

## 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な情報を探して、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。

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

### 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 每の特徴マップ  $F_i(x)$  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。

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

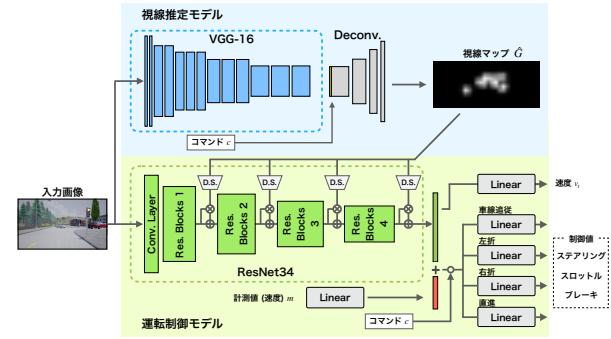


図 1：提案手法のネットワーク構造

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に、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。

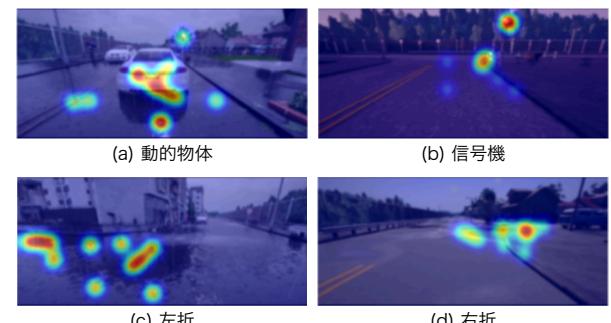


図 2：記録した視線情報の例

### 4.2 実験概要

視線推定モデルの入力は  $400 \times 176$  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  $200 \times 88$  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

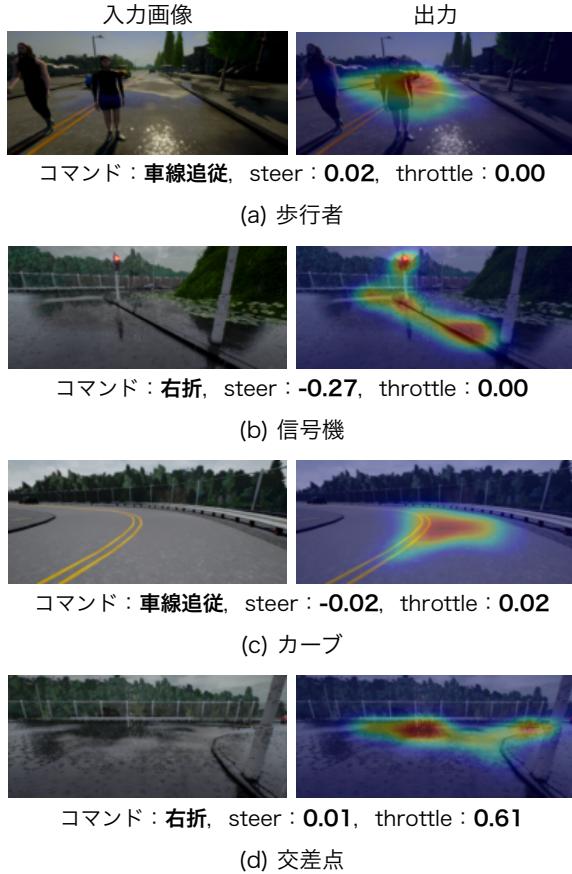


図 3：視線マップ推定例 : steer は負数で左, 正数で右のステアリング操作を表す. throttle は, 正数でアクセル操作を表す.

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

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	<b>95</b>
	Regular	63	<b>77</b>
	Dense	15	<b>25</b>
New weather	Empty	92	<b>96</b>
	Regular	64	60
	Dense	8	<b>28</b>
New town	Empty	54	<b>66</b>
	Regular	29	<b>46</b>
	Dense	8	<b>12</b>
New town & weather	Empty	<b>72</b>	66
	Regular	<b>44</b>	42
	Dense	8	<b>12</b>
平均		45.8	<b>52.1</b>

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	<b>4</b>	
		Col. Veh.	21	<b>11</b>	
New weather	Dense	Col. Ped.	21	<b>16</b>	
		Col. Veh.	57	<b>47</b>	
New town	Regular	Col. Ped.	<b>6</b>	14	
		Col. Veh.	22	<b>18</b>	
New town & weather	Dense	Col. Ped.	<b>14</b>	16	
		Col. Veh.	68	<b>48</b>	
New town	Regular	Col. Ped.	9	9	
		Col. Veh.	30	<b>16</b>	
New town & weather	Dense	Col. Ped.	<b>10</b>	16	
		Col. Veh.	68	<b>58</b>	
平均		Col. Ped.	<b>11.1</b>	13.1	
		Col. Veh.	45.0	<b>33.3</b>	

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.

#### 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. Dosovitskiy, et al., “CARLA: An Open Urban Driving Simulator”, CoRL, 2017.

#### 研究業績

- [1] 森啓介 等, “視線情報を用い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)