dilation block detailing the interaction between the various modules. The sampler is solely responsible for the spatial processing of the input tensor where the receptive field is driven by the input range image. Further details are described in text.

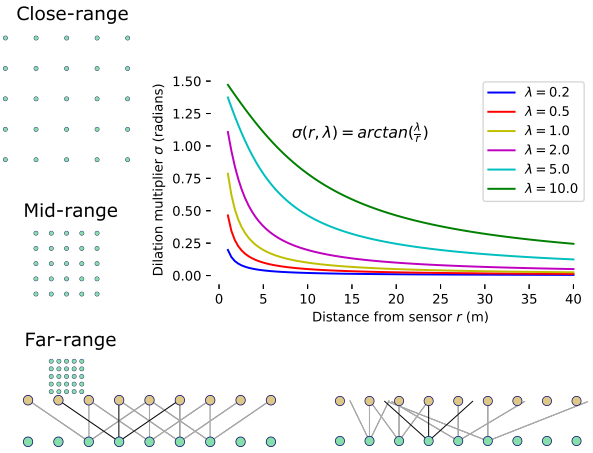


Figure 23: (a) Illustration of dilated sampling at different ranges. (b) The rate of the convolutional filter dilation as a function of both the distance and the nominal object size λ. (c) A reference for a traditional convolution with a discrete dilation rate (here set to 2) while (d) illustrates our continuous dilation rates used by our range-conditioned-dilated convolution going form narrow dilation to wide, which corresponds to a change in range from far to near, respectively.

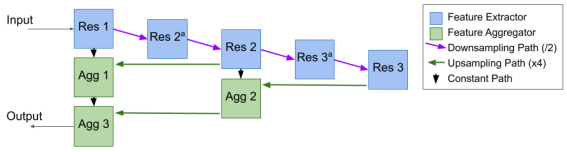


Figure 24: 2D CNN backbone after the initial RCD layer. Number of bottlenecks units in each block: (Res1: 5), (Res2,Res2a: 7), (Res3,Res3a: 9), (Agg1,Agg2,Agg3: 4). The bottleneck is the one described in [8] with bottleneck depth set to the bottleneck input channel size divided by 4. The output channel size from each layer are 64, 128, 256.

mimic such scenarios. In this paper, we take advantage of our autonomous driving platform to develop novel challenging benchmarks for the tasks of stereo, optical flow, visual odometry/SLAM and 3D object detection. Our recording platform is equipped with four high resolution video cameras, a Velodyne laser scanner and a state-of-the-art localization system. Our benchmarks comprise 389 stereo and optical flow image pairs, stereo visual odometry sequences of 39.2 km length, and more than 200k 3D object annotations captured in cluttered scenarios (up to 15 cars and 30 pedestrians are visible per image). Results from state-of-the-art algorithms reveal that methods ranking high on established datasets such as Middlebury perform below average when being moved outside the laboratory to the real world. Our goal is to reduce this bias by providing challenging benchmarks with novel difficulties to the computer vision community. Our benchmarks are available online at: www.cvlibs.net/datasets/kitti.

2.2. 2016

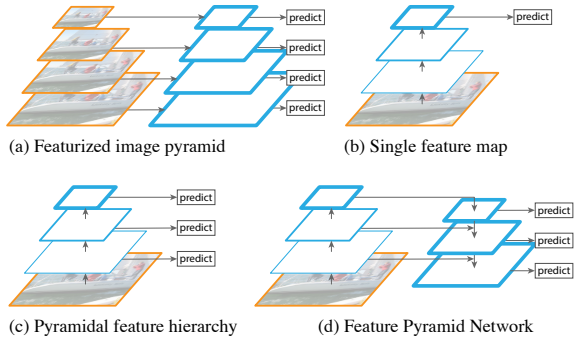
2.2.1 Deep Residual Learning for Image Recognition [8]

Abract: Deeper neural networks are more difficult to train. We present a residual learning framework to ease the training of networks that are substantially deeper than those used previously. We explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions. We provide comprehensive empirical evidence showing that these residual networks are easier to optimize, and can gain accuracy from considerably increased depth. On the ImageNet [21] dataset we evaluate residual nets with a depth of up to 152 layers-8x deeper than VGG nets but still having lower complexity. An ensemble of these residual nets achieves 3.57% error on the ImageNet test set. This result won the 1st place on the ILSVRC 2015 classification task. We also present analysis on CIFAR-10 with 100 and 1000 layers. The depth of representations is of central importance for many visual recognition tasks. Solely due to our extremely deep representations, we obtain a 28% relative improvement on the COCO [12] object detection dataset. Deep residual nets are foundations of our submissions to ILSVRC & COCO 2015 competitions1, where we also won the 1st places on the tasks of ImageNet detection, ImageNet localization, COCO detection, and COCO segmentation.
URL: https://github.com/KaimingHe/deep-residual-networks

2.3. 2017

2.3.1 Feature Pyramid Networks for Object Detection [11]

Abract: Feature pyramids are a basic component in recognition systems for detecting objects at different scales. But pyramid representations have been avoided in recent object



(a) Featurized image pyramid (b) Single feature map

(c) Pyramidal feature hierarchy (d) Feature Pyramid Network

Figure 25: (a) Using an image pyramid to build a feature pyramid. Features are computed on each of the image scales independently, which is slow. (b) Recent detection systems have opted to use only single scale features for faster detection. (c) An alternative is to reuse the pyramidal feature hierarchy computed by a ConvNet as if it were a featurized image pyramid. (d) Feature Pyramid Network (FPN) is fast like (b) and (c), but more accurate. In this figure, feature maps are indicate by blue outlines and thicker outlines denote semantically stronger features.

detectors that are based on deep convolutional networks, partially because they are slow to compute and memory intensive. In this paper, we exploit the inherent multi-scale, pyramidal hierarchy of deep convolutional networks to construct feature pyramids with marginal extra cost. A top-down architecture with lateral connections is developed for building high-level semantic feature maps at all scales. This architecture, called a Feature Pyramid Network (FPN), shows significant improvement as a generic feature extractor in several applications. Using a basic Faster R-CNN [20] system, our method achieves state-of-the-art single-model results on the COCO [12] detection benchmark without bells and whistles, surpassing all existing single-model entries including those from the COCO 2016 challenge winners. In addition, our method can run at 5 FPS on a GPU and thus is a practical and accurate solution to multi-scale object detection. Code will be made publicly available.

Contribution:

1. Build a top-down feature pyramids with lateral connections at all scales.

2. Create a in-network feature pyramids that can be used to replace featurized image pyramids with out sacrificing representational power, speed, or memory.

Figures: Figure 25, Figure 26, Figure 27, Figure 28

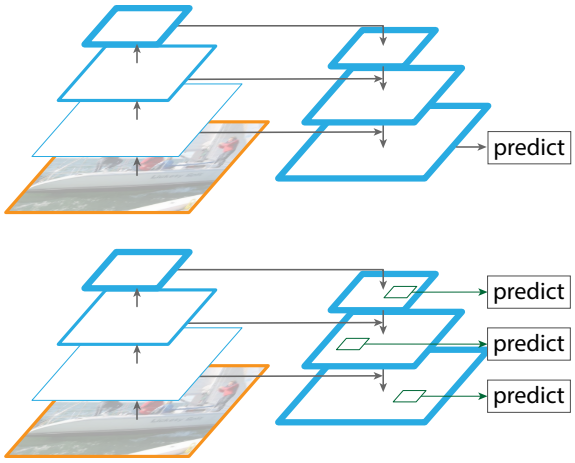


Figure 26: Top: a top-down architecture with skip connections, where predictions are made on the finest level. Bottom: FPN that has a similar structure but leverages it as a *feature pyramid*, with predictions made independently at all levels.

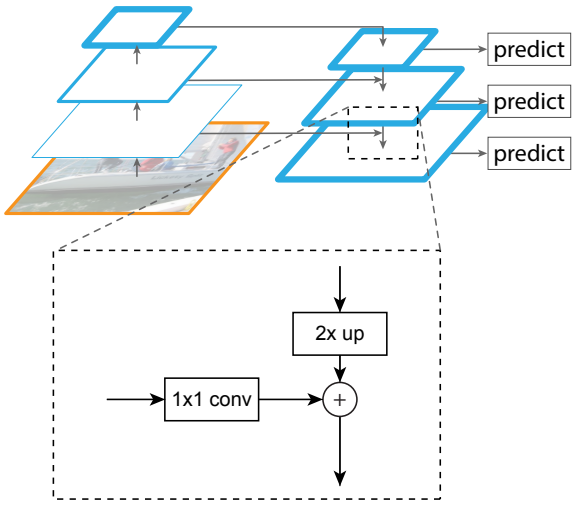


Figure 27: A building block illustrating the lateral connection and the top-down pathway, merged by addition.

2.3.2 PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation [15]

Abract: Point cloud is an important type of geometric data structure. Due to its irregular format, most researchers transform such data to regular 3D voxel grids or collections of images. This, however, renders data unnecessarily voluminous and causes issues. In this paper, we design a novel type of neural network that directly consumes point clouds, which well respects the permutation invariance of points in the input. Our network, named PointNet, provides a unified architecture for applications ranging from object classification, part segmentation, to scene semantic parsing. Though

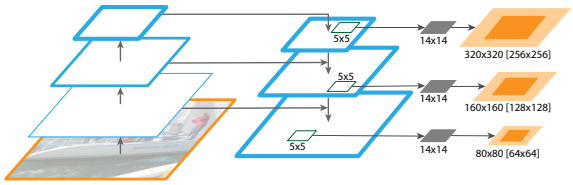


Figure 28: FPN for object segment proposals. The feature pyramid is constructed with identical structure as for object detection. We apply a small MLP on 5×5 windows to generate dense object segments with output dimension of 14×14. Shown in orange are the size of the image regions the mask corresponds to for each pyramid level (levels P_{3-5} are shown here). Both the corresponding image region size (light orange) and canonical object size (dark orange) are shown. Half octaves are handled by an MLP on 7×7 windows ($7 \approx 5\sqrt{2}$), not shown here.

simple, PointNet is highly efficient and effective. Empirically, it shows strong performance on par or even better than state of the art. Theoretically, we provide analysis towards understanding of what the network has learnt and why the network is robust with respect to input perturbation and corruption.

URL: https://github.com/charlesq34/pointnet

Contribution:

1. The authors design a novel deep net architecture suitable for consuming unordered point sets in 3D
2. The authors show such a net can be trained to perform 3D shape classification, shape part segmentation and scene semantic parsing tasks
3. The authors provide thorough empirical and theoretical analysis on the stability and efficiency of our method
4. The authors illustrate the 3D features computed by the selected neurons in the net and develop intuitive explanations for its performance.

Figures: Figure 29

Proof of Theorem: Let $\mathcal{X} = \{S : S \subseteq [0,1] \text{ and } |S| = n\}$.

$f : \mathcal{X} \to \mathbb{R}$ is a continuous function on $\mathcal{X}$ w.r.t to Hausdorff distance $d_H(\cdot,\cdot)$ if the following condition is satisfied:

$\forall \epsilon > 0, \exists \delta > 0$, for any $S, S' \in \mathcal{X}$, if $d_H(S,S') < \delta$, then $|f(S) - f(S')| < \epsilon$.

We show that f can be approximated arbitrarily by composing a symmetric function and a continuous function.

Theorem 1. *Suppose $f : \mathcal{X} \to \mathbb{R}$ is a continuous set function w.r.t Hausdorff distance $d_H(\cdot,\cdot)$. $\forall \epsilon > 0$, $\exists$ a continuous function h and a symmetric function $g(x_1,\dots,x_n) = \gamma \circ MAX$, where γ is a continuous function, MAX is a vector max operator that takes n vectors as input and returns a new vector of the element-wise maximum, such that for any $S \in \mathcal{X}$,*

$$|f(S) - \gamma(MAX(h(x_1),\dots,h(x_n)))| < \epsilon$$

where $x_1,\dots,x_n$ are the elements of S extracted in certain order,

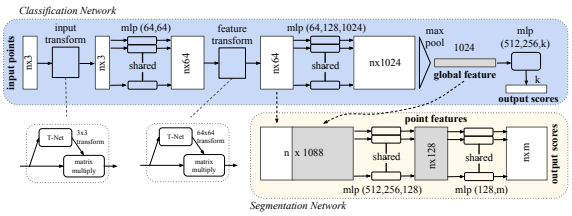


Figure 29: PointNet Architecture. The classification network takes n points as input, applies input and feature transformations, and then aggregates point features by max pooling. The output is classification scores for k classes. The segmentation network is an extension to the classification net. It concatenates global and local features and outputs per point scores. "mlp" stands for multi-layer perceptron, numbers in bracket are layer sizes. Batchnorm is used for all layers with ReLU. Dropout layers are used for the last mlp in classification net.

Proof. By the continuity of f, we take δ_ϵ so that $|f(S) - f(S')| < \epsilon$ for any $S, S' \in \mathcal{X}$ if $d_H(S,S') < \delta_\epsilon$.

Define $K = \lceil 1/\delta_\epsilon \rceil$, which split $[0,1]$ into K intervals evenly and define an auxiliary function that maps a point to the left end of the interval it lies in:

$$\sigma(x) = \frac{\lfloor Kx \rfloor}{K}$$

Let $\tilde{S} = \{\sigma(x) : x \in S\}$, then

$$|f(S) - f(\tilde{S})| < \epsilon$$

because $d_H(S,\tilde{S}) < 1/K \leq \delta_\epsilon$.

Let $h_k(x) = e^{-d(x,[\frac{k-1}{K},\frac{k}{K}])}$ be a soft indicator function where $d(x,I)$ is the point to set (interval) distance. Let $\mathbf{h}(x) = [h_1(x);\dots;h_K(x)]$, then $\mathbf{h} : \mathbb{R} \to \mathbb{R}^K$.

Let $v_j(x_1,\dots,x_n) = \max\{\tilde{h}_j(x_1),\dots,\tilde{h}_j(x_n)\}$, indicating the occupancy of the j-th interval by points in S. Let $\mathbf{v} = [v_1;\dots;v_K]$, then $\mathbf{v} : \underbrace{\mathbb{R} \times \dots \times \mathbb{R}}_{n} \to \{0,1\}^K$ is a symmetric function, indicating the occupancy of each interval by points in S.

Define $\tau : \{0,1\}^K \to \mathcal{X}$ as $\tau(v) = \{\frac{k-1}{K} : v_k \geq 1\}$, which maps the occupancy vector to a set which contains the left end of each occupied interval. It is easy to show:

$$\tau(\mathbf{v}(x_1,\dots,x_n)) \equiv \tilde{S}$$

where $x_1, \ldots, x_n$ are the elements of S extracted in certain order.

Let $\gamma : \mathbb{R}^K \to \mathbb{R}$ be a continuous function such that $\gamma(\mathbf{v}) = f(\tau(\mathbf{v}))$ for $v \in \{0,1\}^K$. Then,

$$\begin{aligned}&|\gamma(\mathbf{v}(x_1, \ldots, x_n)) - f(S)| \\ =&|f(\tau(\mathbf{v}(x_1, \ldots, x_n))) - f(S)| < \epsilon\end{aligned}$$

Note that $\gamma(\mathbf{v}(x_1, \ldots, x_n))$ can be rewritten as follows:

$$\begin{aligned}\gamma(\mathbf{v}(x_1, \ldots, x_n)) =&\gamma(\text{MAX}(\mathbf{h}(x_1), \ldots, \mathbf{h}(x_n))) \\ =&(\gamma \circ \text{MAX})(\mathbf{h}(x_1), \ldots, \mathbf{h}(x_n))\end{aligned}$$

Obviously $\gamma \circ \text{MAX}$ is a symmetric function. □

Next we give the proof of Theorem 2. We define $\mathbf{u} = \underset{x_i \in S}{\text{MAX}}\{h(x_i)\}$ to be the sub-network of f which maps a point set in $[0,1]^m$ to a K-dimensional vector. The following theorem tells us that small corruptions or extra noise points in the input set is not likely to change the output of our network:

Theorem 2. *Suppose* $\mathbf{u} : \mathcal{X} \to \mathbb{R}^K$ *such that* $\mathbf{u} = \underset{x_i \in S}{\text{MAX}}\{h(x_i)\}$ *and* $f = \gamma \circ \mathbf{u}$. *Then,*

(a) $\forall S, \exists\, \mathcal{C}_S, \mathcal{N}_S \subseteq \mathcal{X}$, $f(T) = f(S)$ *if* $\mathcal{C}_S \subseteq T \subseteq \mathcal{N}_S$;

(b) $|\mathcal{C}_S| \leq K$

Proof. Obviously, $\forall S \in \mathcal{X}$, $f(S)$ is determined by $\mathbf{u}(S)$. So we only need to prove that $\forall S, \exists\, \mathcal{C}_S, \mathcal{N}_S \subseteq \mathcal{X}, f(T) = f(S)$ if $\mathcal{C}_S \subseteq T \subseteq \mathcal{N}_S$.

For the jth dimension as the output of $\mathbf{u}$, there exists at least one $x_j \in \mathcal{X}$ such that $h_j(x_j) = \mathbf{u}_j$, where h_j is the jth dimension of the output vector from h. Take $\mathcal{C}_S$ as the union of all x_j for $j = 1, \ldots, K$. Then, $\mathcal{C}_S$ satisfies the above condition.

Adding any additional points x such that $h(x) \leq \mathbf{u}(S)$ at all dimensions to $\mathcal{C}_S$ does not change $\mathbf{u}$, hence f. Therefore, $\mathcal{T}_S$ can be obtained adding the union of all such points to $\mathcal{N}_S$. □

2.4. 2019

2.4.1 PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud [23]

Abract: In this paper, we propose PointRCNN for 3D object detection from raw point cloud. The whole framework is composed of two stages: stage-1 for the bottom-up 3D proposal generation and stage-2 for refining proposals in the canonical coordinates to obtain the final detection results. Instead of generating proposals from RGB image or projecting point cloud to bird's view or voxels as previous methods do, our stage-1 sub-network directly generates a small number of high-quality 3D proposals from point cloud in a bottom-up manner via segmenting the point cloud of the whole scene into foreground points and background. The stage-2 sub-network transforms the pooled points of each proposal to canonical coordinates to learn better local spatial features, which is combined with global semantic features of each point learned in stage-1 for accurate box refinement and confidence prediction. Extensive experiments on the 3D detection benchmark of KITTI [4] dataset show that our proposed architecture outperforms state-of-the-art methods with remarkable margins by using only point cloud as input. The code is available at https://github.com/sshaoshuai/PointRCNN

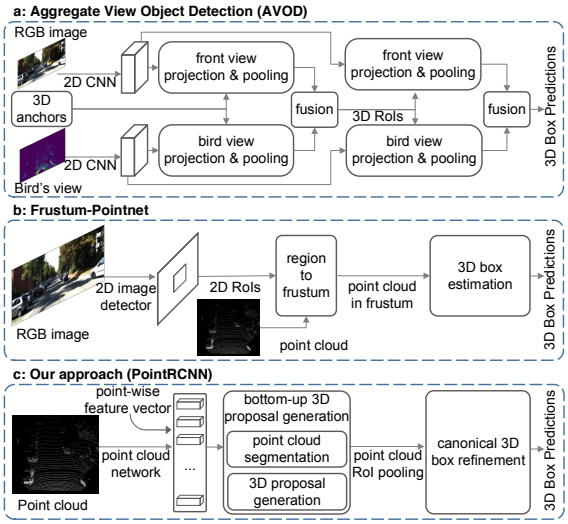


Figure 30: Comparison with state-of-the-art methods. Instead of generating proposals from fused feature maps of bird's view and front view, or RGB images, our method directly generates 3D proposals from raw point cloud in a bottom-up manner.

Contribution:

1. PointRCNN propose a novel bottom-up point cloud-based 3D bounding box proposal generation algorithm, which generates a small number of high-quality 3D proposals via segmenting the point cloud into foreground objects and background. The learned point representation from segmentation is not only good at proposal generation but is also helpful for the later box refinement.

2. The proposed canonical 3D bounding box refinement takes advantages of the high-recall box proposals generated from stage-1 and learns to predict box coordinates refinements in the canonical coordinates with robust bin-based losses.

Figures: Figure 30, Figure 31, Figure 32, Figure 33

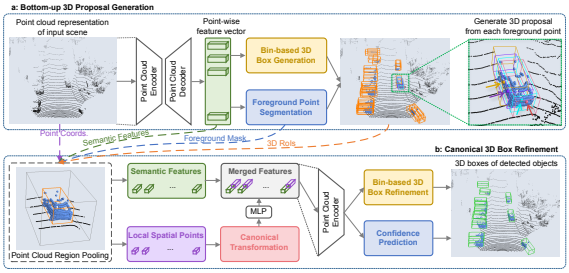


Figure 31: The **PointRCNN** architecture for 3D object detection from point cloud. The whole network consists of two parts: (a) for generating 3D proposals from raw point cloud in a bottom-up manner. (b) for refining the 3D proposals in canonical coordinate.

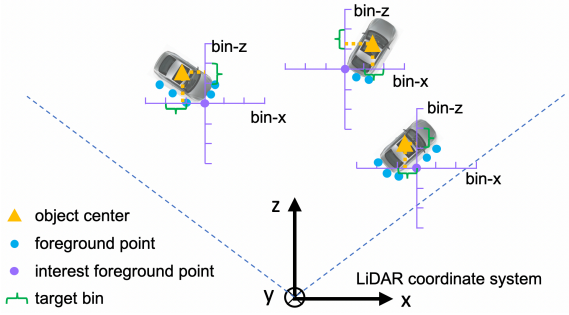


Figure 32: Illustration of bin-based localization. The surrounding area along X and Z axes of each foreground point is split into a series of bins to locate the object center.

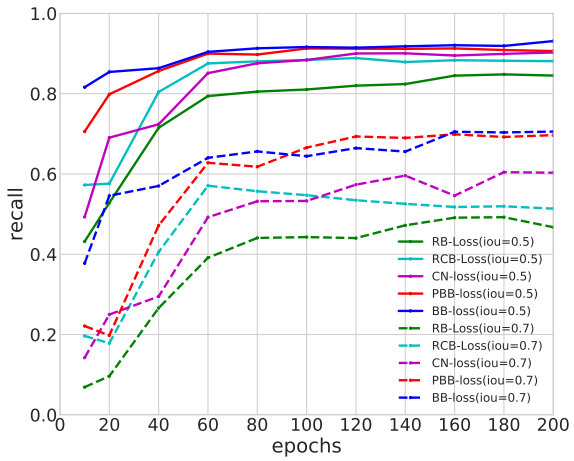


Figure 33: Recall curves of applying different bounding box regression loss function.

2.4.2 PointPillars: Fast Encoders for Object Detection From Point Clouds [10]

Abract: Object detection in point clouds is an important aspect of many robotics applications such as autonomous driving. In this paper, we consider the problem of encoding a point cloud into a format appropriate for a downstream detection pipeline. Recent literature suggests two types of encoders; fixed encoders tend to be fast but sacrifice accuracy, while encoders that are learned from data are more accurate, but slower. In this work, we propose PointPillars, a novel encoder which utilizes PointNets[15, 16] to learn a representation of point clouds organized in vertical columns (pillars). While the encoded features can be used with any standard 2D convolutional detection architecture, we further propose a lean downstream network. Extensive experimentation shows that PointPillars outperforms previous encoders with respect to both speed and accuracy by a large margin. Despite only using lidar, our full detection pipeline significantly outperforms the state of the art, even among fusion methods, with respect to both the 3D and bird's eye view KITTI [4] benchmarks. This detection performance is achieved while running at 62 Hz: a 2 - 4 fold runtime improvement. A faster version of our method matches the state of the art at 105 Hz. These benchmarks suggest that PointPillars is an appropriate encoding for object detection in point clouds.
URL: `https://github.com/nutonomy/second.pytorch`
Contribution:

1. By learning features instead of relying on fixed encoders, PointPillars can leverage the full information represented by the point cloud.

2. By operating on pillars instead of voxels there is no need to tune the binning of the vertical direction by hand.

3. Pillars are highly efficient because all key operations can be formulated as 2D convolutions which are extremely efficient to compute on a GPU.

Figures: Figure 34

2.5. 2020

2.5.1 Designing Network Design Spaces [18]

Abract: In this work, we present a new network design paradigm. Our goal is to help advance the understanding of network design and discover design principles that generalize across settings. Instead of focusing on designing individual network instances, we design network design spaces that parametrize populations of networks. The overall process is analogous to classic manual design of networks, but elevated to the design space level. Using our methodology we explore the structure aspect of network design and arrive at a low-dimensional design space consisting of simple, regular networks that we call `RegNet`. The core insight of the `RegNet` parametrization is surprisingly simple: widths

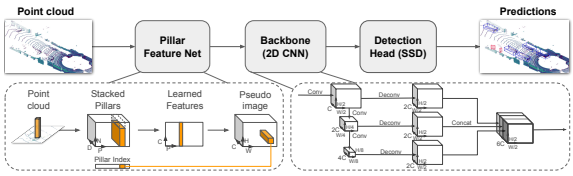


Figure 34: Network overview. The main components of the network are a Pillar Feature Network, Backbone, and SSD Detection Head. The raw point cloud is converted to a stacked pillar tensor and pillar index tensor. The encoder uses the stacked pillars to learn a set of features that can be scattered back to a 2D pseudo-image for a convolutional neural network. The features from the backbone are used by the detection head to predict 3D bounding boxes for objects. Note: here we show the backbone dimensions for the car network.

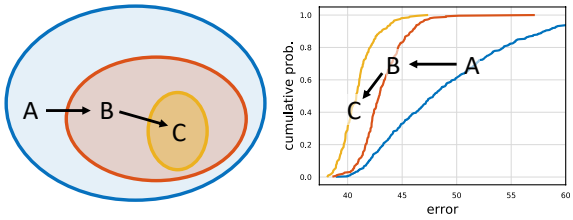


Figure 35: **Design space design.** We propose to *design* network design spaces, where a design space is a parametrized *set* of possible model architectures. Design space design is akin to manual network design, but elevated to the *population* level. In each step of our process the input is an initial design space and the output is a refined design space of simpler or better models. We characterize the quality of a design space by sampling models and inspecting their *error distribution*. For example, in the figure above we start with an initial design space `A` and apply two refinement steps to yield design spaces `B` then `C`. In this case `C` $\subseteq$ `B` $\subseteq$`A` (left), and the error distributions are strictly improving from `A` to `B` to `C` (right). The hope is that *design principles* that apply to model populations are more likely to be robust and generalize.

and depths of good networks can be explained by a quantized linear function. We analyze the `RegNet` design space and arrive at interesting findings that do not match the current practice of network design. The `RegNet` design space provides simple and fast networks that work well across a wide range of flop regimes. Under comparable training settings and flops, the `RegNet` models outperform the popular EfficientNet models while being up to 5$\times$ faster on GPUs.

URL: `https://github.com/facebookresearch/pycls`

Figures: Figure 35

Algorithm 1 Pseudocode of MoCo in a PyTorch-like style.

```
# f_q, f_k: encoder networks for query and key
# queue: dictionary as a queue of K keys (CxK)
# m: momentum
# t: temperature

f_k.params = f_q.params # initialize
for x in loader: # load a minibatch x with N samples
   x_q = aug(x) # a randomly augmented version
   x_k = aug(x) # another randomly augmented version

   q = f_q.forward(x_q) # queries: NxC
   k = f_k.forward(x_k) # keys: NxC
   k = k.detach() # no gradient to keys

   # positive logits: Nx1
   l_pos = bmm(q.view(N,1,C), k.view(N,C,1))

   # negative logits: NxK
   l_neg = mm(q.view(N,C), queue.view(C,K))

   # logits: Nx(1+K)
   logits = cat([l_pos, l_neg], dim=1)

   # contrastive loss, Eqn.(1)
   labels = zeros(N) # positives are the 0-th
   loss = CrossEntropyLoss(logits/t, labels)

   # SGD update: query network
   loss.backward()
   update(f_q.params)

   # momentum update: key network
   f_k.params = m*f_k.params+(1-m)*f_q.params

   # update dictionary
   enqueue(queue, k) # enqueue the current minibatch
   dequeue(queue) # dequeue the earliest minibatch
```

`bmm`: batch matrix multiplication; `mm`: matrix multiplication; `cat`: concatenation.

2.5.2 Momentum Contrast for Unsupervised Visual Representation Learning [7]

Abract: We present Momentum Contrast (MoCo) for unsupervised visual representation learning. From a perspective on contrastive learning as dictionary look-up, we build a dynamic dictionary with a queue and a moving-averaged encoder. This enables building a large and consistent dictionary on-the-fly that facilitates contrastive unsupervised learning. MoCo provides competitive results under the common linear protocol on ImageNet [21] classification. More importantly, the representations learned by MoCo transfer well to downstream tasks. MoCo can **outperform** its supervised pre-training counterpart in **7** detection/segmentation tasks on PASCAL VOC, COCO [12], and other datasets, sometimes surpassing it by large margins. This suggests that the gap between unsupervised and supervised representation learning has been largely closed in many vision tasks.

URL: `https://github.com/facebookresearch/moco`

Algorithm: Algorithm 1

Figures: Figure 36, Figure 37, Figure 38

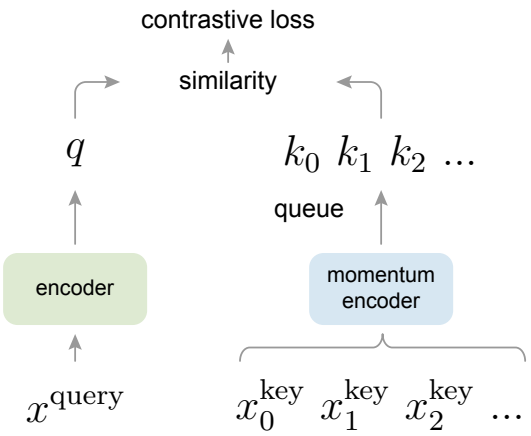


Figure 36: Momentum Contrast (MoCo) trains a visual representation encoder by matching an encoded query q to a dictionary of encoded keys using a contrastive loss. The dictionary keys $\{k_0, k_1, k_2, ...\}$ are defined on-the-fly by a set of data samples. The dictionary is built as a queue, with the current mini-batch enqueued and the oldest mini-batch dequeued, decoupling it from the mini-batch size. The keys are encoded by a slowly progressing encoder, driven by a momentum update with the query encoder. This method enables a large and consistent dictionary for learning visual representations.

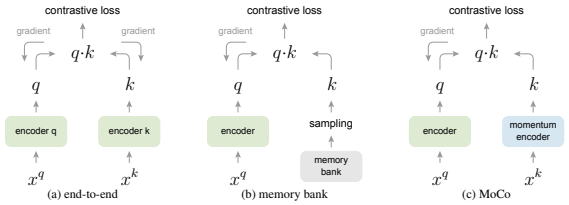


Figure 37: **Conceptual comparison of three contrastive loss mechanisms** (empirical comparisons are in Figure 38). Here we illustrate one pair of query and key. The three mechanisms differ in how the keys are maintained and how the key encoder is updated. **(a)**: The encoders for computing the query and key representations are updated ***end-to-end*** by back-propagation (the two encoders can be different). **(b)**: The key representations are sampled from a ***memory bank***. **(c)**: ***MoCo*** encodes the new keys on-the-fly by a momentum-updated encoder, and maintains a queue (not illustrated in this figure) of keys.

2.5.3 Point-GNN: Graph Neural Network for 3D Object Detection in a Point Cloud [24]

Abract: In this paper, we propose a graph neural network to detect objects from a LiDAR point cloud. Towards this end, we encode the point cloud efficiently in a fixed radius near-neighbors graph. We design a graph neural network, named Point-GNN, to predict the category and shape of the object that each vertex in the graph belongs to. In Point-GNN, we propose an auto-registration mechanism to reduce translation variance, and also design a box merging and scoring operation to combine detections from multiple vertices accurately. Our experiments on the KITTI [4] benchmark show the proposed approach achieves leading accuracy using the point cloud alone and can even surpass fusion-based algorithms. Our results demonstrate the potential of using the graph neural network as a new approach for 3D object detection. The code is available at https://github.com/WeijingShi/Point-GNN.

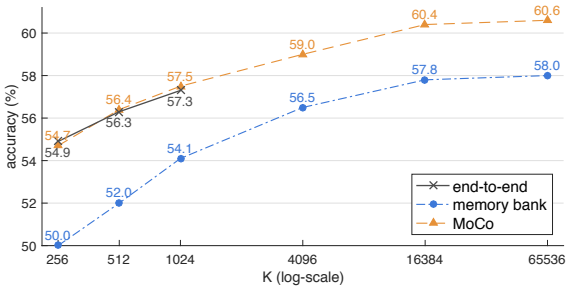


Figure 38: **Comparison of three contrastive loss mechanisms** under the ImageNet linear classification protocol. We adopt the same pretext task and only vary the contrastive loss mechanism Figure 37). The number of negatives is K in memory bank and MoCo, and is $K-1$ in end-to-end (offset by one because the positive key is in the same mini-batch). The network is ResNet-50.

Contribution:

1. The authors propose a new object detection approach using graph neural network on the point cloud.

2. The authors design Point-GNN, a graph neural network with an auto-registration mechanism that detects multiple objects in a single shot.

3. The authors achieve state-of-the-art 3D object detection accuracy in the KITTI [4] benchmark and analyze the effectiveness of each component in depth.

Algorithm: Algorithm 1
Figures: Figure 39, Figure 40

2.5.4 PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection [22]

Abract: We present a novel and high-performance 3D object detection framework, named PointVoxel-RCNN (PV-RCNN), for accurate 3D object detection from point clouds. Our proposed method deeply integrates both 3D voxel Convolutional Neural Network (CNN) and PointNet-based [16]

Algorithm 1: NMS with Box Merging and Scoring

Input: $\mathcal{B} = \{b_1, ..., b_n\}, \mathcal{D} = \{d_1, ..., d_n\}, T_h$
$\mathcal{B}$ is the set of detected bounding boxes.
$\mathcal{D}$ is the corresponding detection scores.
T_h is an overlapping threshold value.
Green color marks the main modifications.

$\mathcal{M} \leftarrow \{\}, \mathcal{Z} \leftarrow \{\}$
while $\mathcal{B} \neq empty$ **do**
  $i \leftarrow argmax\ D$
  $\mathcal{L} \leftarrow \{\}$
  for b_j *in* $\mathcal{B}$ **do**
    if $iou(b_i, b_j) > T_h$ **then**
      $\mathcal{L} \leftarrow \mathcal{L} \cup b_j$
      $\mathcal{B} \leftarrow \mathcal{B} - b_j, \mathcal{D} \leftarrow \mathcal{D} - d_j$
    end
  end
  $m \leftarrow median(\mathcal{L})$
  $o \leftarrow occlusion(m)$
  $z \leftarrow (o+1)\sum_{b_k \in \mathcal{L}} IoU(m, b_k) d_k$
  $\mathcal{M} \leftarrow \mathcal{M} \cup m, \mathcal{Z} \leftarrow \mathcal{Z} \cup z$
end
return $\mathcal{M}$, $\mathcal{Z}$

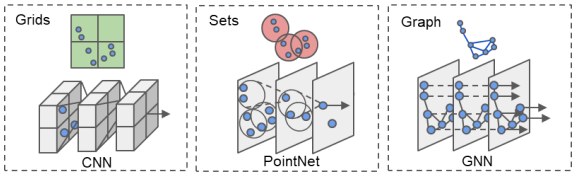


Figure 39: Three point cloud representations and their common processing methods.

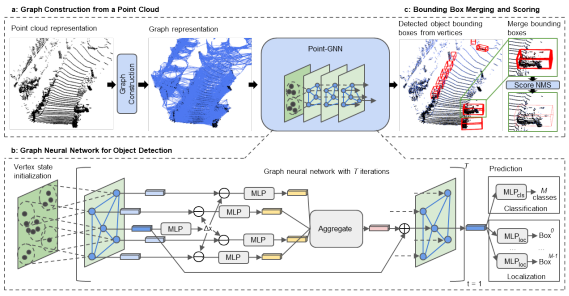


Figure 40: The architecture of the proposed approach. It has three main components: (a) graph construction from a point cloud, (b) a graph neural network for object detection, and (c) bounding box merging and scoring.

set abstraction to learn more discriminative point cloud features. It takes advantages of efficient learning and high-quality proposals of the 3D voxel CNN and the flexible receptive fields of the PointNet-based networks. Specifically, the proposed framework summarizes the 3D scene with a 3D voxel CNN into a small set of keypoints via a novel voxel set abstraction module to save follow-up computations and also to encode representative scene features. Given the high-quality 3D proposals generated by the voxel CNN, the RoI-grid pooling is proposed to abstract proposal-specific features from the keypoints to the RoI-grid points via keypoint set abstraction with multiple receptive fields. Compared with conventional pooling operations, the RoI-grid feature points encode much richer context information for accurately estimating object confidences and locations. Extensive experiments on both the KITTI [4] dataset and the Waymo [25] Open dataset show that our proposed PV-RCNN surpasses state-of-the-art 3D detection methods with remarkable margins by using only point clouds.

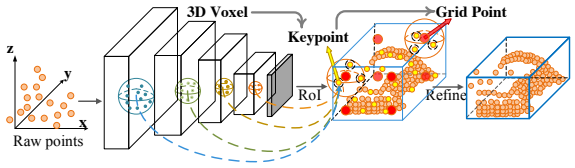


Figure 41: The PV-RCNN framework deeply integrates both the voxel-based and the PointNet-based networks via a two-step strategy including the voxel-to-keypoint 3D scene encoding and the keypoint-to-grid RoI feature abstraction for improving the performance of 3D object detection.

URL: https://github.com/sshaoshuai/PCDet
Contribution:

1. PV-RCNN takes advantages of both voxel-based and point-based methods for 3D point-cloud feature learning.
2. The proposed method utilize the voxel-to-keypoint scene encoding scheme to encode multi-scale voxel features of the whole scene to a small set of sampled keypoints by the voxel set abstraction layer. These keypoint features not only preserve accurate location but also encode the rich scene context, which boost the 3D detection performance significantly.
3. For accurate box refinement and confidence prediction, the proposed method aggregates richer context information from the scene with multiple receptive fields by a multi-scale RoI feature abstraction layer for grid points in each proposal.

Figures: Figure 41, Figure 42, Figure 43

2.5.5 RandLA-Net: Efficient Semantic Segmentation of Large-Scale Point Clouds [9]

Abract: We study the problem of efficient semantic segmentation for large-scale 3D point clouds. By relying on

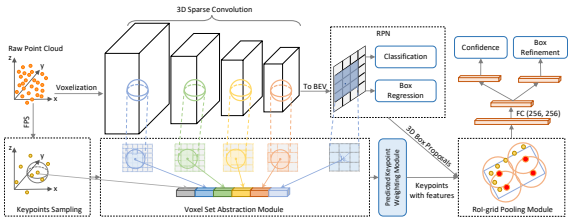


Figure 42: The overall architecture of PV-RCNN. The raw point clouds are first voxelized to feed into the 3D sparse convolution based encoder to learn multi-scale semantic features and generate 3D object proposals. Then the learned voxel-wise feature volumes at multiple neural layers are summarized into a small set of key points via the novel voxel set abstraction module. Finally the keypoint features are aggregated to the RoI-grid points to learn proposal specific features for fine-grained proposal refinement and confidence prediction.

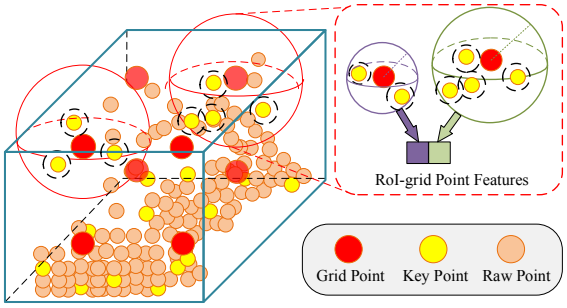


Figure 43: Illustration of RoI-grid pooling module. Rich context information of each 3D RoI is aggregated by the set abstraction operation with multiple receptive fields.

expensive sampling techniques or computationally heavy pre/post-processing steps, most existing approaches are only able to be trained and operate over small-scale point clouds. In this paper, we introduce RandLA-Net, an efficient and lightweight neural architecture to directly infer per-point semantics for large-scale point clouds. The key to our approach is to use random point sampling instead of more complex point selection approaches. Although remarkably computation and memory efficient, random sampling can discard key features by chance. To overcome this, we introduce a novel local feature aggregation module to progressively increase the receptive field for each 3D point, thereby effectively preserving geometric details. Extensive experiments show that our RandLA-Net can process 1 million points in a single pass with up to 200× faster than existing approaches. Moreover, our RandLA-Net clearly surpasses state-of-the-art approaches for semantic segmentation on two large-scale benchmarks Semantic3D and SemanticKITTI.

URL: https://github.com/QingyongHu/

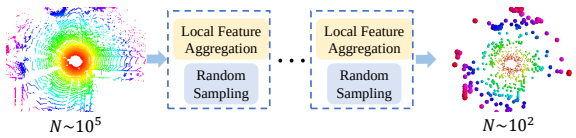


Figure 44: In each layer of RandLA-Net, the large-scale point cloud is significantly downsampled, yet is capable of retaining features necessary for accurate segmentation.

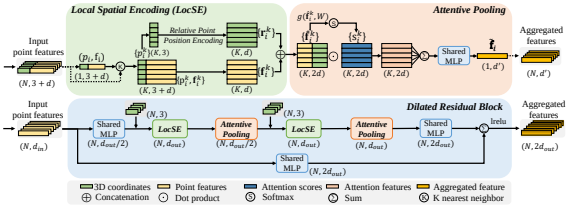


Figure 45: The proposed local feature aggregation module. The top panel shows the location spatial encoding block that extracts features, and the attentive pooling mechanism that weights the most important, based on the local context and geometry. The bottom panel shows how two of these components are chained together, to increase the receptive field size, within a residual block.

RandLA-Net

Contribution:

1. The authors analyze and compare existing sampling approaches, identifying random sampling as the most suitable component for efficient learning on large-scale point clouds.
2. The authors propose an effective local feature aggregation module to automatically preserve complex local structures by progressively increasing the receptive field for each point.
3. The authors demonstrate significant memory and computational gains over baselines, and surpass the state-of-the-art semantic segmentation methods on multiple large-scale benchmarks.

Figures: Figure 44, Figure 45, Figure 46

2.5.6 Scalability in Perception for Autonomous Driving: Waymo Open Dataset [25]

Abract: The research community has increasing interest in autonomous driving research, despite the resource intensity of obtaining representative real world data. Existing selfdriving datasets are limited in the scale and variation of the environments they capture, even though generalization within and between operating regions is crucial to the overall viability of the technology. In an effort

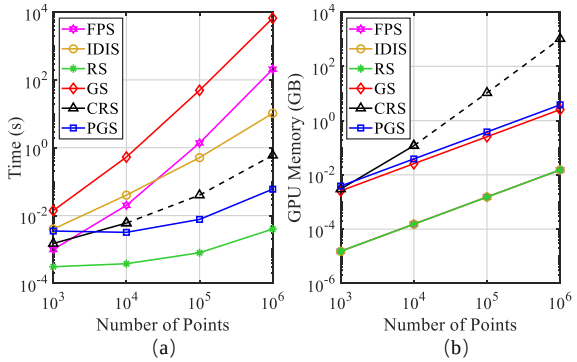


Figure 46: Time and memory consumption of different sampling approaches. The dashed lines represent estimated values due to the limited GPU memory.

to help align the research community's contributions with real-world selfdriving problems, we introduce a new large scale, high quality, diverse dataset. Our new dataset consists of 1150 scenes that each span 20 seconds, consisting of well synchronized and calibrated high quality LiDAR and camera data captured across a range of urban and suburban geographies. It is 15x more diverse than the largest camera+LiDAR dataset available based on our proposed diversity metric. We exhaustively annotated this data with 2D (camera image) and 3D (LiDAR) bounding boxes, with consistent identifiers across frames. Finally, we provide strong baselines for 2D as well as 3D detection and tracking tasks. We further study the effects of dataset size and generalization across geographies on 3D detection methods. Find data, code and more up-todate information at http://www.waymo.com/open.

3. ECCV

3.1. 2014

3.1.1 Microsoft COCO: Common Objects in Context [12]

Abract: We present a new dataset with the goal of advancing the state-of-the-art in object recognition by placing the question of object recognition in the context of the broader question of scene understanding. This is achieved by gathering images of complex everyday scenes containing common objects in their natural context. Objects are labeled using per-instance segmentations to aid in precise object localization. Our dataset contains photos of 91 objects types that would be easily recognizable by a 4 year old. With a total of 2.5 million labeled instances in 328k images, the creation of our dataset drew upon extensive crowd worker involvement via novel user interfaces for category detection, instance spotting and instance segmentation. We present a detailed statistical analysis of the dataset in comparison to PASCAL,

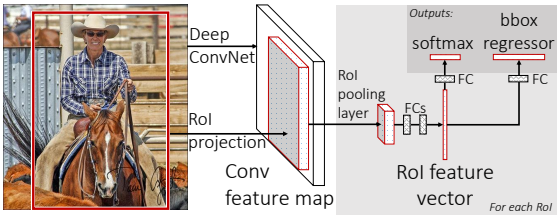


Figure 47: Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

ImageNet [21], and SUN. Finally, we provide baseline performance analysis for bounding box and segmentation detection results using a Deformable Parts Model.
URL: http://cocodataset.org/

4. ICCV

4.1. 2015

4.1.1 Fast R-CNN [5]

Abract: This paper proposes a Fast Region-based Convolutional Network method (Fast R-CNN) for object detection. Fast R-CNN builds on previous work [6] to efficiently classify object proposals using deep convolutional networks. Compared to previous work, Fast R-CNN employs several innovations to improve training and testing speed while also increasing detection accuracy. Fast R-CNN trains the very deep VGG16 network 9x faster than R-CNN, is 213x faster at test-time, and achieves a higher mAP on PASCAL VOC 2012. Compared to SPPnet, Fast R-CNN trains VGG16 3x faster, tests 10x faster, and is more accurate. Fast R-CNN is implemented in Python and C++ (using Caffe) and is available under the open-source MIT License at https://github.com/rbgirshick/fast-rcnn.
Contribution:

1. Higher detection quality (mAP) than R-CNN [6], SPPnet.
2. Training is single-stage, using a multi-task loss.
3. Training can update all network layers.
4. No disk storage is required for feature caching.

Figures: Figure 47, Figure 48

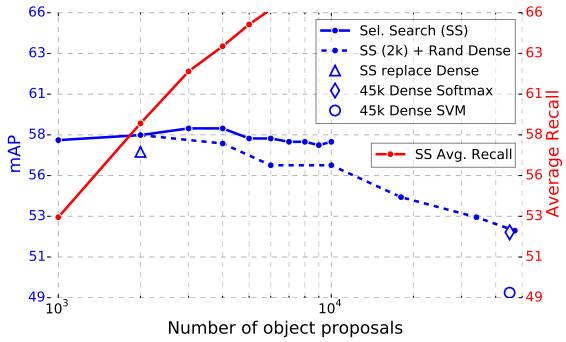


Figure 48: VOC07 test mAP and AR for various proposal schemes.

4.2. 2019

4.2.1 M3D-RPN: Monocular 3D Region Proposal Network for Object Detection [2]

Abract: Understanding the world in 3D is a critical component of urban autonomous driving. Generally, the combination of expensive LiDAR sensors and stereo RGB imaging has been paramount for successful 3D object detection algorithms, whereas monocular image-only methods experience drastically reduced performance. We propose to reduce the gap by reformulating the monocular 3D detection problem as a standalone 3D region proposal network. We leverage the geometric relationship of 2D and 3D perspectives, allowing 3D boxes to utilize well-known and powerful convolutional features generated in the image-space. To help address the strenuous 3D parameter estimations, we further design depth-aware convolutional layers which enable location specific feature development and in consequence improved 3D scene understanding. Compared to prior work in monocular 3D detection, our method consists of only the proposed 3D region proposal network rather than relying on external networks, data, or multiple stages. M3D-RPN is able to significantly improve the performance of both monocular 3D Object Detection and Bird's Eye View tasks within the KITTI [4] urban autonomous driving dataset, while efficiently using a shared multi-class model.
URL: https://github.com/garrickbrazil/M3D-RPN
Contribution:

1. The authors formulate a standalone monocular 3D region proposal network (M3D-RPN) with a shared 2D and 3D detection space, while using prior statistics to serve as strong initialization for each 3D parameter.

2. The authors propose depth-aware convolution to improve the 3D parameter estimation, thereby enabling the network to learn more spatially-aware high-level features.

Algorithm 2 Post 3D→2D Algorithm. The algorithm takes input of 2D / 3D box $b'_{2d}, [x, y, z]'_{\mathbf{P}}, [w, h, l, \theta]'_{3d}$, step size σ, termination β, and decay γ parameters, then iteratively tunes θ via L_1 corner consistency loss.

Input: $b'_{2d}, [x, y, z]'_{\mathbf{P}}, [w, h, l, \theta]'_{3d}, \sigma, \beta, \gamma$

$\rho \leftarrow$ box-project$([x, y, z]_{\mathbf{P}}, [w, h, l, \theta - \sigma]_{3d})$
$\eta \leftarrow L_1(b'_{2d},\ \rho)$
while $\sigma \geq \beta$ **do**
  $\rho^- \leftarrow$ box-project$([x, y, z]_{\mathbf{P}}, [w, h, l, \theta - \sigma]_{3d})$
  $\rho^+ \leftarrow$ box-project$([x, y, z]_{\mathbf{P}}, [w, h, l, \theta + \sigma]_{3d})$
  $loss^- \leftarrow L_1(b'_{2d},\ \rho^-)$
  $loss^+ \leftarrow L_1(b'_{2d},\ \rho^+)$
  if $\min(loss^-, loss^+) > \eta$ **then**
    $\sigma \leftarrow \sigma \cdot \gamma$;
  else if $loss^- < loss^+$ **then**
    $\theta \leftarrow \theta - \sigma$;
    $\eta \leftarrow loss^-$
  else
    $\theta \leftarrow \theta + \sigma$;
    $\eta \leftarrow loss^+$

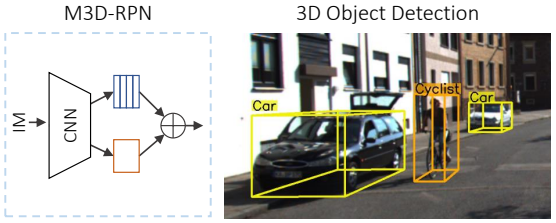


Figure 49: M3D-RPN uses a *single* monocular 3D region proposal network with global convolution (orange) and local depth-aware convolution (blue) to predict multi-class 3D bounding boxes.

3. The authors propose a simple orientation estimation post-optimization algorithm which use 3D projections and 2D detection to improve the θ estimation.

Algorithm: Algorithm 2
Figures: Figure 49, Figure 50, Figure 51

5. IJCV

5.1. 2015

5.1.1 ImageNet Large Scale Visual Recognition Challenge [21]

Abract: The ImageNet Large Scale Visual Recognition Challenge is a benchmark in object category classification and detection on hundreds of object categories and millions of images. The challenge has been run annually from 2010

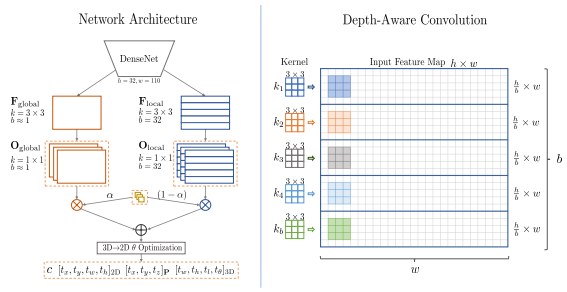


Figure 50: **Overview of M3D-RPN**. The proposed method consist of parallel paths for global (orange) and local (blue) feature extraction. The global features use regular spatial-invariant convolution, while the local features denote depth-aware convolution, as detailed right. The depth-aware convolution uses non-shared kernels in the row-space k_i for $i = 1 \dots b$, where b denotes the total number of distinct bins. To leverage both variants of features, we weightedly combine each output parameter from the parallel paths.

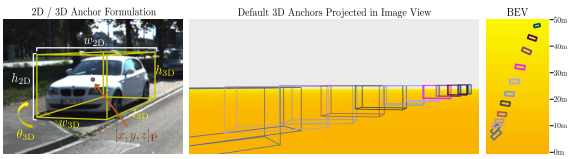


Figure 51: **Anchor Formulation** and **Visualized 3D Anchors**. We depict each parameter of within the 2D / 3D anchor formulation (left). We visualize the precomputed 3D priors when 12 anchors are used after projection in the image view (middle) and Bird's Eye View (right). For visualization purposes only, we span anchors in specific x_{3d} locations which best minimize overlap when viewed.

to present, attracting participation from more than fifty institutions. This paper describes the creation of this benchmark dataset and the advances in object recognition that have been possible as a result. We discuss the challenges of collecting large-scale ground truth annotation, highlight key breakthroughs in categorical object recognition, provide a detailed analysis of the current state of the field of large-scale image classification and object detection, and compare the state-ofthe-art computer vision accuracy with human accuracy. We conclude with lessons learned in the 5 years of the challenge, and propose future directions and improvements.

URL: http://www.image-net.org/

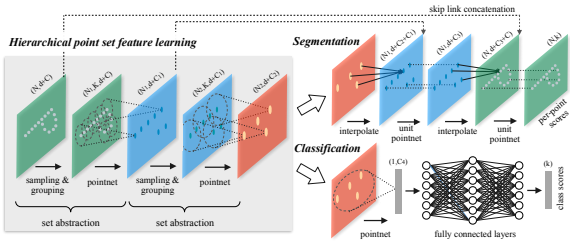


Figure 52: Illustration of our hierarchical feature learning architecture and its application for set segmentation and classification using points in 2D Euclidean space as an example. Single scale point grouping is visualized here. For details on density adaptive grouping, see Figure 53

6. NIPS

6.1. 2017

6.1.1 PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space [16]

Abract: Few prior works study deep learning on point sets. PointNet [15] is a pioneer in this direction. However, by design PointNet does not capture local structures induced by the metric space points live in, limiting its ability to recognize fine-grained patterns and generalizability to complex scenes. In this work, we introduce a hierarchical neural network that applies PointNet recursively on a nested partitioning of the input point set. By exploiting metric space distances, our network is able to learn local features with increasing contextual scales. With further observation that point sets are usually sampled with varying densities, which results in greatly decreased performance for networks trained on uniform densities, we propose novel set learning layers to adaptively combine features from multiple scales. Experiments show that our network called PointNet++ is able to learn deep point set features efficiently and robustly. In particular, results significantly better than state-of-the-art have been obtained on challenging benchmarks of 3D point clouds.

URL: https://github.com/charlesq34/pointnet2

Contribution:

1. Generate the overlapping partitioning of point set.
2. Abstract sets of points or local features through a local feature learner.
3. Leverage neighborhoods at multiple scales to achieve both robustness and detail capture.

Figures: Figure 52, Figure 53

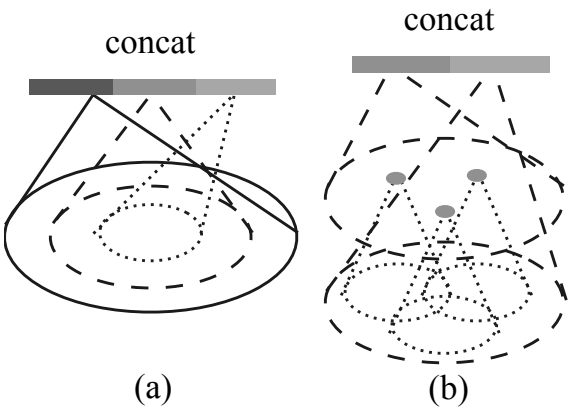


Figure 53: (a) Multi-scale grouping (MSG); (b) Multi-resolution grouping (MRG).

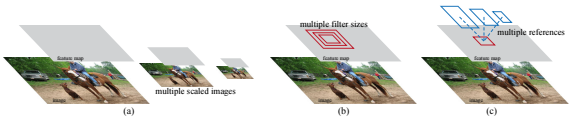


Figure 54: Different schemes for addressing multiple scales and sizes. (a) Pyramids of images and feature maps are built, and the classifier is run at all scales. (b) Pyramids of filters with multiple scales/sizes are run on the feature map. (c) We use pyramids of reference boxes in the regression functions.

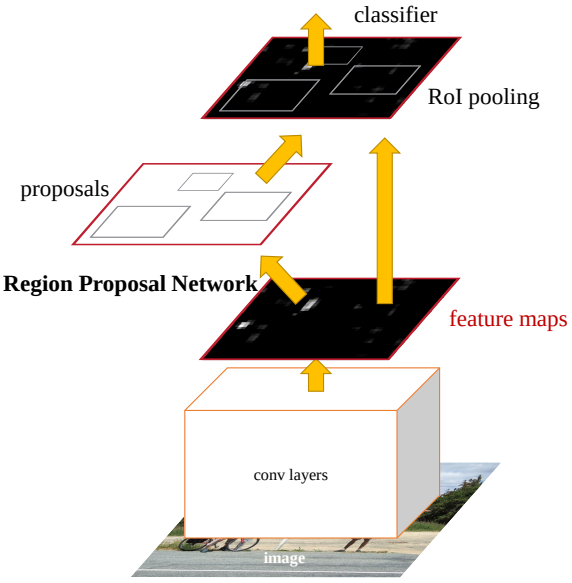


Figure 55: Faster R-CNN is a single, unified network for object detection. The RPN module serves as the 'attention' of this unified network.

7. T-PAMI

7.1. 2016

7.1.1 Region-Based Convolutional Networks for Accurate Object Detection and Segmentation [6]

Abract: Object detection performance, as measured on the canonical PASCAL VOC Challenge datasets, plateaued in the final years of the competition. The best-performing methods were complex ensemble systems that typically combined multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 50 percent relative to the previous best result on VOC 2012—achieving a mAP of 62.4 percent. Our approach combines two ideas: (1) one can apply high-capacity convolutional networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data are scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, boosts performance significantly. Since we combine region proposals with CNNs, we call the resulting model an R-CNN or Region-based Convolutional Network . Source code for the complete system is available at http://www.cs.berkeley.edu/~rbg/rcnn.

7.2. 2017

7.2.1 Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks [20]

Abract: State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN [5] have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work, we introduce a Region Proposal Network (RPN) that shares full-image convolutional features with the detection network, thus enabling nearly cost-free region proposals. An RPN is a fully-convolutional network that simultaneously predicts object bounds and objectness scores at each position. RPNs are trained end-to-end to generate highquality region proposals, which are used by Fast R-CNN for detection. With a simple alternating optimization, RPN and Fast R-CNN can be trained to share convolutional features. For the very deep VGG-16 model, our detection system has a frame rate of 5fps (including all steps) on a GPU, while achieving state-of-the-art object detection accuracy on PASCAL VOC 2007 (73.2% mAP) and 2012 (70.4% mAP) using 300 proposals per image. Code is available at https://github.com/ShaoqingRen/faster_rcnn.

Figures: Figure 54, Figure 55, Figure 56

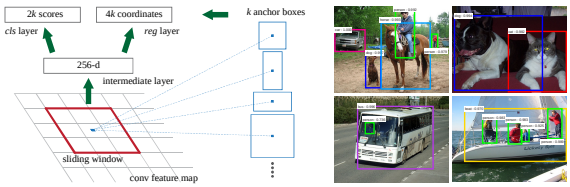


Figure 56: **Left**: Region Proposal Network (RPN). **Right**: Example detections using RPN proposals on PASCAL VOC 2007 test. Our method detects objects in a wide range of scales and aspect ratios.

7.3. 2019

7.3.1 The ApolloScape Open Dataset for Autonomous Driving and its Application [26]

Abract: Autonomous driving has attracted tremendous attention especially in the past few years. The key techniques for a self-driving car include solving tasks like 3D map construction, self-localization, parsing the driving road and understanding objects, which enable vehicles to reason and act. However, large scale data set for training and system evaluation is still a bottleneck for developing robust perception models. In this paper, we present the ApolloScape dataset and its applications for autonomous driving. Compared with existing public datasets from real scenes, e.g., KITTI [4] or Cityscapes, ApolloScape contains much large and richer labelling including holistic semantic dense point cloud for each site, stereo, per-pixel semantic labelling, lanemark labelling, instance segmentation, 3D car instance, high accurate location for every frame in various driving videos from multiple sites, cities and daytimes. For each task, it contains at lease 15x larger amount of images than SOTA datasets. To label such a complete dataset, we develop various tools and algorithms specified for each task to accelerate the labelling process, such as joint 3D-2D segment labeling, active labelling in videos etc. Depend on ApolloScape, we are able to develop algorithms jointly consider the learning and inference of multiple tasks. In this paper, we provide a sensor fusion scheme integrating camera videos, consumer-grade motion sensors (GPS/IMU), and a 3D semantic map in order to achieve robust self-localization and semantic segmentation for autonomous driving. We show that practically, sensor fusion and joint learning of multiple tasks are beneficial to achieve a more robust and accurate system. We expect our dataset and proposed relevant algorithms can support and motivate researchers for further development of multi-sensor fusion and multi-task learning in the field of computer vision.
URL: http://apolloscape.auto/

References

[1] Alex Bewley, Pei Sun, Thomas Mensink, Dragomir Anguelov, and Cristian Sminchisescu. Range conditioned dilated convolutions for scale invariant 3d object detection. *arXiv preprint arXiv:2005.09927*, 2020. 1, 9

[2] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9287–9296, 2019. 1, 20

[3] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020. 1, 9

[4] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1, 3, 4, 5, 6, 8, 9, 13, 14, 16, 17, 20, 23

[5] Ross Girshick. Fast r-cnn. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015. 1, 19

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):142–158, 2016. 2, 19, 22

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 1, 9, 15

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 10

[9] Qingyong Hu, Bo Yang, Linhai Xie, Stefano Rosa, Yulan Guo, Zhihua Wang, Niki Trigoni, and Andrew Markham. Randla-net: Efficient semantic segmentation of large-scale point clouds. *arXiv preprint arXiv:1911.11236*, 2019. 1, 17

[10] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12697–12705, 2019. 1, 3, 6, 14

[11] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 1, 10

[12] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014. 1, 4, 10, 11, 15, 19

[13] Chenxi Liu, Piotr Dollár, Kaiming He, Ross B. Girshick, Alan L. Yuille, and Saining Xie. Are labels necessary for neural architecture search. *arXiv preprint arXiv:2003.12056*, 2020. 1, 7

[14] Jiquan Ngiam, Benjamin Caine, Wei Han, Brandon Yang, Yuning Chai, Pei Sun, Yin Zhou, Xi Yi, Ouais Alsharif, Patrick Nguyen, Zhifeng Chen, Jonathon Shlens, and Vijay Vasudevan. Starnet: Targeted computation for object detection in point clouds. *arXiv preprint arXiv:1908.11069*, 2019. 1, 4

[15] Charles Ruizhongtai Qi, Hao Su, Mo Kaichun, and Leonidas J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 77–85, 2017. 1, 11, 14, 21

[16] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J. Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in Neural Information Processing Systems*, pages 5099–5108, 2017. 2, 14, 16, 21

[17] Guocheng Qian, Abdulellah Abualshour, Guohao Li, Ali K. Thabet, and Bernard Ghanem. Pu-gcn: Point cloud upsampling using graph convolutional networks. *arXiv preprint arXiv:1912.03264*, 2019. 1, 4

[18] Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. *arXiv preprint arXiv:2003.13678*, 2020. 1, 14

[19] Esteban Real, Chen Liang, David R. So, and Quoc V. Le. Automl-zero: Evolving machine learning algorithms from scratch. *arXiv preprint arXiv:2003.03384*, 2020. 1, 6

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017. 2, 11, 22

[21] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015. 1, 8, 10, 15, 19, 20

[22] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. *arXiv preprint arXiv:1912.13192*, 2019. 1, 16

[23] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–779, 2019. 1, 13

[24] Weijing Shi and Ragunathan Rajkumar. Point-gnn: Graph neural network for 3d object detection in a point cloud. *arXiv preprint arXiv:2003.01251*, 2020. 1, 16

[25] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam, Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov. Scalability in perception for autonomous driving: Waymo open dataset. *arXiv preprint arXiv:1912.04838*, 2019. 1, 3, 5, 6, 8, 9, 17, 18

[26] Peng Wang, Xinyu Huang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 2, 23

[27] Zining Wang, Di Feng, Yiyang Zhou, Wei Zhan, Lars Rosenbaum, Fabian Timm, Klaus Dietmayer, and Masayoshi Tomizuka. Inferring spatial uncertainty in object detection. *arXiv preprint arXiv:2003.03644*, 2020. 1, 7

[28] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 1, 3

[29] Yin Zhou, Pei Sun, Yu Zhang, Dragomir Anguelov, Jiyang Gao, Tom Ouyang, James Guo, Jiquan Ngiam, and Vijay Vasudevan. End-to-end multi-view fusion for 3d object detection in lidar point clouds. *arXiv preprint arXiv:1910.06528*, 2019. 1, 3