機械知覚&ロボティクスグループ
中部大学 工学研究科 情報工学専攻

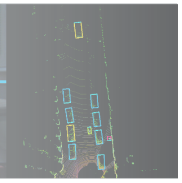
MPRG
MACHINE PERCEPTION AND ROBOTICS GROUP

2020年度　山下研究室　修士論文発表　アブストラクト

Deep Learning　Action Recognition　Graph Convolutional Networks

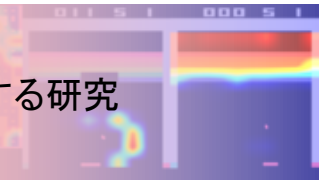
時空間のアテンション情報を考慮したGraph Convolutional Networksによる動作認識に関する研究
白木 克俊

Deep Learning　Object Detection　Point Clouds



Study of 3D Object Detection with　Normal-map on Point Clouds
繆 継樹

Deep Learning　Reinforcement Learning　Visual Explanation



Attention機構による視覚的説明を可能とした深層強化学習に関する研究
板谷 英典

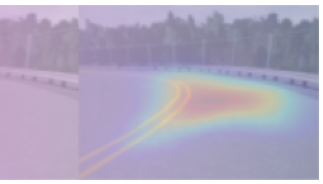
Deep Learning　Change Detection　Semantic Segmentation

セマンティックセグメンテーションによる超高解像度画像からの変化点検出に関する研究
筒井 駿吾

Deep Learning　Saliency Prediction

異なる解像度間の整合性を考慮した顕著性予測モデルに関する研究
瀬尾 俊貴

Deep Learning　Self-driving　Gaze



視線情報を活用した一貫学習ベースの自動運転制御に関する研究
森 啓介

時空間のアテンション情報を考慮した Graph Convolutional Networks による動作認識に関する研究

TP19008 白木克俊　　指導教授：山下隆義

1. はじめに

骨格データは人間の動きを直接捉えることができることや, 認識時における環境や視点の変化に対して頑健な利点がある. 骨格データを用いた Graph Convolutional Networks (GCN) による従来の動作認識では, 人間の骨格パターンでグラフ構造を事前に定義するため, 動作特有の関節間の関係性を考慮できない. また, 認識における関節の重要度も動作ごとに異なることが予想される. 本研究では, 関節の重要度と関係性を考慮して動作認識を行う Spatial Temporal Attention Graph Convolutional Networks (STA-GCN) を提案する. STA-GCN は, フレームごとの関節の重要度と, 動作ごとの関節間の接続関係を獲得する. 重要度を特徴マップに重み付けし, 接続関係を用いてグラフ畳み込み処理を行うことで, 重要度と関係性を考慮しつつ認識を行う. また, マルチモーダル学習において Mechanics-stream 構造を導入し, 高精度化を実現する.

2. 骨格データからの GCN を用いた動作認識

骨格データからの動作認識に, GCN を用いた手法である Spatial Temporal GCN (ST-GCN) [1] がある. ST-GCN は, 同一フレーム内の関節を結ぶ空間グラフと, 隣接フレームの同一関節を結ぶ時間グラフの 2 つのグラフ構造 (図 1) で骨格データを表現することにより高い認識精度を達成した. しかしながら, ST-GCN は, 空間グラフの接続関係 E_S を骨格パターン H で定義しているため, 動作特有の関節間の関係性を考慮できない. また, 動作や時間ごとに変化する関節の重要度も表現できないという問題がある.

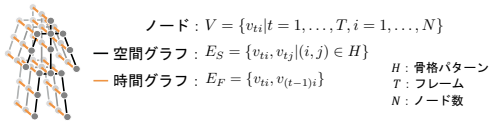


図 1: ST-GCN における骨格データのグラフ表現

3. 提案手法

関節の重要度と関係性を考慮して動作認識を行う Spatial Temporal Attention GCN (STA-GCN) を提案する. STA-GCN は関節の重要度である Attention node と, 関節間の関係性である Attention edge を獲得する. 獲得した Attention を考慮しつつ認識を行う. また, マルチモーダル学習において Mechanics-stream 構造を提案する. Mechanics-stream 構造は, 各モーダルが持つ力学的特性や値のスケールの違いにもとづいてネットワークを構築する.

3.1 STA-GCN のネットワーク構造

図 2 に STA-GCN のネットワーク構造を示す. STA-GCN は, Feature extractor, Attention branch, Perception branch の 3 つのモジュールで構成する. 各モジュールに含まれる STGC-block では, 空間グラフと時間グラフのグラフ畳み込み処理を行う. Feature extractor には 2 つのモーダルを入力し, 人間の骨格パターン $\mathbf{A}^{skeleton}$ を空間グラフとする複数の STGC-block により各モーダルの特徴マップを獲得し, 結合する. Attention branch は, 関節の重要度を表す Attention node と, 関節間の重要な関係性を表す Attention edge を特徴マップから獲得する. Perception branch は, Attention node と Attention edge を考慮しつつ, 最終的なクラス確率を出力する. 学習は, 2 つのブランチからの出力に対してクロスエントロピー誤差で損失を求め, 損失の和をネットワークの学習誤差として用いる.

3.2 Spatial Temporal Attention Graph

Attention node $M(\mathbf{X}_{out}^{FE})$ は関節 × フレームの 2 次元マップであり, 動作特有の関節の重要度をフレームごとに表現できる. 獲得した Attention node は Attention 機構を用いて Feature extractor からの特徴マップ $\mathbf{X}_{out}^{FE}$ へ反映し, Perception branch の入力とする. 式 (1) に Attention 機構を示す.

$$\mathbf{X}'_{out} = M(\mathbf{X}_{out}^{FE}) \cdot \mathbf{X}_{out}^{FE} \tag{1}$$

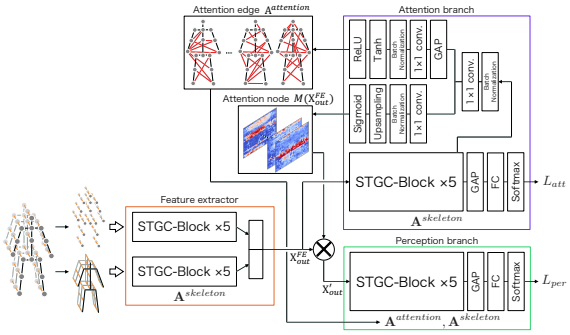


図 2: Spatial Temporal Attention GCN

Attention edge は, 動作ごとの関節間の重要な関係性を表す隣接行列である. 獲得した Attention edge は Perception branch に与える. Perception branch では, Attention edge $\mathbf{A}^{attention}$ と人間の骨格パターン $\mathbf{A}^{skeleton}$ によるグラフ畳み込み処理を行う. Perception branch における入力特徴 $\mathbf{X}_{in}$ に対する空間グラフ畳み込み処理を式 (2) に示す. $\mathbf{M}^{skel}, \mathbf{W}^{skel}, \mathbf{W}^{att}$ は重み行列である. Q は hop 数であり Q 個離れた関節までを結ぶ. K は動作ごとに生成する Attention edge の数である. 本研究では $Q = 3, K = 4$ とする.

$$\mathbf{X}_{out} = \sum_{q}^{Q} \mathbf{M}_q^{skel} \circ \hat{\mathbf{A}}_q^{skel} \mathbf{X}_{in} \mathbf{W}_q^{skel} + \sum_{k}^{K} \hat{\mathbf{A}}_k^{att} \mathbf{X}_{in} \mathbf{W}_k^{att} \tag{2}$$

Attention node と Attention edge によって時間的, 空間的な特徴を強調でき, 2 つの Attention を組み合わせたグラフ表現を Spatial Temporal Attention Graph (Attention graph) と呼ぶ.

3.3 Mechanics-stream 構造

Mechanics-stream 構造 (図 3) は, マルチモーダル学習におけるネットワーク構造である. 本研究では, 関節座標から関節速度, 関節加速度, 骨情報, 骨速度, 骨加速度を算出しネットワークの入力とする. 骨情報は, 関節間の距離であり関節間の方向も表現する. 座標や骨情報は空間的な位置の情報, 速度や加速度は時間的な移動量の情報であり, 各モーダルが持つ特性には違いがある. また, 座標や速度, 加速度は値のスケールが異なるため, 同じネットワークでの学習が困難である. そこで, 各モーダルが持つ力学的特性や値のスケールの違いにもとづいた Mechanics-stream 構造を提案する. Mechanics-stream 構造は 3 つのネットワークを用意し, 特性や値のスケールが近いモーダルを入力とする. 各ネットワークは独立して学習を行い, 各ネットワークから得られるクラス確率をクラスごとに合計することで最終的なクラス確率とする.

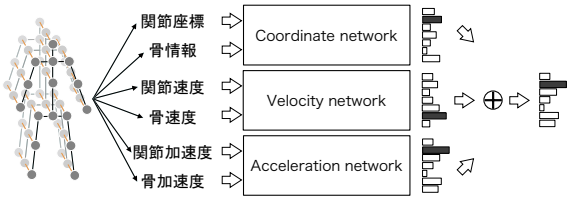
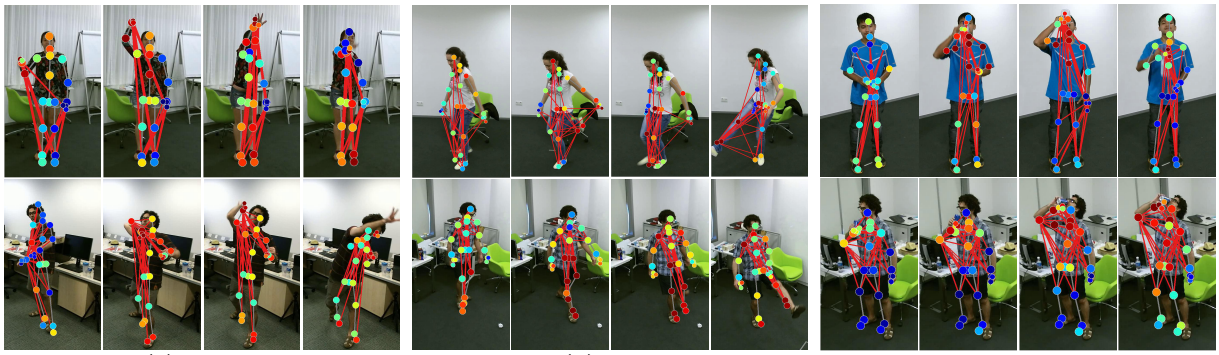


図 3: Mechanics-stream 構造

4. 評価実験

提案手法の有効性を確認するために評価実験を行う. データセットには NTU-RGB+D [2] と NTU-RGB+D120 [3] を用いる. NTU-RGB+D は関節数 25 の 3 次元座標 (X, Y, Z) を持つ動作認識用のデータセットであり, 動作クラス数は 60 である. 評価方法は, 40 名の被験者のデータを学習と検証で分ける Cross subject (x-sub) と, 3 方向から撮影したデータを学習と検証に分ける Cross view (x-view) があ



(a) Throw (b) Kick (c) Drink

図 4: **Attention graph の可視化**: 関節の色は Attention node にもとづいており, 赤いほど重要度が高く, 青いほど重要度が低い. 赤い線は Attention edge を示しており重みの大きい上位 30 本のみを描画している.

表 3: **モーダルの組み合わせの違いによる認識精度 [%]**: 単一のネットワークの認識精度 (Ind. net) と stream 構造を構築したときの認識精度 (X-stream). 評価は NTU-RGB+D の Cross subject.

Input data	w/o Attention		w/ Attention	
	Ind. net	X-stream	Ind. net	X-stream
Joint (coordinate, velocity)	86.0	87.1	86.2	87.2
Bone (coordinate, velocity)	86.1		86.3	
Joint (coordinate, velocity, acceleration)	86.7	87.8	85.7	86.7
Bone (coordinate, velocity, acceleration)	86.7		85.8	
Coordinate (joint, bone)	87.5	89.3	88.6	**90.1**
Velocity (joint, bone)	85.6		87.0	
Coordinate (joint, bone)	87.5	89.1	88.6	89.4
Velocity (joint, bone)	85.6		87.0	
Acceleration (joint, bone)	74.6		77.2	

表 1: 従来手法との認識精度の比較 [%]

NTU-RGB+D			NTU-RGB+D120		
Methods	x-sub	x-view	Methods	x-sub	x-setup
ST-GCN	81.5	88.3	2s-ALSTM	61.2	63.3
AS-GCN	86.8	94.2	BPEM	64.6	66.9
2s-AGCN	88.5	95.1	SkelMotion	67.7	66.9
AGC-SLTM	89.2	95.0	TSRJI	67.9	62.8
DGNN	89.9	**96.1**	ST-GCN	72.8	75.4
STA-GCN	**90.1**	95.8	STA-GCN	**83.9**	**86.5**

表 2: **Attention graph の従来手法への適用結果 [%]**: 評価は NTU-RGB+D の Cross subject.

Attention edge	Attention node	ST-GCN	AS-GCN	2s-AGCN
×	×	81.5	86.8	88.5
✓	×	84.1	86.9	89.2
×	✓	82.8	86.8	89.1
✓	✓	**84.8**	**87.0**	**89.3**

る. NTU-RGB+D120 は NTU-RGB+D に 60 の動作クラスを追加した大規模なデータセットである. 評価方法は, Cross subject と, カメラの高さや被験者との距離によって割り当てられた ID を用いてデータを学習と検証に分ける Cross setup (x-setup) がある.

4.1 従来手法との認識精度の比較

表 1 に, 従来手法との認識精度の比較結果を示す. どちらのデータセットにおいても提案手法の認識精度は, 従来手法と比較して認識精度の向上, または同等の認識精度を達成した. STA-GCN は, 動作特有の特徴を強調することで類似した動作でも異なる特徴を獲得できるため, 認識精度の向上に貢献したといえる.

4.2 Attention graph の可視化

STA-GCN により獲得した Attention graph を図 4 に示す. 投げる動作 (図 4(a)) は, エッジが右腕に集中している. 右腕の重要度は動作中に徐々に高くなり, 投げ終わると重要度が低くなる. 蹴る動作 (図 4(b)) は, 蹴り出す脚に対してエッジが集中する. このことから, 動作特有の Attention edge を獲得したといえる. 飲む動作 (図 4(c)) は投げると同様に右腕を重要視している. しかしながら, 飲む動作の脚の重要度は一貫して低いのに対し, 投げる動作は体重移動を行うため脚の重要度が高くなることから, 動作特有の Attention node を獲得したといえる.

4.3 Ablation study

Attention graph の従来手法への適用: 表 2 に, Attention graph を従来手法へ適用したときの認識精度を示す. 表 2 から, Attention edge, および Attention node のみを適用したほぼ全ての場合で認識精度の向上を確認できる. また, 全ての従来手法において Attention edge と Attention node の両方を適用した場合に最も認識精度が高くなる. この結果から, Attention graph は認識精度の向上に貢献しており, 両方を同時に適用した場合に最も効果的である.

モーダルの組み合わせの違いによる認識精度: 表 3 に, 入力モーダルの違いによる認識精度を示す. 表 3 から, 座標と速度を同じネットワークで学習するより, 別々のネットワークで学習した方が認識精度が高くなる. このことから, Mechanics-stream 構造の有効性を確認できる. しかしながら, 加速度を入力する場合では認識精度が低下した. 加速度は値のスケールが小さく, 認識のための十分な特徴が含まれていないことが考えられる.

5. おわりに

本研究では, 関節の重要度と関係性を考慮して動作認識を行う Spatial Temporal Attention GCN を提案した. また, マルチモーダル学習におけるモーダルの特性の違いにもとづいてネットワーク構築する Mechanics-stream 構造を提案した. 評価実験では, 提案手法が従来手法を超える認識精度を達成し有効性を確認した. また, 動作特有の Attention graph が獲得することを確認した. 今後の課題として, 動作認識以外のタスクへの応用などがあげられる.

参考文献

[1] S. Yan, *et al.*, "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition", AAAI, 2018.

[2] S. Amir, *et al.*, "NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis", CVPR, 2016.

[3] L. Jun, *et al.*, "NTU RGB+D 120: A Large-Scale Benchmark for 3D Human Activity Understanding", TPAMI, 2019.

研究業績

[1] K. Shiraki, *et al.*, "Spatial Temporal Attention Graph Convolutional Networks with Mechanics-Stream for Skeleton-based Action Recognition", ACCV, 2020.
(他 3 件)

1. Introduction

Object detection is one of the most crucial tasks in autonomous driving. In this task, both accuracy and speed are important. There are various methods have been studied to improve accuracy while accelerating the detection speed. Although RGB images captured by cameras are used for object detection, point clouds captured by LiDAR are also used for this task. Because it is insensitive to visible light and can capture objects day or night. However, point clouds are sparse and different from images in that the order is irregular, leading to a slow processing speed as 2D convolution cannot be performed directly. Surface normals extracted from the points have a better ability to represent the shape features of the object than 3D coordinates, which we believe will improve the performance of object detection. In this study, we propose a novel point clouds-based 3D object detection method for achieving higher-accuracy. The proposed method employs You Only Look Once v4 (YOLOv4) as a feature extractor and gives Normal-map as additional input. Our Normal-map is a three channels Bird-eye view (BEV) map, retaining detailed surface normal vectors. It makes the input information have more enhanced spatial shape information and can be associated with other hand-crafted features easily.

2. Bird-eye View Representation

The BEV map represents the point clouds as a 2D pseudo-image from the bird-eye view as shown in Figure 1. This approach converts the unordered point clouds into a sequence ordered image. Conventional methods are to generate three maps by a mapping function $\mathcal{P}_{\Omega i \to j}$, representing normalized point cloud density (R), maximum height (G), and maximum reflection intensity (B), as

$$\begin{aligned} z_r\left(\mathcal{S}_j\right) &= \min(1.0, \log(N+1)/64)\, N = \left|\mathcal{P}_{\Omega i \to j}\right|, \\ z_g\left(\mathcal{S}_j\right) &= \max\left(\mathcal{P}_{\Omega i \to j} \cdot [0,0,1]^T\right), \\ z_b\left(\mathcal{S}_j\right) &= \max\left(I\left(\mathcal{P}_{\Omega i \to j}\right)\right). \end{aligned} \tag{1}$$

Since 2D convolution can be applied to the BEV map, the detection task can be accelerated by using a fast object detection network such as YOLO[1]. However, different points may be arranged to a same pixel in the BEV map, which is less expressive than original data. Besides, the object shape will be lost due to the 3D data is compressed to 2D.

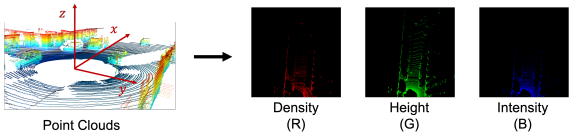


Figure 1 : Bird-eye View Representation.

3. Proposed Method

We propose a method for 3D object detection using BEV map with additional normal information. Figure 2 shows an overview of the proposed method.

3.1 Normal Feature Extraction

The normal vector is estimated from the pre-processed point clouds by Principal Component Analysis (PCA) with the search radius of 30cm and the maximum search number of 50 points. To make all normals point in the same direction, the normals opposing the LiDAR are reversed by an orienting system. Normal-map is generated for enabling the 2D convolution of each point's normals. The mapping function $f_{\mathcal{PS}}$ shown in Eq. (2) is used for creating the Normal-map, which allows us to use the normal vector $\vec{x}$, $\vec{y}$, $\vec{z}$ of the highest point in $\mathcal{P}_{\Omega i \to j}$ when mapping each point into 2D space.

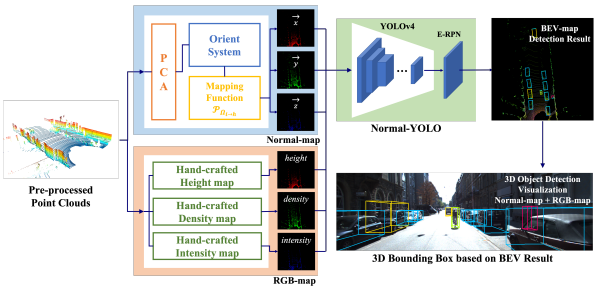


Figure 2 : Overview of Proposed Method.

$$\mathcal{P}_{\Omega i \to j} = \left\{ \mathcal{P}_{\Omega i} = [x, y, z]^T \middle| \mathcal{S}_j = f_{\mathcal{PS}}(\mathcal{P}_{\Omega i}, g) \right\} \tag{2}$$

As shown in Eq. (3), the normal vectors of each point are extracted as $normal_{\vec{x}}$, $normal_{\vec{y}}$, $normal_{\vec{z}}$ from the $\mathcal{P}_{\Omega_{i \to h}}$ for each axis. The normal vectors are represented by a 3-channel 2D pseudo image, as

$$\begin{aligned} normal_{\vec{x}}(\mathcal{S}_j) &= \vec{x}\left(\mathcal{P}_{\Omega j \to h}\right), \\ normal_{\vec{y}}(\mathcal{S}_j) &= \vec{y}\left(\mathcal{P}_{\Omega j \to h}\right), \\ normal_{\vec{z}}(\mathcal{S}_j) &= \vec{z}\left(\mathcal{P}_{\Omega j \to h}\right). \end{aligned} \tag{3}$$

Since the search range of the normal estimation is wider than the pixel representation range of the BEV map, it can include a wider range of information. Moreover, since the normal-map calculated from the normal vectors is also a BEV map, it can be freely combined with other BEV maps to be used as the input data.

3.2 Input Details

Our network uses RGB-map and Normal-map as input. The RGB-map is similar to the BirdNet[2], and consists of the height map, the density map, and the intensity map. The Normal-map is the normal vectors in the x, y, and z axes. Thus, the input is a 6-channel BEV map consisting of these maps concatenated in the channel axis.

3.3 Object Detection Network

The network predicts the class and size of an object with the Euler-Region Proposal Network (E-RPN) for 3D object detection as shown in Figure 3. We employ YOLOv4 as the basis network for 3D object detection. E-RPN predicts the height, width, angle, objectness, and class probability of the bounding box coordinates. In this study, the number of object classes is 6: Car, Van, Truck, Person, Cyclist, and Tram. The loss function is based on Mean-Square Error.

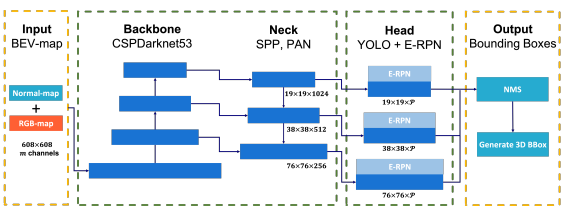


Figure 3 : Normal-YOLO Network Architecture.

4. Evaluation Experiments

In order to examine the effectiveness of the proposed method, some comparison experiments are conducted using the KITTI dataset. We also evaluate the accuracy of the object angle detection by adding a function to calculate the yaw angle.

Table 1 : Evaluation Results for Bird-eye View Performance on the KITTI Benchmark.

Method	FPS	Car			Pedestrian			Cyclist			Average
		Easy	Mod.	Hard	Easy	Mod.	Hard	Easy	Mod.	Hard	
BirdNet	9.1	**84.17**	59.83	57.35	**28.20**	**23.06**	**21.65**	**58.64**	**41.56**	**36.94**	45.93
Complexer-YOLO	16.7	77.24	68.96	64.95	21.42	18.26	17.06	32.00	25.40	22.88	38.68
Ours	5.5	72.84	**71.52**	**67.50**	26.71	21.19	20.17	42.50	36.06	31.18	43.30

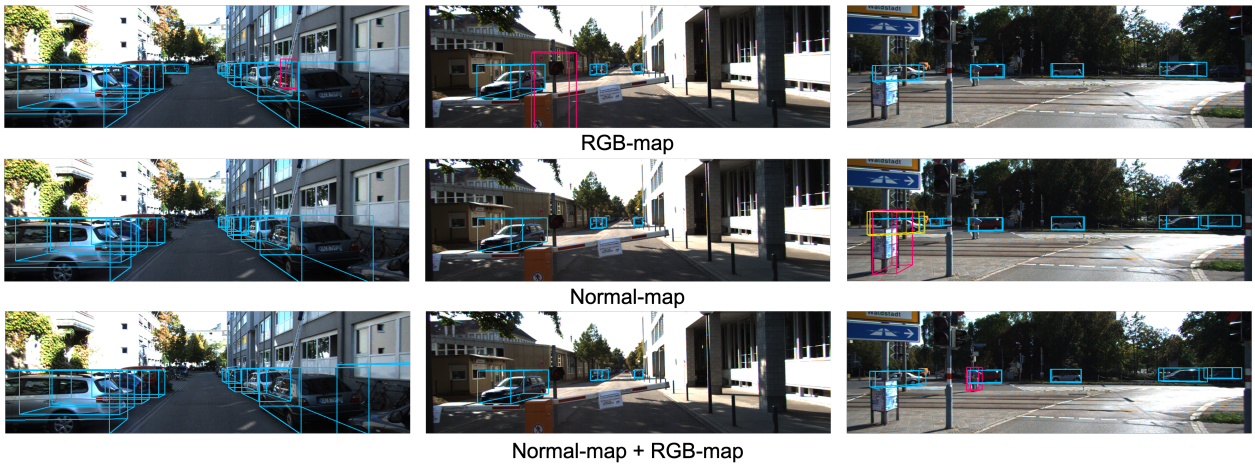


Figure 5 : 3D Object Detection Visualization in Camera View.

4.1 KITTI Benchmark Evaluation Results

Table 1 shows the evaluation results. Compared with the conventional method, the proposed method achieves the detection accuracy of 72.84% in *Car* under the Easy mode, and the highest accuracy of 67.50% at the Hard mode. It achieves almost the same accuracy as BirdNet[2], which is also a BEV-based method. Even for the same class objects with different detection modes, our method shows a more robust performance by adding normal information. In addition, we achieve a higher Average Precision (AP) than Complexer-YOLO[3], which uses the same YOLO-based network. Although the input is BEV map, the accuracy for non-planar objects (e.g. pedestrians and cyclists) is better for each mode.

4.2 Evaluation of Angle Prediction

Since the normal is the object shape information, we assume that object angle accuracy can be improved by adding the Normal-map. Table 2 shows the result of calculating the average included angle θ_k from the estimated object angles and ground truth for 6 classes.

$$score_{class}\left(\theta_k\right) = \left(\frac{1}{n}\sum_{k=1}^{n}\arccos\theta_k\right)^{-1} \tag{4}$$

Normal-map improves the angle accuracy. In particular, the accuracy of objects with large and flat shapes like cars is further improved.

Table 2 : Yaw Angle Prediction of 6 Classes.

Input Map	Score of Angle Accuracy					
	Car	Van	Truck	Person	Cyclist	Tram
RGB	10.18	**7.49**	5.78	**2.22**	**4.44**	3.24
Normal	8.76	5.37	5.35	1.69	3.10	3.28
Normal+RG	9.27	6.43	**10.09**	2.14	3.34	3.55
Normal+RGB	**10.34**	6.57	8.89	2.06	3.86	**5.25**

4.3 Evaluation by Distance

Since the density of the point clouds changes depending on the distance, we evaluate the detection accuracy by distance. Figure 4 shows average precision over distance. From Figure 4, the detection accuracy of the group with added Normal-map does not decrease up to $30m$, and the decrease is smaller than no-normal group even at $40m$ or more.

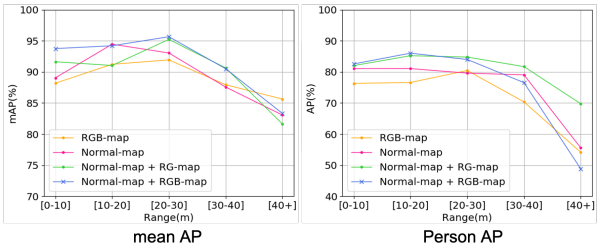


Figure 4 : Average Precision over distance.

4.4 Visualization Results

As shown in Figure 5, due to traffic lights, and traffic signs are cylindrical and have a height similar to a human, the no-normal group could easily make false positive prediction. In contrast, proposed method reduce the number of such mistakes with the addition of normal information. This indicates that the Normal-map is useful in avoiding the false detection of objects similar to human features.

5. Conclusion

We proposed a novel 3D object detection method with Normal-map on point clouds. We have confirmed that the accuracy of BEV map-based object detection is further improved when we introduced normal information to the BEV map. Since the Normal-map can keep high detection accuracy without intensity information, it is possible to use synthesized datasets by simulator with the virtual environment. In the future, to improve the accuracy of the method, we would like to explore deep learning methods for object normal estimation.

References

[1] A. Bochkovskiy, *et al.*, "YOLOv4: Optimal Speed and Accuracy of Object Detection", arXiv, 2020.

[2] J. Beltran, *et al.*, "BirdNet: a 3D Object Detection Framework from Lidar Information", ITSC, 2018.

[3] M. Simon, *et al.*, "Complexer-YOLO: Real-time 3D Object Detection and Tracking on Semantic Point Clouds", CVPR, 2019.

Research Achievements

[1] J. Miao, *et al.*, "3D Object Detection with Normal-map on Point Clouds", VISAPP, 2021.

(Other: 1 conference presentation)

Attention 機構による視覚的説明を可能とした深層強化学習に関する研究

TP19001 板谷英典　　指導教授：山下隆義

1. はじめに

深層強化学習は，エージェントが環境とのインタラクションを通じて得る報酬を頼りに，状況に応じて最適な行動選択するための方策を学習する手法である．方策はネットワークで表現されており，ネットワーク内部の演算は複雑である．そのため，エージェントの行動選択に対して判断根拠を解析することは非常に困難である．

本研究では，深層強化学習の代表的な手法である Asynchronous Advantage Actor-Critic (A3C) [1] に Attention 機構を導入した Mask-Attention A3C (Mask A3C) を提案する．Mask A3C では，方策と状態価値に対するエージェントの注視領域を表現した Mask-attention を獲得できる．推論時に Mask-attention を可視化することで，エージェントの行動選択に対する視覚的説明の実現を目的とする．また，Attention 機構の導入により，Mask-attention を考慮して学習することで，エージェントの性能向上を図る．

2. Asynchronous Advantage Actor-Critic

A3C [1] は，Asynchronous と Advantage を導入した Actor-Critic 法ベースの深層強化学習手法である．Asynchronous は複数環境における非同期でのパラメータ更新，Advantage は数ステップ先の報酬を考慮した学習である．A3C のネットワーク構造は，畳み込み層により特徴マップを抽出する Feature extractor，方策を出力する Policy branch，状態価値を出力する Value branch から構成される．方策はある状態において選択する行動の確率分布である．状態価値はある状態における報酬の期待値であり，ある状態にいることの価値を表す．

3. Mask-Attention A3C

エージェントの行動選択に対する判断根拠を解析するため，A3C に Attention 機構を導入した Mask-Attention A3C (Mask A3C) を提案する．Mask A3C では，Policy branch と Value branch に対し Attention 機構を導入することで，各ブランチの出力に対して注視した領域を表す Mask-attention を獲得する．推論時における各ブランチの Mask-attention を可視化することで，方策と状態価値の異なる 2 つの視点から，エージェントの行動選択に対する視覚的説明を実現する．

3.1 ネットワーク構造

図 1 に提案する Mask A3C のネットワーク構造を示す．Mask A3C は，Feature extractor，Policy branch，Value branch，Attention 機構から構成される．従来の A3C では，時系列情報を考慮するため，Feature extractor に LSTM を用いる．しかし，Mask A3C に LSTM を導入すると，入力画像に対する空間情報が欠落するため，Mask-attention を獲得できない．そこで，時空間情報を考慮できる Convolutional LSTM (ConvLSTM) [2] を導入する．ブランチ i の Mask-attention M_i は，Feature extractor から出力される特徴マップ F_i に対し，$1 \times 1 \times$ # of chanels の畳み込み層と Sigmoid 関数を適用することで獲得する．

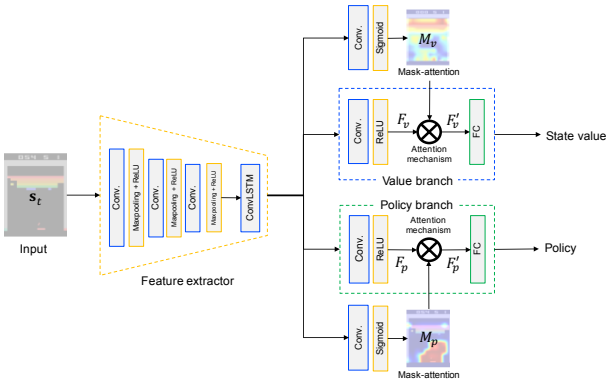


図 1: Mask A3C のネットワーク構造

3.2 Attention 機構

Mask A3C では，Policy branch と Value branch に Attention 機構を導入する．これにより，獲得した Mask-attention M_i を考慮し，方策及び状態価値を出力する．Attention 機構は，ブランチ i 内における中間層の特徴マップ F_i に対し，Mask-attention M_i を用いてマスク処理を行う．特徴マップに対する Mask-attention を用いたマスク処理を式 (1) に示す．ここで，$\mathbf{s}_t$ は入力である現状態，$F_i(\mathbf{s}_t)$ はブランチ i 内における中間層の特徴マップ，$M_i(\mathbf{s}_t)$ はブランチ i における Mask-attention，$F'_i(\mathbf{s}_t)$ はマスク処理後の特徴マップである．

$$F'_i(\mathbf{s}_t) = F_i(\mathbf{s}_t) \cdot M_i(\mathbf{s}_t) \tag{1}$$

4. 評価実験

Mask A3C の有効性を確認するため，OpenAI Gym [3] のゲームタスクを用いて評価実験を行う．

4.1 実験概要

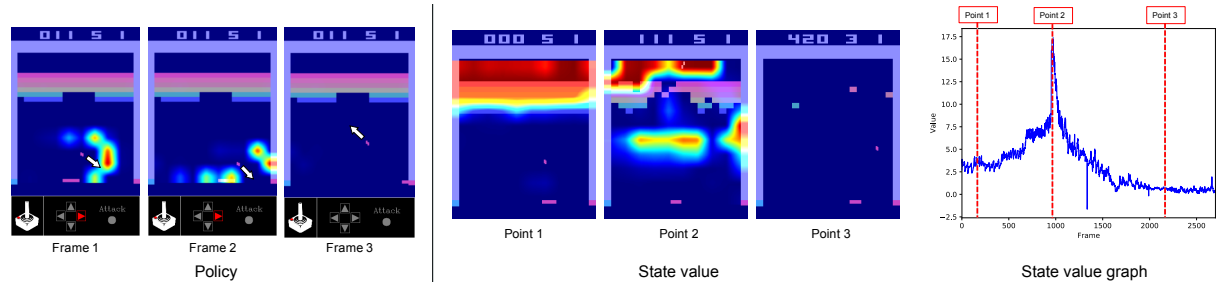
使用するゲームは，Breakout と Ms.Pac-Man，Space Invaders の 3 種類である．比較手法は，A3C と Mask A3C，各ブランチのみに対して Attention 機構を導入した Mask A3C (Policy Mask A3C，Value Mask A3C) の計 4 つである．入力はゲーム画面のグレースケール画像とし，出力である行動は各ゲームにおける操作とする．学習条件は worker 数を 35，学習係数を 0.0001，割引率を 0.99 とする．学習終了条件は global step 数が 1.0×10^8 に到達した場合とする．また，エピソードの終了条件は各ゲームにおける 1 プレイ終了，及び step 数が 1.0×10^4 に到達した場合とする．

4.2 スコア比較

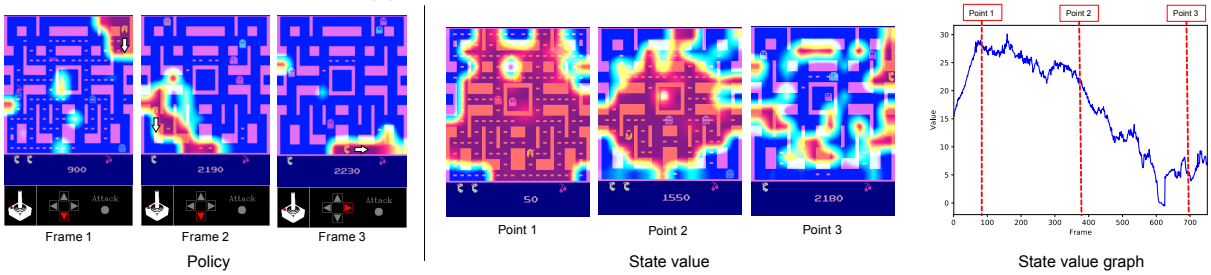
各ゲームタスクにおける 100 エピソード間の最大/平均スコアを表 1 に示す．表 1 から，Breakout における最大スコアは全手法において 864 である．このスコアは，Breakout で獲得できる最高スコアである．また，Breakout における平均スコアは，Mask A3C が A3C と比較しほぼ同等である．Ms.Pac-Man では，最大/平均スコア共に Mask A3C が最も高いスコアである．Space Invaders では，最大スコアは Policy Mask A3C，平均スコアは Mask A3C が最も高いスコアである．Breakout はパドルでボールを打ち返すのみであり，外的要因のない単純なタスクである．そのため，A3C と Mask A3C が同等のスコアであったと考えられる．一方，Ms.Pac-Man と Space Invaders は，敵などの外的要因を考慮して行動を選択する必要がある．Policy branch に Attention 機構を導入した Policy Mask A3C と Mask A3C では，クッキーやインベーダーなど，方策に関連した領域を強調する．そのため，A3C と比較しスコアが向上したと考えられる．

4.3 Mask-attention を用いた視覚的説明

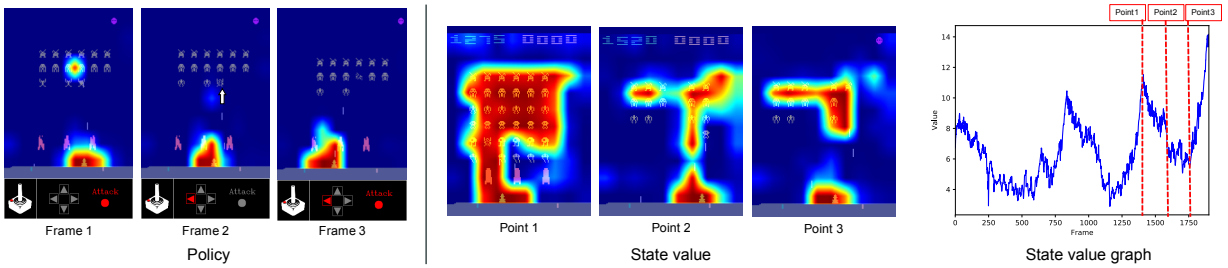
各ゲームにおける Mask-attention の可視化例を図 2 に示す．図 2(a) の Policy から，Frame 1 ではボールの進行方向を注視している．Frame 2 ではパドルを含めたボール周囲を注視し，ボールを打ち返した後である Frame 3 では注視している領域がない．ここから，Breakout ではボールを打ち返す直前において，ボールの進行方向を予測しパドルを制御していると考えられる．図 2(b) の Policy から，Frame 1 ではパックマン周囲を注視している．Frame 2 では画面内に存在するクッキーを注視し，Frame 3 では Frame 2 における注視領域にパックマンが移動している．ここから，Ms.Pac-Man ではパックマンの周囲を警戒しながら，クッキーへ向かうように制御していると考えられる．図 2(c) の Policy から，Frame 1 でインベーダーを注視し，行動は攻撃を選択している．そして，Frame 3



(a) Breakout: 白矢印はボールの進行方向を示す.



(b) Ms.Pac-Man: 白矢印はパックマンの進行方向を示す.



(c) Space Invaders: 白矢印はビームの進行方向を示す.

図 2: Mask-attention の可視化例: Policy における下部のコントローラは，現フレームでモデルが推論した行動である.

表 1: 各ゲームおける 100 エピソード間の最大/平均スコア: 各手法で 5 試行ずつ学習し，平均スコアが最も高いモデルのスコアを示す．max/mean は最大/平均スコアである.

	Att. mechanism		Breakout		Ms.Pac-Man		Space Invaders	
	Policy	Value	max	mean	max	mean	max	mean
A3C			864	**662.0**	5380	4573.3	19505	18531.8
提案手法	✓		864	595.8	6330	4833.8	19860	19102.8
		✓	864	606.9	4830	4044.5	19675	18537.8
	✓	✓	864	640.0	6610	**5314.1**	19810	**19212.5**

表 2: Mask-attention の反転によるスコア比較: normal は Mask-attention を反転しない場合，inverse は反転した場合，random はランダムに行動選択した場合である.

Att. mechanism			Breakout		Ms.Pac-Man		Space Invaders	
Policy	Value		max	mean	max	mean	max	mean
✓		normal	864	595.8	6630	4833.8	19860	19102.8
		inverse	4	2.2	290	268.9	805	306.9
✓	✓	normal	864	640.0	6610	5314.1	19810	19212.5
		inverse	5	1.8	410	194.4	915	420.2
random			5	1.2	1080	247.8	460	142.1

において Frame 1 で注視したインベーダーを撃破している．ここから，Space Invaders では撃破するインベーダーを認識し，エージェントを制御していると考えられる．また，図 2(a)(b)(c) の State value から，Breakout ではブロック，Ms.Pac-Man ではクッキー，Space Invaders ではインベーダーを注視している．そして，注視した物体の減少に合わせ注視領域が縮小している．これらの結果から，Policy の Mask-attention は現状態に対し行動に直結する物体，State value の Mask-attention はスコアに寄与する物体を表していると考えられる.

4.4 Mask-attention の有効性

Mask A3C による行動の視覚的説明を行うにあたり，Mask-attention がモデルの出力である方策に対して有益な領域を表しているか検証する．検証方法として，Mask A3C における Policy branch の Mask-attention を反転したマップを作成し，そのマップを Attention 機構に用いた場合のスコアを算出する．Mask-attention を反転する場合と反転しない場合におけるスコアを比較することで，Mask-attention が行動の視覚的説明に有効であることを確認する．表 2 に，Mask-attention の反転によるスコア比較を示す．表 2 から，全ゲームにおいて inverse が normal と比較して著しくスコアが低下し，random と同等のスコアである．したがって，Policy branch における Mask-attention は，高スコアを獲得する行動に対して有益な領域を獲得したと言える.

5. おわりに

本研究では，深層強化学習の代表的な手法である A3C に Attention 機構を導入した Mask A3C を提案した．Mask A3C では，推論時に Mask-attention を可視化することで，エージェントの行動選択に対する判断根拠の視覚的説明を実現した．今後は，ロボット制御などの更に複雑なタスクへの適用と可視化に取り組む予定である.

参考文献

[1] V. Minh, *et al.*, “Asynchronous Methods for Deep Reinforcement Learning”, ICML, 2016.

[2] X. Shi, *et al.*, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting”, NeurIPS, 2015.

[3] G. Brockman, *et al.*, “OpenAI Gym”, *arXiv preprint arXiv:1606.01540*, 2016.

研究業績

[1] 板谷英典 等, “A3C における Attention 機構を用いた視覚的説明”, 人工知能学会全国大会, 2020. (学生奨励賞受賞)
(他 2 件)

セマンティックセグメンテーションによる超高解像度画像からの変化点検出に関する研究

TP19015 筒井駿吾　　指導教授：山下隆義

1. はじめに

変化点検出は，異なる時刻に撮影された同一シーン画像から変化した領域を抽出する問題である．CNN を用いた変化点検出法 [1][2] は，位置ずれや環境変化に頑健であるため，マップの自動更新や土地利用調査への利用が期待されている．土地利用調査に応用する場合，変化した領域の検出だけでなく，農地や森林など過去と現在の土地の属性情報が必要である．これにより，変化した場所の内訳を把握することができる．

そこで本研究では，変化領域とオブジェクトクラスを同時に推定し，変化領域に土地の属性情報を付与することができる変化点検出法を提案する．提案手法は，セマンティックセグメンテーションと変化点検出を 1 つのネットワークで実現する．また，セグメンテーションモデルより抽出した特徴マップを，変化点検出モデルの入力として使用することで，両タスクの精度向上を図る．

2. 変化点検出の従来法

CNN を用いた変化点検出法は，特徴量ベースと学習ベースの 2 つのアプローチに分類できる．

特徴量ベースの変化点検出法：CNN より抽出した 2 枚の特徴マップの距離計算から変化領域を推定する．そのため，変化点検出のための学習データセットを必要としない．特徴量ベースの代表的な手法である CNNF-F[1] は，ImageNet で事前学習した VGG19 を用いて，画像ペアから特徴マップ抽出し変化領域を検出する．検出にはユークリッド距離を使用し，変化領域を決定するために閾値処理を施す．変化領域のみを検出する手法であるため，変化領域に属性を付与することができない．

学習ベースの変化点検出法：変化点検出のためのデータセットを用いて画像間の類似性 (非類似性) を学習するため，前述のアプローチと比べて視点のずれに頑健な検出が可能である．学習ベースの代表的な手法である FC-EF-Res[2] は，変化点検出タスクとセマンティックセグメンテーションタスクをそれぞれ別々のネットワークで処理し，その結果を統合する手法である．2 つのネットワークは U-Net[3] をベースに構成され，残差ブロックを導入している．セグメンテーションネットワークから抽出した特徴マップの差分を，変化点検出ネットワークにスキップ接続することで，画像間の明示的な比較を行いながら学習する．しかし，タスクごとに独立したネットワークを用いるため，学習中にタスク間で情報を相互利用することができないという問題がある．

3. 提案手法

本研究では，変化領域とオブジェクトクラスを同時に推定することができる変化点検出法を提案する．提案手法のネットワーク構造を図 1 に示す．ネットワークは，Encoder-Decoder 構造を持つ 2 つのモデルから構成されており，変化点検出モデルにはセグメンテーションモデルの最終層から抽出した特徴マップを入力する．

3.1 セグメンテーションモデル

セグメンテーションモデルは，U-Net をベースとするネットワーク構造である．画像ペアのそれぞれを個別にネットワークへ入力する．なお，各ネットワークの重みは共有している．セグメンテーションモデルの損失関数は，従来の U-Net と同様，クロスエントロピー損失を使用する．

3.2 変化点検出モデル

変化点検出モデルは，2 つの Encoder と 1 つの Decoder で構成したネットワーク構造となっており，それぞれ 3 層の畳み込み層を持つ．2 つの Encoder により，画像間で異なる特徴を獲得することが可能である．また，各 Encoder にはセグメンテーションモデルより得られた特徴マップをそれぞれ入力し，Decoder には各 Encoder より得られた特徴マップをチャンネル方向に連結し入力する．

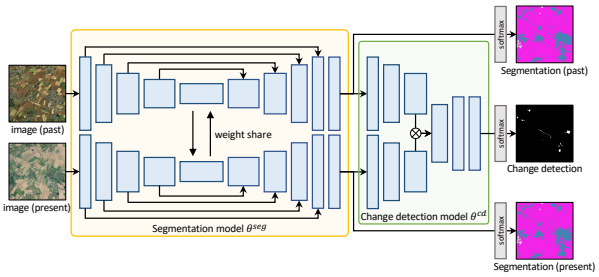


図 1：変化点検出ネットワークの概要

3.3 段階的な学習と損失関数

提案手法は，セグメンテーションモデルと変化点検出モデルにより構成され，段階的な学習を行う．

Step1：まず，ベースとなるセグメンテーションモデル θ^{seg} を学習する．ピクセル総数を N，クラス数を S, 正解ラベルを t，ソフトマックス関数の出力を y とすると，セマンティックセグメンテーションタスクの損失関数 L_{Seg} は式 (1) で表すことができる．

$$L_{Seg} = -\frac{1}{N}\sum_{i}^{N}\sum_{c}^{S} w_c t_{i,c} \log y_{i,c} \tag{1}$$

ここで，w_c は各タスクの重みである．衛星画像などの超高解像度画像で構成されるデータセットを使用する場合，クラス不均衡が生じる．ラベルが不均衡のまま学習すると，出現確率の高いクラスに対して早期に収束してしまい，出現確率の低いクラスがほとんど反映されない問題がある．そのため，式 (2) に示すように，出現確率 p_c の逆数に対数を重みとする．

$$w_c = \log \frac{1}{p_c} \tag{2}$$

これにより，出現確率の高いクラスにオーバーフィッティングすることを防ぐ．

Step2：次に，セグメンテーションモデル θ^{seg} の重みを固定し，変化点検出モデル θ^{cd} を学習する．式 (2) で定義した重みを使用し，クラス数を D とすると，変化点検出タスクの損失関数 L_{Diff} は式 (3) で表すことができる．

$$L_{Diff} = -\frac{1}{N}\sum_{i}^{N}\sum_{c}^{D} w_c t_{i,c} \log y_{i,c} \tag{3}$$

Step3：そして，2 つのモデルの相互関係を保つため，θ^{seg} と θ^{cd} を同時に学習する．相互関係を保つための損失関数 L_{Mut} を式 (4) に示す．Step2 と同様，L_{Mut} の算出には θ^{cd} より抽出した特徴マップを使用する．

$$L_{Mut} = -\frac{1}{N}\sum_{i}^{N}\sum_{c}^{D} t_{i,c} \log y_{i,c} \tag{4}$$

Step4：各 iteration で Step1 から Step3 を繰り返す．

以上に示す段階的な学習により，タスク間で情報を相互利用し，両タスクの精度向上が期待できる．

3.4 学習データの入力方法

提案手法は，10,000×10,000 ピクセルの超高解像度画像を対象とし，学習時は 512×512 ピクセルにランダムクロップする．しかし，クラス不均衡が生じる場合，ランダムクロップでは出現確率の低いクラスが選択されず，学習が不安定となる．そこで，出現確率の低いクラスを多く含むパッチ画像を別で用意し，通常のパッチ画像とともに学習を行う．これにより，クラス不均衡による影響を抑えることができるため，学習の安定性につながる．

4. 評価実験

提案手法の有効性を示すために，変化点検出及びセマンティックセグメンテーションの評価実験を行う．

表 2：変化点検出とセマンティックセグメンテーションの比較結果

		Semantic segmentation				Change detection		
		Global Accuracy	Class Accuracy	mean IoU	Kappa	Global Accuracy	mean IoU	Kappa
U-Net	Seg	90.96	62.78	45.83	78.56	92.36	48.28	6.86
	Diff	-	-	-	-	**99.25**	54.56	17.72
CNNF-F		-	-	-	-	88.66	-	3.28
FC-EF-Res		89.01	-	-	71.92	98.30	-	25.49
Proposed		**91.46**	**63.21**	**53.48**	**79.24**	99.17	**59.32**	**32.20**

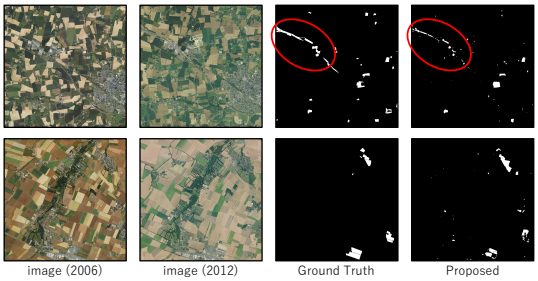


図 2：提案手法による変化点検出例

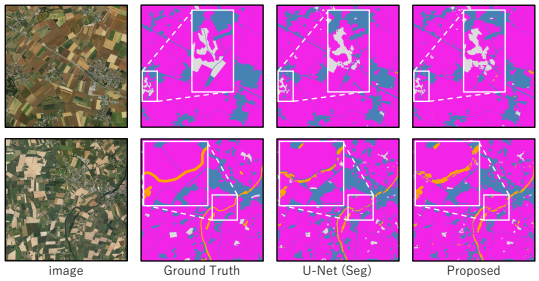


図 3：各手法のセグメンテーション例

4.1 データセット

本実験では，異なる時期に撮影された衛星画像ペアで構成される HRSCD データセット [2] を使用する．HRSCD データセットは，大規模な超高解像度の変化点検出用データセットである．RGB 画像ペアと，各ピクセルでの変化情報を付与した変化検出ラベル (変化領域なし，変化領域あり)，土地利用情報を付与したセグメンテーションラベル (情報なし，人工物，農業地帯，森林，湿地，水域) が含まれる．データ総数は 582 枚あり，学習用に 292 枚 (146 組の画像ペア)，評価用に 290 枚 (145 組の画像ペア) を使用する．表 1 に HRSCD データセットの各クラスにおける出現確率と重みを示す．表 1 より，出現確率が小さいほど，クラス重みが大きくなることがわかる．

表 1：各クラスにおける出現確率と重み

	Semantic Segmentation						Change Detection	
	情報なし	人工物	農業地帯	森林	湿地	水域	変化なし	変化あり
出現確率 [%]	17.70	11.51	61.84	8.36	0.02	0.58	99.23	0.77
クラス重み	0.0	2.16	0.48	2.48	8.54	5.15	0.01	4.87

4.2 実験概要

HRSCD データセットは，学習サンプルが少ないため，幾何変換，ノイズ付加等の Data Augmentation を行い，バリエーションを増加する．評価時は，超高解像度画像を 512×512 ピクセルの画像としてタイル状に切り出して入力する．定量的評価指標として，変化点検出，セマンティックセグメンテーション共に Global Accuracy，mean IoU，Kappa 係数を用いる．セマンティックセグメンテーションでは，さらに Class Accuracy を用いる．また，提案手法の性能比較として，U-Net および CNNF-F，FC-EF-Res を使用する．U-Net は，単一のネットワークとして変化点検出タスク U-Net (Diff) とセマンティックセグメンテーションタスク U-Net (Seg) それぞれの学習を行う．

4.3 各タスクにおける評価結果

各手法を使用した時の，変化点検出とセマンティックセグメンテーションの定量的評価を表 2 に，定性的評価を図 2 および図 3 に示す．U-Net (Seg) では，各画像に対するセグメンテーション結果を利用した変化点検出を行っており，画像間でクラスが異なる領域を変化領域とする．

変化点検出：表 2 より，提案手法が従来手法と比較し，mean IoU と Kappa 係数において高精度であることが確認できる．一方，U-Net (Seg) ではセグメンテーション精度は高いものの，検出精度は低下した．これは，セグメンテーション結果においてクラス境界の誤識別が多くあるため，変化領域の誤検出を招いたと考えられる．また，図 2 より提案手法は道路などの細い領域の検出可能であることが確認できる．

セマンティックセグメンテーション：表 2 より，提案手法は全ての評価指標で最も高精度であることが確認できる．特に U-Net (Seg) は，提案手法のセグメンテーションモデルと同一のネットワーク構造であるが，識別精度が最大約 0.7 ポイント向上した．これは，タスク間で情報を相互利用したからであると考えられる．また，図 3 より農業地帯などの領域の広いクラスと，検出が困難な川など領域が細いクラスにおいても高精度なセグメンテーションができていることが分かる．

4.4 クラス不均衡に対する有効性の調査

人工物クラスを含むパッチ画像を追加することによる精度比較を行う．表 3 に，追加前と追加後それぞれの定量的評価を示す．出現確率の低いクラスを含むパッチ画像を追加することで学習が安定し，ほぼ全てのクラスで精度向上を確認した．また，出現確率の高い農業地帯の誤識別が減少したことで，森林クラスの識別精度が約 35.7 ポイント向上した．

表 3：入力方法変更による比較結果

	Semantic Segmentation					Change Detection	
	人工物	農業地帯	森林	湿地	水域	変化なし	変化あり
追加前	67.51	85.27	37.81	0.0	31.56	**99.20**	17.74
追加後	**68.97**	**89.88**	**73.55**	0.0	**35.01**	99.17	**19.47**

5. おわりに

本研究では，変化領域のオブジェクトクラスを推定することのできる変化点検出法を提案した．2 つのモデルを使用することで，高精度な変化領域の検出を実現し，単一の U-Net 同様のセグメンテーション精度を達成した．今後は，出現確率の低いクラスの更なる精度向上を検討する．

参考文献

[1] E. Amin, *et al.*, “Convolutional neural network features based change detection in satellite images”, First International Workshop on Pattern Recognition, 2016.

[2] R. C. Daudt, *et al.*, “Multitask Learning for Large-scale Semantic Change Detection”, CVIU, 2019.

[3] O. Ronneberger, *et al.*, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, MICCAI, 2015.

研究業績

[1] 筒井駿吾 等,“セマンティックセグメンテーションによる超高解像度画像からの変化点検出”, 動的画像処理実利用化ワークショップ, 2021.

(他 1 件)

異なる解像度間の整合性を考慮した顕著性予測モデルに関する研究

TP19010 瀬尾俊貴　　指導教授：山下隆義

1. はじめに

顕著性予測は，人が興味・関心を持つと考えられる領域をヒートマップにより表現する．顕著性予測のアプローチとして，周囲とは異なる色彩やエッジ特徴を抽出して顕著領域を推定する手法がある．また，Convolutional Neural Network (CNN) の発展により，人の視線から得られた顕著性マップを学習・推論に用いる高精度な顕著性予測手法も提案されている．一方で，顕著性予測の応用先として自動運転システムを想定すると，予測精度だけではなくメモリ消費量や計算時間の短縮が重要となる．そこで，本研究ではメモリ消費量を効率化した顕著性予測モデルを提案する．さらに，パラメータの少ないモデルで高精度に推定するために，画像解像度毎の顕著性の一貫性を考慮した学習手法も提案する．

2. 顕著性予測

顕著性予測の代表的な手法として，RARE2012 [1] が挙げられる．RARE2012 では，主成分分析によりカラーチャンネル毎に白色化を行う．その後，得られた画像に対してマルチスケールのガボールフィルタで特徴量を抽出する．最後に，得られた特徴量の確率密度を計算し，自己情報量に基づいて顕著性予測を行う．ただし，RARE2012 では危険予知など人の事前知識にもとづく顕著性の違いを考慮できないという問題がある．また，人の視線情報を利用した CNN による高精度な顕著性推定手法がある．SalNet [2] では，畳み込み層 3 層，全結合層 2 層から構成されるネットワークを用いて顕著性マップを出力する．ネットワークの学習には人の視線データをもとにしたマップを教師として用いる，また，EML-Net [3] では，異なるデータセットで事前学習した複数ネットワークの視覚特徴を統合することで高精度な予測が可能である．しかし，EML-Net は，大規模なネットワークを利用して特徴抽出を行うため，モデルのパラメータ数が膨大となり，計算時間がかかる問題がある．

3. 提案手法

本研究では，メモリ消費量及び計算時間の短縮と予測精度の向上を両立する顕著性予測手法を提案する．まず，メモリ消費量と計算時間の短縮のために，MobileNetV2 と Mixed Depthwise Convolution を利用してネットワークを構築する．これにより，パラメータ数を削減できる．次に，予測精度を向上させるために，人の視覚的特性に基づいた解像度毎の視覚的特徴を効率よく学習する損失関数を提案する．

3.1 ネットワーク構造

提案手法で用いるネットワーク構造を図 1 に示す．はじめに，MobileNetV2 を特徴抽出器として画像から特徴を得る．この時，MobileNetV2 は IRL，Batch Normalization (BN)，ReLU6 を組み合わせた計 7 層の構成であり，前半の 3 層と後半の 4 層から特徴を得る．

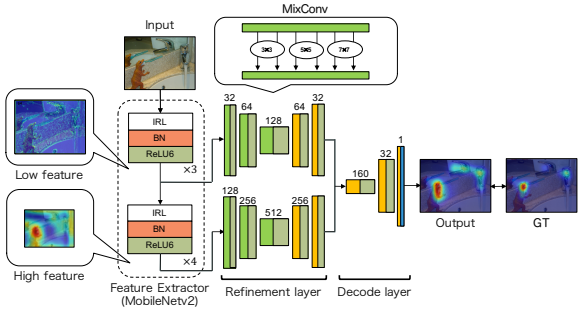


図 1：提案手法のネットワーク構造

前半の特徴は，浅い層から得られる特有の特徴である周囲とは異なる色情報や識別に有用なエッジなどの単純な視覚情報となる．また，後半の特徴は，深い層から得られる特有の特徴である物体などの大域的な視覚情報となる．次に，得られた特徴を Refinement layer に入力する．Refinement layer は得られた特徴を明瞭化するモジュールとなっており，入出力の解像度を変更しない Encoder-Decoder 構造から構成されている．さらに，Refinement layer には様々な受容野を 1 度の畳み込み層に集約を行った Mixed-depthwise Convolution (MixConv) を用いることで，演算量を抑えつつもロバストな特徴の抽出を行う．最後に，Refinement layer から得られた 2 つの特徴を統合し，Deconvolution 層を用いて顕著性マップの推論を行う．

3.2 損失関数

画像解像度と顕著性マップの関係性について，人の視線を収集し調査した研究がある [4]．これによると，解像度を 32 分の 1 までリサイズした画像の顕著性マップは元の解像度の顕著性マップと非常に相関が高いことが確認されている．この知見を活かし，元の解像度から予測される顕著性マップと低解像度化した画像から予測される顕著性マップの整合性を損失として求める．これを定式化すると式 (1) のように定義できる．

$$L_{res} = BCE(S^{GT}, S^{P_b}) + \sum_{i=1}^{N} BCE(S^{P_b}, S^{P_{2^i}}) \quad (1)$$

第一項は，バイナリクロスエントロピーを用いた真値と元の解像度から予測される顕著性マップの損失である．第二項は，元の解像度から予測される顕著性マップと低解像度化した画像から予測される顕著性マップからの損失である．ここで，BCE はバイナリクロスエントロピー，S^{GT} は顕著性マップの真値，S^{P_b} は元の解像度から予測された顕著性マップ，$S^{P_{2^i}}$ は解像度を 2^i 分の 1 にした時の推論結果，N はダウンサンプルする回数である．図 2 に示すように，低解像度の画像は元の解像度の画像を 2^i 分の 1 にダウンサンプリング後，バイリニア補間により元の解像度へアップサンプリングした画像とし，モデルに入力する．

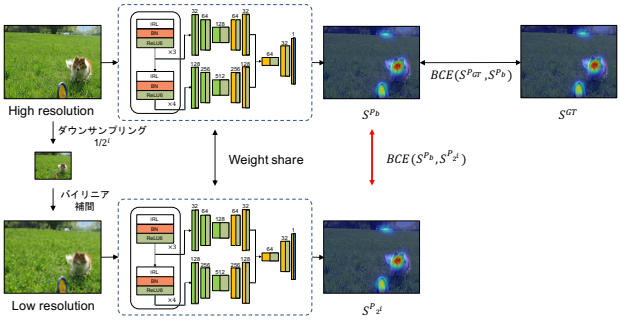


図 2：各解像度間の整合性を考慮した一貫性損失

4. 評価実験

提案手法の有効性を示すために，静止画における人の視線データから得られた顕著性を収集したデータセットを用いて，評価実験を行う．

4.1 実験概要

評価データには，SALICON 及び CAT2000 を用いる．SALICON は，様々な自然画像から構成される MS-COCO データセットから 20,000 枚を抜粋し，約 60 名の被験者から顕著性を獲得して付与したものである．また，CAT2000 は 120 名の被験者を対象に，漫画，アート，オブジェクト，低解像度画像，屋内，屋外，ランダムな画像，線画といった異なるタイプのシーンをカバーし，20 のカテゴリから構成されている．画像の入力サイズは 640 × 480 で

表 1：SALICON，及び CAT2000 データセットによる定量的評価結果

	SALICON						CAT2000					
	Params	SIM ↑	CC ↑	AUC ↑	NSS ↑	KL ↓	Params	SIM ↑	CC ↑	AUC ↑	NSS ↑	KL ↓
EML-Net[1]	47.08M	**0.765**	**0.878**	**0.864**	**1.987**	0.520	47.08M	**0.782**	**0.885**	**0.866**	**2.060**	**0.298**
Vanilla	**4.72M**	0.686	0.779	0.844	1.577	0.393	**4.72M**	0.740	0.833	0.858	1.731	0.321
Vanilla+R	5.75M	0.714	0.811	0.850	1.673	0.338	5.75M	0.757	0.842	0.860	1.799	0.311
Vanilla+R+L_{res}	5.75M	0.742	0.847	0.861	1.762	**0.282**	5.75M	0.776	0.851	0.866	1.875	0.299

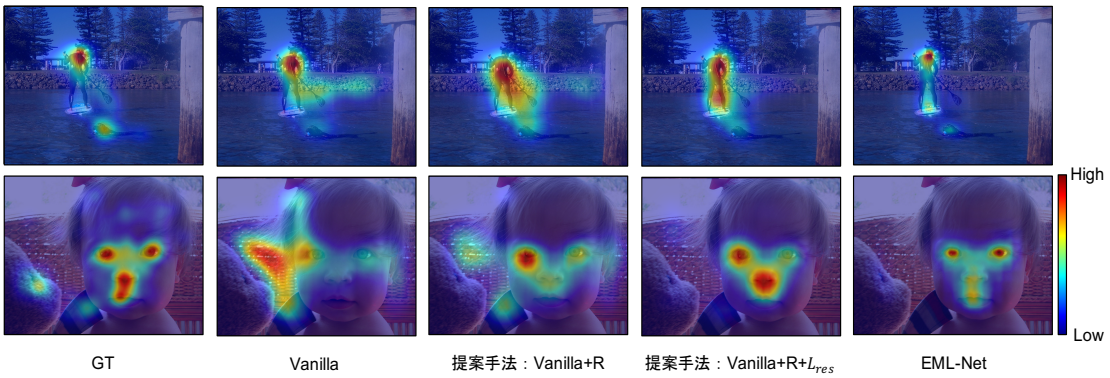


図 3：各手法による顕著性マップの推定例

ある．SAICON は学習用に 10,000 枚，評価用に 5,000 枚を，CAT2000 は学習用に 1,600 枚，評価用に 400 枚を用いる．また，学習モデルの汎化性能の観点から前処理として入力画像を [0,1] へ正規化，及び左右反転のデータ拡張を行っている．評価に利用する従来手法とモデルを以下のように定義する．

EML − Net：EML-Net は，複数の事前学習済みモデルを活用した学習を行うことで，事前にエンコードされた識別に有効な特徴を効果的に利用する手法である．

Vanilla：MobileNetV2 のデコード部を全結合層から Deconvoliution 層を 3 層に変更することで，顕著性予測用にネットワークを変更したモデルである．

Vanilla + R(Ours)：Refinement layer 及び MixConv を活用して獲得した特徴を統合して Deconvolution 層へ入力するモデルである．

Vanilla + R + $\mathbf{L_{res}}$(Ours)：Vanilla+R のモデルに対して，各解像度の整合性を考慮した損失関数 L_{res} を導入したモデルである．

4.2 ベースラインとの定量的・定性的な比較

顕著性予測における定量的評価指標として，SIM(Similarity)，CC(Correlation coefficient)，AUC(Area Under Curve)，NSS(Normalized Scanpath Saliency)，KL(KL divergence) がある．これらの評価指標を用いて，SALICON データセットと CAT2000 データセットにおける各精度の比較結果とパラメータ数を表 1 に示す．提案手法は，整合性を保つ損失関数 L_{res} と Refinement layer により，従来手法と比較してほぼ同等の精度であることがわかる．また，IRL と MixConv を利用することにより，パラメータ数を 87.7%削減できていることが確認できる．

次に，図 3 に各手法による顕著性マップの推定例を示す．図 3 より，Vanilla では岩や木材で編まれた椅子などエッジ等の特徴が密集する位置に誤って顕著性が現れている．一方，提案手法は Refinement layer と L_{res} の導入により誤りの少ない顕著性マップとなっている．

4.3 損失関数の有効性の調査

図 4 に，異なる解像度間の画像を入力としたときの CC による精度変化を示す．なお，元の解像度を横軸の 1 とし，右へ移るにつれて解像度が 2 分の 1 ずつ低下している．図 4 から，一貫性損失を用いていない Vanilla+R (Without L_{res}) と比べ，Vanilla+R+L_{res} (With L_{res}) では低解像度でも精度は低下しない事が確認できる．人間の視線では，32 分の 1 までの解像度であれば一定の顕著性マップが得られることから，各解像度間の整合性を考慮した一貫性損失を利用することで，人間に近い視覚的特徴を学習できているといえる．

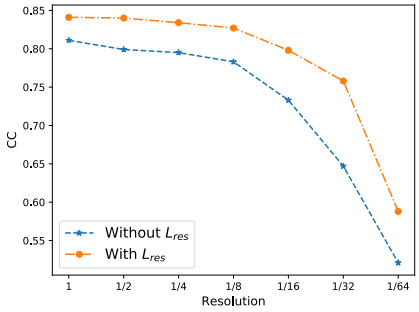


図 4：複数解像度に対する提案手法の精度比較

5. おわりに

本研究では，顕著性予測において MobileNetV2 や MixConv を利用したネットワークの設計によりパラメータ数を約 87.7%削減した．また，各解像度間の顕著性の整合性を保つ損失関数 L_{res} の提案により，L_{res} を利用しない場合と比べ精度が向上し，従来手法と比べてパラメータ数を削減したにも関わらず，ほぼ同等の精度となることを確認した．今後は，更なる精度向上を狙うためにアンサンブルモデルや AutoML によるネットワーク構造の最適化を行う予定である．

参考文献

[1] N. Riche, *et al.*, “A multi-scale rarity-based saliency detection with its comparative statistical analysis”, SPIC, 2013.

[2] J. Pan, *et al.*, “Shallow and Deep Convolutional Networks for Saliency Prediction”, CVPR, 2016.

[3] S. Jia, *et al.*, “EML-NET:An Expandable Multi-Layer NETwork for Saliency Prediction”, IVC, 2020.

[4] T. Judd, *et al.*, “Fixations on low-resolution images.”, JOV, 2011.

研究業績

[1] 瀬尾俊貴 等, “FlowNetC を導入した D&T における物体検出の高精度化”, 画像センシングシンポジウム, 2019.

[2] T. Seo, *et al*, “Video Object Detection and Tracking based on Angle Consistency between Motion and Flow”, IV, 2020.

(他 学会発表 1 件)

視線情報を活用した一貫学習ベースの自動運転制御に関する研究

TP19023 森啓介　　指導教授：山下隆義

1. はじめに

自動運転制御は，周辺環境の認識，経路計画，車両制御を順次行うアプローチが用いられる．一方，Convolutional Neural Network (CNN) の発展により，新たな自動運転制御として，車載カメラ画像から制御値を直接推定する一貫学習ベースの手法が提案されている [1]．本手法は，人間が実際に運転した際の情報を教師データとして模倣学習を行う．入力には主に画像が用いられ，画像には背景などの運転制御に無関係な領域が多く含まれることがある．この時，CNN がそれらの不要な領域を注視すると，正しい制御値を推定できないことが考えられる．人間の視覚システムは，視線移動によって重要となる情報を獲得して，視覚的に認識を行うことが知られている．そこで本研究では，視線情報を運転制御モデルに利用した一貫学習による自動運転を実現する．

2. 一貫学習ベースの自動運転制御

代表的な一貫学習ベースの自動運転手法として，Bojarski らの研究 [1] がある．この研究では，5 層の畳み込み層と 4 層の全結合層の CNN に画像を入力し，ステアリング値を推定する．また，Conditional Imitation Learning RS (CILRS) [2] は，ResNet を特徴抽出部に導入し，マルチタスクとして車両速度推定を追加することにより，運転性能を改善している．CILRS では，全結合層で構成した出力部を直進や右左折といったコマンドの数だけ用意している．コマンドに応じて出力部を切り替えることで，一貫学習ベースの自動運転において困難であった，交差点での高精度な制御を実現している．しかし，運転制御に関係ない領域に対する特徴を抽出することで，制御値の正しい推論に失敗するという問題が考えられる．

3. 提案手法

本研究では，人の視線情報を利用することで，運転に必要な領域を重視できる運転制御モデルを提案する．提案するモデルのネットワークを図 1 に示す．

3.1 視線推定モデル

自動運転時に，人が注視する位置を常に計測するのは人への負担が大きい．そこで，人の注視領域を表す視線マップを推定する視線推定モデルを実現し，推定した視線情報を運転制御モデルに利用する．これにより，人の視線情報を用いることなく，運転制御モデルの学習および評価が可能となる．視線推定モデルのネットワークはエンコーダおよびデコーダで構成する．エンコーダは VGG-16 の畳み込み層を，デコーダは 4 層の deconvolution 層を用いる．交差点の場面では，進行方向によって注視する箇所が大きく異なる．そのため，運転指示であるコマンド c をネットワークに追加することで，運転指示に合わせた視線推定を学習する．コマンドは，エンコーダから獲得された特徴マップと同サイズにリサイズし，チャンネル方向に結合して追加する．視線推定モデルの学習には，式 (1) に示す Binary Cross Entropy (BCE) Loss を用いる．ここで，G，$\hat{G}$ はそれぞれ視線マップの真値と予測値である．

$$L_g = \frac{1}{WH}\sum_{i}^{W}\sum_{j}^{H} -G_{i,j}\log\hat{G}_{i,j} - (1-G_{i,j})\log(1-\hat{G}_{i,j}) \tag{1}$$

3.2 運転制御モデル

運転制御モデルには CILRS をベースに用いる．視線情報を考慮するため，特徴抽出部を構成する Res. Block 毎の特徴マップ $F_i(x)$ に視線マップ G_i を画素毎に乗算する．$\hat{G}_i$ は $\hat{G}$ を i 番目の Res. Block における特徴マップの解像度にダウンサンプリングしたものである．そして，注視領域以外の特徴量が消失するのを防ぐため，式 (2) に示すように乗算前の特徴マップを加算する．

$$F_i'(x) = F_i(x) \otimes \hat{G}_i + F_i(x) \tag{2}$$

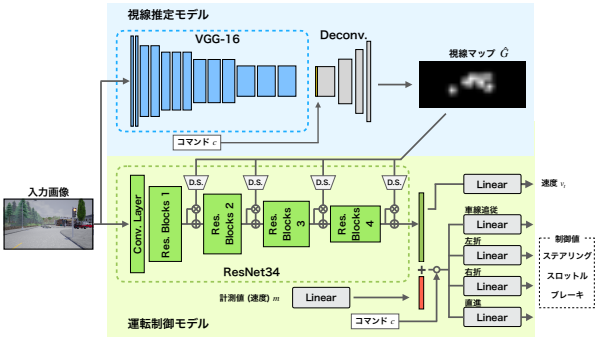


図 1：提案手法のネットワーク構造

特徴抽出部では，下位層でエッジなどの詳細な特徴を抽出し，上位層でより大局的な空間特徴を獲得する．本手法では，視線情報を深さの異なる 4 箇所に追加する．これにより，スケールの異なる特徴ごとに重要な領域を強調できる．

運転制御モデルの出力部はコマンドごとに用意する．コマンド c に対応した出力部を選択して，制御値を出力する．学習時には，運転制御モデルの各出力に対して L1 Loss を損失関数に用いる．ここで，運転制御モデルの出力は，車両速度とステアリング，スロットル，ブレーキの各値である．

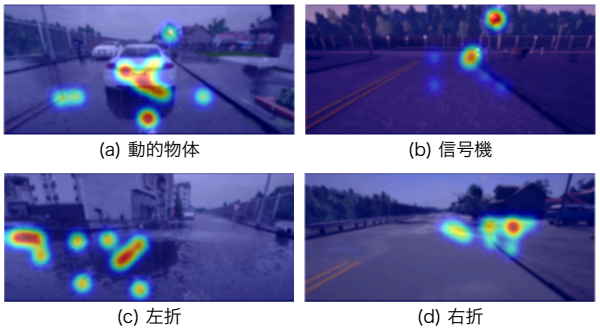
4. 評価実験

提案手法の有効性を調査するため，視線情報の有無による運転精度の比較を行う．

4.1 データセット

本実験では，自動運転シミュレータである CARLA[3] にて作成された CARLA100 データセットを運転制御モデルの学習に用いる．CARLA100 データセットは，車両の位置や信号機の状態などの内部情報から自動運転を行うエキスパートエージェントを用いて，100 時間におよぶ運転データをシミュレータ上で自動的に収集したデータセットである．

視線推定モデルの学習には，CARLA100 データからランダムで選択した 2,000 枚のデータに対して，視線情報を付与したデータセットを用いる．1 フレーム毎に十分な視線情報が含まれるようにするため，静止画に対する視線情報を記録する．視線情報の取得には Tobii Pro X3 130 を使用し，1 枚の静止画をディスプレイに 5 秒間表示することで視線データを収集した．本データセットの視線データの例を図 2 に示す．



(a) 動的物体　(b) 信号機

(c) 左折　(d) 右折

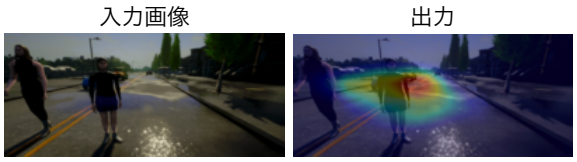
図 2：記録した視線情報の例

4.2 実験概要

視線推定モデルの入力は 400×176 にリサイズした RGB 画像を用いる．最適化手法には MomentumSGD を用い，学習率を 0.03 とする．

運転制御モデルは，CARLA100 データセットから 10 時間のサブセットを学習に用いる．ここで，入力は 200×88 の RGB 画像を用いる．最適化手法には Adam を使用し，学習率を 0.0002 とする．

学習は 2 段階に分けて行う．はじめに視線推定モデルを学習する．そして，視線推定モデルの重みを固定して運転



コマンド：**車線追従**，steer：**0.02**，throttle：**0.00**

(a) 歩行者

コマンド：**右折**，steer：**-0.27**，throttle：**0.00**

(b) 信号機

コマンド：**車線追従**，steer：**-0.02**，throttle：**0.02**

(c) カーブ

コマンド：**右折**，steer：**0.01**，throttle：**0.61**

(d) 交差点

図 3：**視線マップ推定例**：steer は負数で左，正数で右のステアリング操作を表す．throttle は，正数でアクセル操作を表す．

制御モデルを学習する．

評価実験では，シミュレータ環境として CARLA を用いた走行実験を行う．評価指標には，NoCrash Benchmark [2] を用いる．NoCrash Benchmark では，CARLA 上で異なるタスクを実行し，自動運転の性能を評価する．走行するマップは，学習環境である Town01 と未知環境の Town02 である．タスクは，動的物体の数により Empty town，Regular traffic，Dense traffic の 3 種類に分けられ，異なる環境や交通状況における運転性能を評価することが可能である．事故発生時または制限時間を超えた場合，エピソードが失敗となり，目的地へ到達することでエピソード成功となる．

4.3 視線推定の定性的評価

視線推定結果の例を図 3 に示す．図 3(a) の歩行者が横断するシーンでは，衝突の可能性が高い歩行者に視線マップが反応し，減速を行っている．図 3(b) では，前方の赤信号に反応し，運転に必要な物体を注視できている．図 3(c) のようなカーブにおいて，カーブの行き先に反応しており，図 3(d) の交差点においては，交差点の右方向に反応している．この結果から，運転指示に合わせた進行方向を注視できていることがわかる．

4.4 運転性能評価

表 1 に走行実験の成功率を示す．表 1 より，ほとんどのタスク及び環境条件において CILRS より成功率が高いことが確認できる．これにより，視線情報が運転精度向上に貢献していることがわかる．また，精度向上は，どのタスクにおいても確認でき，静的物体と動的物体のどちらにおいても視線情報が有効であることがわかる．

同一の実験におけるエピソード失敗原因のうち，動的物体との衝突が原因のエピソード失敗率を表 2 に示す．ここで，Col. Ped. は歩行者との衝突による失敗，Col. Veh. は車との衝突による失敗である．表 2 より，提案手法はすべての条件及びタスクにおいて自動車との事故率が低いことが確認できる．成功率が低下していた未知天候を含む条件においても事故率が改善されており，CILRS に比べてより自動車に反応できていることがわかる．一方，歩行者との事故率は，学習環境においては改善されているが，未知の環境においては事故率が改善していないことがわかる．これは，学習データセット内の歩行者が含まれるシーンが不足し，その他のシーンと比べて歩行者に対する視線推定が困難であることが考えられる．

表 1：**NoCrash Benchmark** 成功率の比較結果 [%]

実験条件	交通条件	CILRS	Ours
Training condition	Empty	92	**95**
	Regular	63	**77**
	Dense	15	**25**
New weather	Empty	92	**96**
	Regular	**64**	60
	Dense	8	**28**
New town	Empty	54	**66**
	Regular	29	**46**
	Dense	8	**12**
New town & weather	Empty	**72**	66
	Regular	**44**	42
	Dense	8	**12**
平均		45.8	**52.1**

表 2：衝突事故によるエピソード失敗率の比較結果 [%]

実験条件	交通条件	失敗原因	CILRS	Ours
Training condition	Regular	Col. Ped.	9	**4**
		Col. Veh.	21	**11**
	Dense	Col. Ped.	21	**16**
		Col. Veh.	57	**47**
New weather	Regular	Col. Ped.	**6**	14
		Col. Veh.	22	**18**
	Dense	Col. Ped.	**14**	16
		Col. Veh.	68	**48**
New town	Regular	Col. Ped.	9	9
		Col. Veh.	30	**16**
	Dense	Col. Ped.	**10**	16
		Col. Veh.	68	**58**
New town & weather	Regular	Col. Ped.	**6**	10
		Col. Veh.	24	**16**
	Dense	Col. Ped.	**14**	20
		Col. Veh.	70	**52**
平均		Col. Ped.	**11.1**	13.1
		Col. Veh.	45.0	**33.3**

5. おわりに

本研究では，視線情報を活用した一貫学習ベースの自動運転手法を提案した．視線推定モデルを用いて，運転シーンにおける重要な領域を強調することで，シミュレータ上での走行実験において運転精度が改善されることを確認した．今後は，歩行者シーンの精度向上や視線推定モデルの特徴量を運転制御モデルに利用した手法の考案などを検討する．

参考文献

[1] M. Bojarski, *et al.*, “End to End Learning for Self-Driving Cars ”, arXiv preprint arXiv:1704.07911 2016.

[2] F. Codevilla, *et al.*, “Exploring the Limitations of Behavior Cloning for Autonomous Driving”, ICCV, 2019.

[3] A. Dosoviskiy, *et al.*, “CARLA: An Open Urban Driving Simulator”, CoRL, 2017.

研究業績

[1] 森啓介 等, “視線情報を利用した一貫学習ベースによる自動運転制御の高精度化”, ViEW, 2020.
(他 2 件)