Self-Attention を用いた PointPillars による 3 次元物体検出の高精度化

三原一真 †　平川翼 †　山下隆義 †　藤吉弘亘 †
† 中部大学
E-mail: mhrkzm11@mprg.cs.chubu.ac.jp

1 はじめに

自動運転には，センサによって得られた周囲の環境から障害物を検出して回避を行うタスクが求められる．全方位 LiDAR は，レーザーを回転させながら全方位に照射し，3 次元点群データを獲得するセンサである．LiDAR センサによって得られる点群データは，深層学習のネットワークに入力することによって点群データからの物体検出やクラス分類が可能である．PointPillars [1] は点群を直接入力する物体検出手法であり，高精度な物体検出を実現している．しかし，自転車や歩行者の識別精度は，自動車の識別精度と比較して低いという問題点がある．そこで本研究では，PointPillars をベースとして Self-Attention Block [2] を導入し，小物体に対して高精度な物体検出手法を提案する．提案手法は，PointPillars の PillarFeatureNet の出力に対し Self-Attention Block の処理を行うことによって，pillar 間の特徴を補足することによる小物体の識別精度の向上を目的とする．

2 関連研究

2.1 点群データの入力方法

点群データの入力方法は，複数の入力方法がある．Voxel は，点群データを 3 次元座標系を格子状に区切り，格子の内部にある点群データの情報によってブロックのように変換をする方法 [3] である．Range View，Birds-Eye View は，点群データを特定の視点から疑似的な 2 次元画像に変換して入力する方法 [4] [5] である．Point Set は，点群データの変換を行わずネットワークに直接入力する方法 [6] [7] [8] がある．点群データを直接入力する方法はデータの変換を行わないため正確な情報を保持することが可能であり，処理が効率的である．

2.2 PointNet

PointNet [6] は，Point Set の点群データの入力方法を用いた物体検出手法であり，点群を直接入力して物体の Classification Network でクラス分類，Segmentation Network でセグメンテーションを行う手法である．点群データは，不定形なデータ形状や順序付けがされていないという特徴があり，画像に用いられる畳み込み処理が困難である．PointNet は，点群データからの物体検出を行う際の難点となる点群データの特徴に対応したネットワークである．しかし，空間の細部の情報や複雑な情報を捉えることが困難であるという問題点がある．

2.3 PointPillars

PointPillars [1] は，点群データを直接入力する手法であり，はじめに，PillarFeatureNet によって点群データを $x-y$ 平面から垂直に格子構造に沿って切り出したデータ (pillar) を入力し，それぞれの pillar 内部の持つ点の情報から点の次元を変換している．次に，Backbone によって 2 次元畳み込み処理を行うことによって特徴を抽出している．その後，Detection Head によってクラス分類と bounding box の位置，大きさ，向きの推定を行っている．PointPillars は自動車クラスの識別精度が高精度である．しかし，自転車，歩行者の識別精度は自動車の識別精度と比較して低精度であるという問題点がある．

3 提案手法

従来手法は，小物体に対して識別精度が低いという問題点がある．そこで本研究では，PointPillars [1] をベースとしたネットワーク構造に Self-Attention Block [2] を導入し，Pillar 間の特徴を補足することによって小物体への識別精度向上を目標とする．提案手法のネットワーク構造を図 1 に示す．

3.1 PillarFatureNet

PillarFeatureNet は，点群データを入力として特徴を抽出する．PillarFeatureNet のネットワーク構造を図 2 に示す．はじめに，点群データを $x-y$ 平面に対して垂直に柱状 (pillar) に切り出して入力を行う．次に，入力された pillar ごとに pillar 内部の点の平均からの距離，Pillar の中心からのオフセットを算出する．その後，pillar の点の持つ情報 D，pillar 内部の点の数 N，pillar の数 P から 3 次元のテンソルを作成する．作成したテンソルは簡易的な PointNet へ入力することによってチャンネル数 C，P，N のテンソルとなり，点の数 N の方向に最大値を取り C，P の 2 次元のテンソルへ

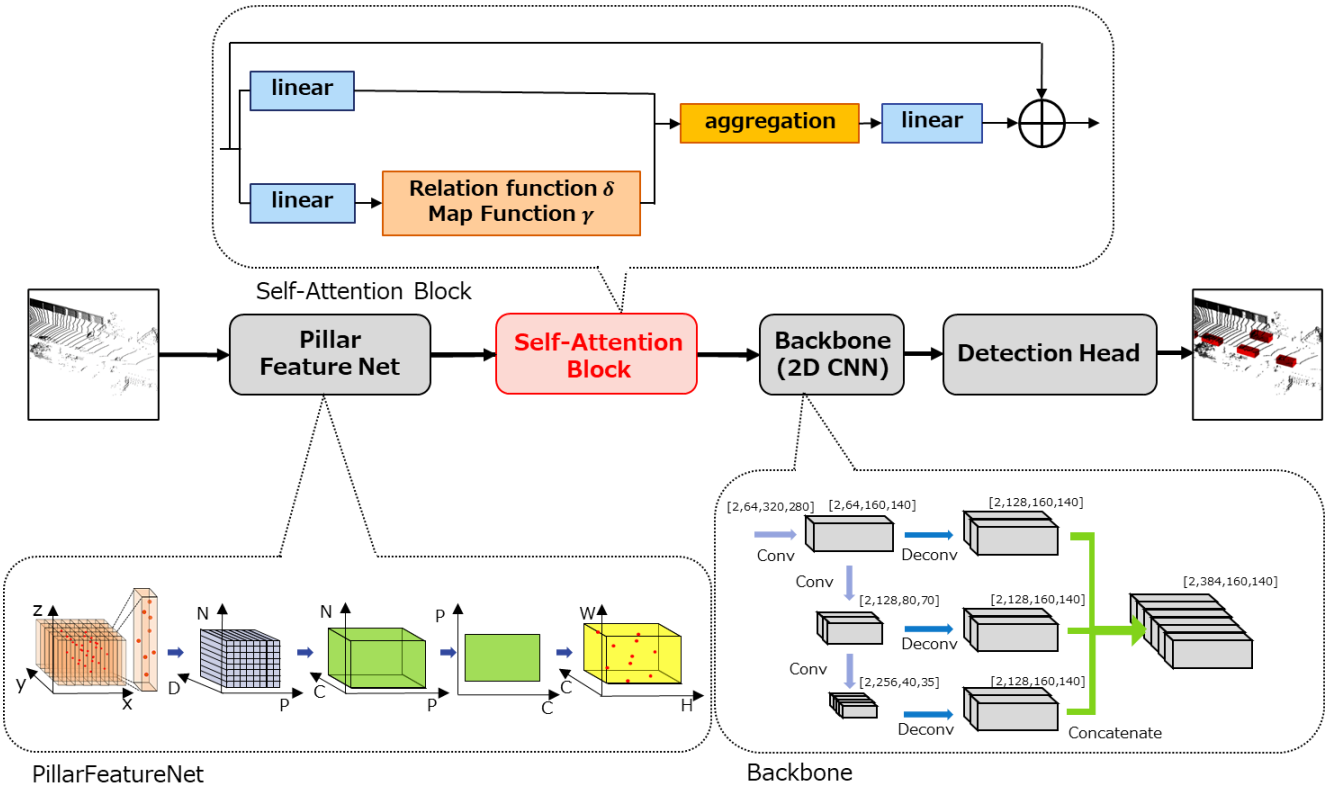


図 1: 提案手法のネットワーク構造

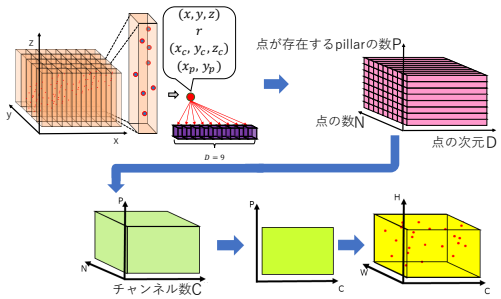


図 2: PillarFatureNet の構造

変換する．最後に，2 次元のテンソルを点群データの元の位置に頒布することによって点群データの大きさに対応する C，H，W の三次元のテンソルを出力とする．

3.2 Self-Attneton Block

Self-Attention Block [2] では，PillarFeatureNet の出力に対して小物体の特徴を抽出する．Self-Attention Block のネットワーク構造を図 3 に示す．Self-Attention Block の式を式 (1) に，ネットワーク構造を図に示す．Self-Attention Block は，はじめに入力されたベクトルを α と β の 2 つの関数に入力される．

$$y_i = \sum_{j \in R(i)} \alpha(x_{R(i)})_j \odot \beta(x_j) \tag{1}$$

式 1 において，i，j は特徴マップの位置であり，x は注目した重み，$R(i)$ は近傍の重みである．また，β は線

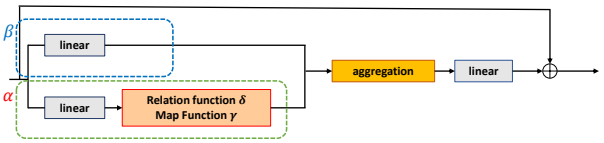


図 3: Self-Attneton Block の構造

形変換を行う関数であり，α は，以下の式 (2) のように表される．

$$\alpha(x_{R(i)}) = \gamma(\delta(x_{R(i)})) \tag{2}$$

式 (2) において，γ はマップ関数，δ は関係関数である．関係関数は，パッチ内の全てのベクトルの特徴を捉えて単一のベクトルを出力する．マップ関数は，β との加重和を行うためにベクトルの整形を行う．その後，α と β の加重和を取り，線形変換をかけた後入力ベクトルとの結合を行う．Self-Attention Block は，注目した重みとその周辺の重みから求めた重みに乗算する係数を決定するため，pillar 間の特徴を補足することが可能である．

3.3 Backbone

Backbone では，Self-Attention Block によって得られた特徴に 2 次元畳み込み処理を行う．Backbone のネットワーク構造を図 4 に示す．はじめに，3 層の 2 次元畳み込み処理を行う．次に，Deconvolution を 2 次元畳み込み処理のそれぞれの段階に対して行う．その

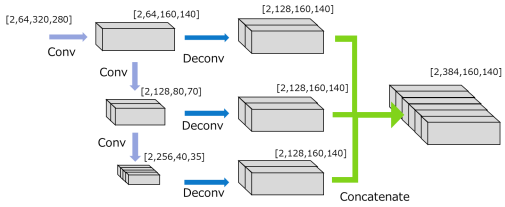


図 4: Backbone の構造

後，Deconvolution の出力を結合した特徴を出力としている．

4 評価実験

本実験では，提案手法で導入した Self-Attnetion Block [2] の有効性を確認する．評価実験では，PointPillars [1] と Self-Attention Block を導入した提案手法の定量的評価と定性的評価を行う．

4.1 実験概要

本実験では，提案手法に導入した Self-Attention Block の有効性を確認するために評価実験を行う．評価実験では，エポック数は 160，ミニバッチサイズは 2 に設定する．最適化手法は Adam，初期学習率は 0.0002 と設定し，15 epoch ごとに 0.8 倍する．損失関数は PointPillars[1] の損失関数を使用する．損失関数を式 (4.1) に示す．

$$L = \frac{1}{N_{pos}}(\beta_{loc}L_{loc} + \beta_{cls}L_{cls} + \beta_{dir}L_{dir}) \quad (3)$$

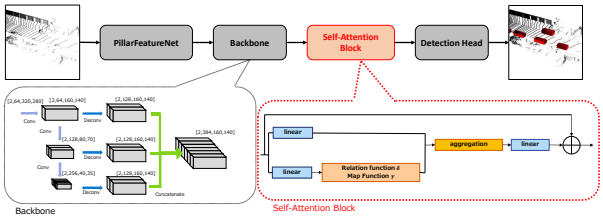
式 (4.1) において，N_{pos} は正解と判断した bounding box の数であり，β_{loc}，β_{cls}，β_{dir} は定数である．式 (4.1) 中の L_{dir} は bounding box の向きの損失関数である．また，式 (4.1) 中の L_{loc}, L_{cls} は，それぞれ bounding box の位置と大きさの損失関数とクラス分類の損失関数である．bounding box の位置と大きさの損失関数を式 (4)，クラス分類の損失関数を式 (5) に示す．

$$L_{loc} = \sum_{b\in(x,y,z,w,l,h,\theta)} SmoothL1(\Delta b) \quad (4)$$

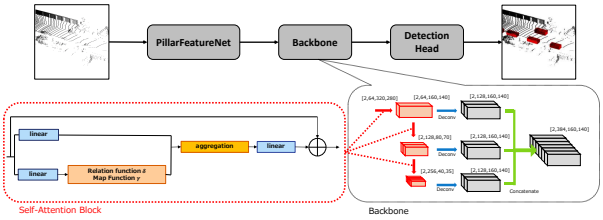
式において，x, y, z, w, l, h, θ は bounding box の位置，大きさ，向きである．

$$L_{cls} = -\alpha_a(1-p^a)^\gamma \log p^a \quad (5)$$

式 5 において，a は定数であり，p^a はクラス確率である．また，α と γ はそれぞれ 0.25，0.2 に設定する．また，Self-Attention Block の導入箇所を変更したネットワークにおいても評価実験を行う．ネットワーク構造を図 5 に示す．図 5(a) は，Self-Attention Block を Backbone の後に配置したネットワーク構造，図 5(b) は，Self-Attention Block を Backbone の畳み込み層を



(a) proposed 2



(b) proposed 3

図 5: Self-Attention Block の導入箇所を変更した構造

Self-Attention Block に置き換えたネットワーク構造である．データセットは，KITTI 3D Object Detection Benchmark [9] を使用する．データセットは学習用に 7,481 枚，評価用 7,518 枚のデータがあり，本実験では学習用データを 3,712 枚と 3,769 枚に分割し，それぞれ学習用と評価用にデータセットを使用する．また，データセットは，Car，Van，Truck，Pedestrian，Person (siting)，Cyclist，Tram，Misc の 8 つのクラスがあり，本実験では自動車，自転車，歩行者の 3 クラスの検出を行う．

4.2 実験結果

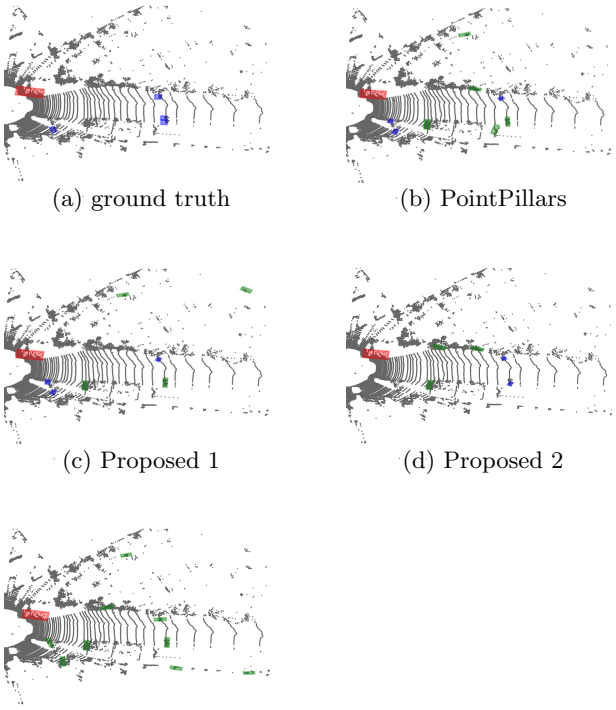
従来手法との精度を比較した定量的評価は，表 1 のようになった．また，Self-Attneiton Block [2] の入力箇所を変更したネットワーク構造との精度を比較した定量的評価は，表 2 のようになった．表 2 の proposed 1 は図 1 のネットワーク構造，proposed 2 は図 5(a) のネットワーク構造，proposed 3 は図 5(b) のネットワーク構造である． 表 1 から，提案手法は従来手法と比較して KITTI test 3D detection benchmark において mAP が 2.89pt 向上した．また，自転車の Moderate における AP が 6.65pt，歩行者の Moderate における AP が 0.90pt 向上した．また，表 2 から，PilllarFeatureNet の出力に対して Self-Attention Block の処理を行う proposed 1 が，他の Self-Attention Block の導入方法と比較して最も高精度であることが分かる．これらの結果から，Self-Attention Block の導入による pillar 間の関係性の補足が，自転車や歩行者のような小物体に対する識別精度の向上に有効性があると考えられる．また，定性的評価は，図 6 のようになった．表 6 の proposed 1 は図 1 のネットワーク構造，proposed 2 は図 5(a) のネットワーク構造，proposed 3 は図 5(b) のネットワーク構造であ

表 1: 従来手法と提案手法の精度比較

method	mAP	Car			Cyclist			Pedestrian		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
PointPillars	51.62	80.06	69.47	67.03	62.08	50.79	48.25	35.36	34.60	32.65
proposed	**54.51**	**81.99**	**70.58**	**67.25**	**69.49**	**57.44**	**54.50**	**36.76**	**35.50**	**33.35**

表 2: Self-Attention の導入方法と提案手法の精度比較

method	mAP	Car			Cyclist			Pedestrian		
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard
proposed 1	**54.51**	**81.99**	**70.58**	**67.25**	**69.49**	**57.44**	**54.50**	**36.76**	**35.50**	**33.35**
proposed 2	50.23	79.13	68.97	67.06	66.10	53.13	49.86	29.58	28.59	27.68
proposed 3	37.96	67.09	55.09	48.26	54.96	43.20	40.20	18.68	15.59	14.92



(a) ground truth (b) PointPillars

(c) Proposed 1 (d) Proposed 2

(e) Proposed 3

Car：Red　Cyclist：Green　Pedestrian：Blue

図 6: 従来手法と提案手法の定性的評価の比較

る．図 6 から，従来手法と比較して proposed 1 は誤識別が減少し識別精度が向上していることが分かる．

5 おわりに

本研究では，PointPillars をベースとしたネットワークに，小物体の識別精度を改善する狙いで周囲の重みから加重和を取り，個々の重みを算出可能な Self-Attention Block を導入することによって KITTI test 3D detection benchmark において自転車の Moderate における AP が 6.65pt，歩行者の Moderate における AP が 0.90pt 向上した．この結果から，Self-Attention Block の導入が，小物体の対して有効であることが考えられる．今後の予定として，Self-Attention Block の導入箇所の変更，Detection Head の変更を検討して識別精度の向上を目標とする．

参考文献

[1] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12697–12705, 2019.

[2] Hengshuang Zhao, Jiaya Jia, and Vladlen Koltun. Exploring self-attention for image recognition. In *CVPR*, 2020.

[3] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4490–4499, 2018.

[4] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 1907–1915, 2017.

[5] Martin Simony, Stefan Milzy, Karl Amendey, and Horst-Michael Gross. Complex-yolo: An euler-region-proposal for real-time 3d object detection on point clouds. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pp. 0–0, 2018.

[6] Charles R Qi, Hao Su, Kaichun Mo, and

Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 652–660, 2017.

[7] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, Vol. 30, , 2017.

[8] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 770–779, 2019.

[9] 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*, pp. 3354–3361. IEEE, 2012.