デジタル空間と深層学習を用いた道路車線検出法の評価

市川　竜也[1)]　　浅田　健吾[1)]　　松尾　裕一[1)]

Evaluation of a Road Lane Detection Method Using Digital Environment and Deep Learning

Tatsuya Ichikawa　　Kengo Asada　　Yuichi Matsuo

Many scenarios and driving data are required when road markings recognition for deep learning, so the use of simulation is becoming important. In this study, we re-created a traffic environment using MATLAB and UnrealEngine and used the generated data and deep learning to develop a system that can recognize road markings in difficult to see road markings situations and tested how well the systems adapts to real-world environments.

KEY WORDS: **Electronics and control, Image recognition system, Simulation**

1. まえがき

近年，より安全な自動車社会の形成のために自動運転技術の研究・開発が世界中で活発になっている．自動運転技術は認知，判断，予測の 3 つの要素技術に分けることができる．安全な自動運転のためには，周囲環境を正確に認知する必要がある．この認知の一つに道路に表示されている白線や黄線，破線などの道路区画線（車線）を認識する技術がある．周囲環境を認知するために機械学習が用いられるが，車線認識分野では実際に時間やコストをかけて作成された運転画像の公開データセットを学習に用いて車線の認識精度が競われている[(1)]．代表的なデータセットとしては Tusimple[(2)]が有名であり，高速道路のみの画像で構成されている．一方で，実際の運用では高速道路のみならず，様々な天候，車両の混雑，路面状況に対応する必要があり，環境に応じた学習データが必要である．しかしながら，天候や車両の混雑状況などは現実世界で意図した学習データを集めるのは非常に困難である．そこで，デジタル空間を活用して学習を行うことが考えられているが，未だにデジタル空間を活用した車線認識の研究は少ない．

そこで本研究では，デジタル空間で作成した画像（デジタル画像）を用いて CNN（Convolutional Neural Network）で学習した車線認識を提案する．デジタル空間の作成には MATLAB/Simulink[(3)]と Unreal Engine[(4)]を利用した．認識精度の評価には自動運転の認識性能評価に用いられる Precision と Recall，F 値を用いた[(5)]．評価対象として気候や時間を変化させた 5 種類のデジタル画像を用意した．また，デジタル画像で学習した車線認識が現実画像に対して機能するか，当大学周辺の道路を撮影した 3 枚の画像で確認した．

1) 東京理科大学（125-8585 東京都葛飾区新宿 6-3-1）

2. デジタル空間の構築

近年，自動運転の分野に限らずデジタル空間の活用が注目されており，自動運転プラットフォームやシミュレータがいくつか開発されている[(6)(7)]．本研究では機械学習ライブラリが豊富で新たな方法の実装がしやすい MATLAB/Simulink と，デジタル空間の構築に対して，リアルなモデル・環境を構築可能なゲームエンジン UnrealEngine を用いた．

2. 1. MATLAB/Simulink と Unreal Engine の連携

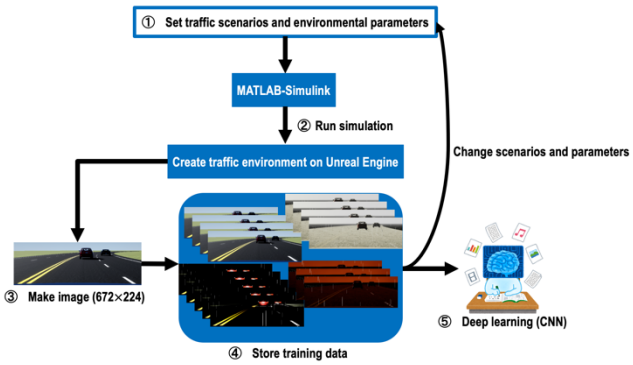


Fig.1 Present framework for road lane detection method

MATLAB はアルゴリズム開発やデータ解析，可視化，数値計算などの統合開発環境である．現在，MATLAB では自動運転開発のプラットフォームとしても力を入れており，MATLAB の Simulink 機能を用いることでモデルベースに自動運転の開発が可能となっている．また光環境の再現性と高い画像処理性能が特徴のゲームエンジン UnrealEngine と Simulink の連携により Simulink で作成したモデルを UnrealEngine 上でシミュレートすることが可能となる．

図 1 に本研究における車線認識法の流れを示した．①交通シナリオと天候パラメータの設定，②MATLAB/Simulink のモデルのシミュレーションを実行し UnrealEngine 上に交通環境を再現，③画像を作成(672×224)，④学習データに追加，①～

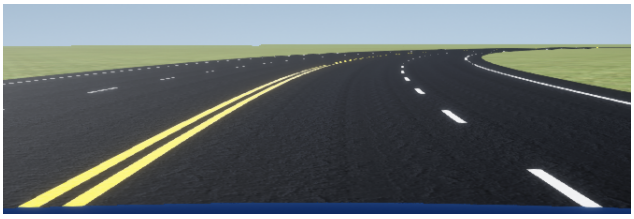
④を用意したシナリオと環境設定で繰り返し，⑥CNN を用いて学習する．

2.2. 学習データの生成

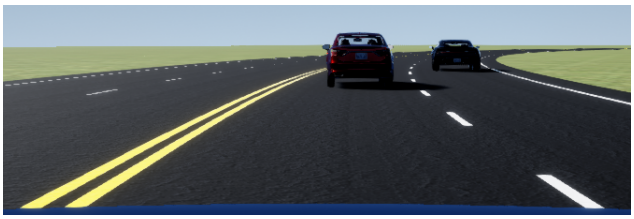
学習データの生成には MATLAB の車線認識システムのデモとして用意されている LaneMarkerDetectorTestBench にある簡単な交通状況が再現されているシナリオを用いた．シナリオは全て片側 2 車線の右側走行となっており，6 種類用意されている．本研究では 6 つのシナリオの内 2 つのシナリオを学習用に利用し，テストシナリオには実際の交通状況を想定して他車両が存在するシナリオを用いた．図 2 に本研究で利用した学習とテストのシナリオイメージを示した．学習シナリオ 1（図 2(a)）は直線道路で右車線を走行し車両は 1 台．学習シナリオ 2（図 2(b)）は曲線道路で左車線を走行し車両は 1 台．テストシナリオ（図 2(c)）は曲線道路で左車線を走行し車両は 3 台が混在する．

これらのシナリオで環境パラメータ 3 つとヘッドライトを変化させることで様々な状況を再現した．時間のパラメータは太陽光度（s）を変化させ，昼を 40 度，夕方を 0 度，夜を

(a) Training scenario1



(b) Training scenario2



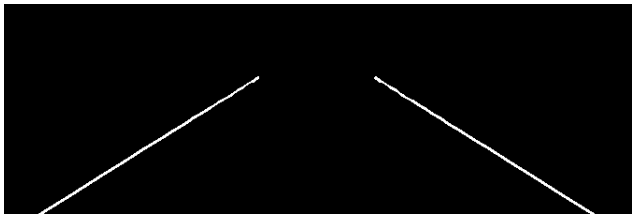
(c) Test scenario

Fig.2 Scenario images (672×224)

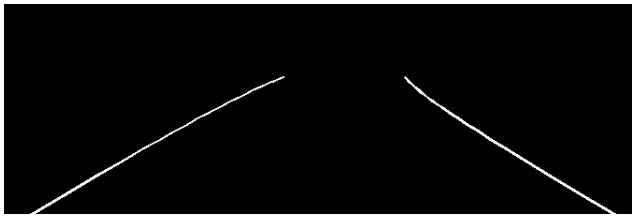
−90 度とした．天候のパラメータは霧（f）と雨（r）を用いた．霧のある状態とない状態を 100%と 0%に設定し，雨は降っていない状態を 0%，弱く降っている状態を 25%，強く降っている状態を 100%に設定した．また車のヘッドライト（l）はハイビームを使用した状態と使用しない状態の光量を 0cd と 10^6cd に設定した．

2.3. ラベルデータの生成

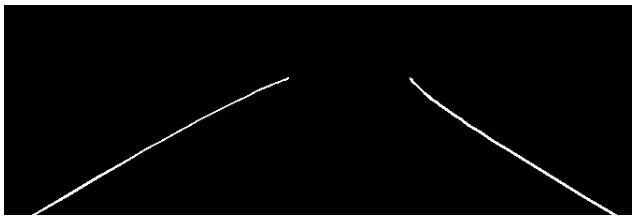
ラベルデータの生成も LaneMarkerDetectorTestBench の機能を用いて作成した．このデモには車線を認識して表示する機能があり，認識結果を評価するための正解データが用意されている．その正解データをラベルデータとして利用した．図 3 に 2 つの学習シナリオとテストシナリオのラベルデータを示す．



(a) Label data for training scenario1



(b) Label data for training scenario2



(c) Label data for test scenario

Fig.3 Label data images (672×224)

3. 車線認識の学習

本研究では活性化関数に ReLU（rectified linear unit）アルゴリズムを用いた CNN によりセマンティックセグメンテーションネットワークを構築し，車線の認識システムを作成した．構築したネットワークを表 1 に示した．

Table1 Structure of the CNN

layer	type	Output Size
1	Convolution1	672 × 224 × 64
2	Pooling1	336 × 112 × 64
3	Convolution2	336 × 112 × 64
4	UpSampling	672 × 224 × 64
5	Convolution3	672 × 224 × 2

セマンティックセグメンテーション[8]は各ピクセル単位でカテゴリ分類を行う手法で，車線認識の分野においてもよく用いられる手法である．CNN は画像認識の分野だけでなく音声や時系列データなど分類問題に広く利用されている手法であり，車線を認識する際に雨や霧などのノイズが多い場合にも高い認識精度を有することがわかっている[9]．CNN は

Convolution と呼ばれる畳み込み層と Pooling と呼ばれるプーリング層によって主に構成される．以下で簡単に畳み込み層とプーリング層について記す[10][11]．

畳み込み層は画像サイズ$W \times H$のKチャネル（RGB の場合チャネル数は 3，グレー画像の場合チャネル数は 1）の画像x_{ijk}（$(i,j,k) \in [0, W-1] \times [0, H-1] \times [0, K-1]$）を入力とし，この画像サイズを$W \times H \times K$と表す．この畳み込み層を第$l$層とし，直前の第$l-1$層から$K$チャネルの画像$z_{ijk}^{(l-1)}$を受け取り，これに$M$種類のフィルタを畳み込む作業を行う．各フィルタは入力と同じチャネル数Kを持ち，そのサイズを$L \times L \times K$とする．フィルタを$h_{pqkm}((p,q,k,m) \in [0, L-1] \times [0, L-1] \times [0, K-1] \times [0, M-1])$と表す．$M$種類のフィルタについて並行に計算が実行され，それぞれ 1 チャネルのu_{ijm}が出力される．この計算は各チャネル$k(k = 0, \dots K-1)$について並行に画像とフィルタの畳み込みを行った後，結果を画素ごとに全チャネルにわたって加算するもので，

$$u_{ijm} = \sum_{K-1}^{k=0} \sum_{L-1}^{p=0} \sum_{L-1}^{q=0} z_{i+p,J+q,k}^{(l-1)} h_{pqkm} + b_m \tag{1}$$

と表せる．b_mはバイアスでチャネルmごとに全出力ユニット(i,j)間（全画素間）で共通とする．このように，入力画像のチャネル数によらず 1 つのフィルタからの出力は常に 1 チャネルになる．

次に得られたu_{ijm}に活性化関数である ReLU を適用し

$$z_{ijm} = f(u_{ijm}) \tag{2}$$

が得られる．活性化関数fにはシグモイド関数や$tanh$関数などがあるが，近年の深層学習では ReLU が利用されている．ReLU は単純で計算量が小さく，上述の 2 つの関数よりも学習がより早く進み，最終的により良い結果が得られることが多い[10]．ReLU は次の式（3）で表すことができる．

$$f(u) = \max(u, 0) \tag{3}$$

畳み込み層の入力と出力のユニット数は$W \times H \times K$と$W \times H \times M$であり，局所性を反映すると出力層のユニット 1 つ（チャネルmの 1 つの画素）は入力層の$W \times H \times K$個のユニットのみと結合する．その結合の重みがフィルタの係数h_{pqkm}（mは今考えているチャネル）となり，この重みは出力層の同じチャネルの全てユニットで共有される．これを重み共有と呼び，結合の局所性と重みを共有することが畳み込み層の特徴である．

プーリング層は基本的に畳み込み層の直後に設置され，畳み込み層の出力がプーリング層の入力となる．プーリング層での計算は次のように表すことができる．サイズ$W \times H \times K$の画像上で画素(i,j)を中心とする$P \times P$正方領域をとり，この中に含まれる画素の集合P_{ij}で表す．P_{ij}内の画素についてチャネルkごとに独立にP^2個ある画素値を使って 1 つの画素値u_{ijk}を求める．求める方法はいくつかあるが，本研究では最大プーリングを用いて求めた．最大プーリングはP^2個の画素値の最大値を選び，式（4）で表すことができる．

$$u_{ijk} = \max_{(p,q) \in P_{ij}} z_{pqk} \tag{4}$$

CNN の学習パラメータはバッチを 32，エポック数を 100，初期学習率を10^{-3}，最適化手法は adam を利用した．バッチは学習データをバッチサイズに分割することで複数セットにわたって学習をする指標で，使用機器の GPU のメモリが処理できる最大の 32 で行った．エポックはバッチサイズに分割した学習データセットを計何回学習するかの指標であり，学習が進み学習精度が収束する 100 で行った．初期学習率と最適化手法は学習中に取り出された特徴量を次の学習に用いるとき，どれだけ重要な特徴であるかを更新する際に用いられるものである．初期学習率を小さくすることで一度の学習で大きな更新を防ぎ，学習精度の頭打ちを回避することができる．adam はニューラルネットワークで主な最適化手法であるため利用した．学習データは学習シナリオ 1 が 1890 枚と学習シナリオ 2 が 1890 枚の計 3780 枚の画像を用意した．

4. 評 価 方 法

認識精度の評価指標として Precision, Recall, F 値を用いる．各式は次のように表される．

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$F\ measure = \frac{2Precision \cdot Recall}{Precision + Recall} \tag{7}$$

Precision と Recall で用いられている TP と FP，FN は混同行列から求めることができる．混同行列を表 2 に示す．

Table2 Confusion Matrix

		Prediction	
		lane	other
Truth	lane	TP	FN
	other	FP	TN

TP は認識システムが車線であると予測したうち正解のものである．FP は認識システムが車線であると予測したが正解でなかったものである．FN は認識システムが車線でないと予測したが実際には車線であったものである．Precision だけで評価を行うとどれだけ車線を見逃して認識しているのかがわからず，Recall だけで評価を行うとシステムの正確度がわからない問題がある．そのため，全体的な TP の割合を評価するために Precision と Recall の調和平均である F 値で評価を行う．

5. 結 果 と 考 察

5.1. デジタル画像に対して

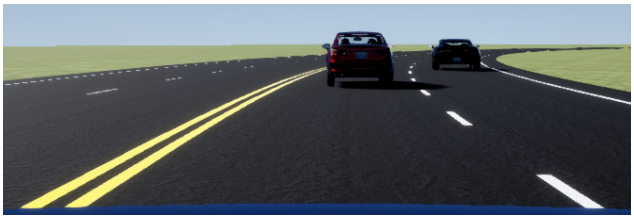
デジタル画像へのテストとしてテストシナリオ（図 2 (c)）を用いて 5 種類の環境パラメータ設定で評価を行った．表 3

にデジタル画像のテストケースと環境パラメータの設定を示した.

Table3 Set test cases and environment parameters

Test case	s	f	r	l
Case1	40	0	0	0
Case2	0	0	100	0
Case3	0	100	100	0
Case4	-90	0	100	10^6
Case5	-90	100	100	10^6

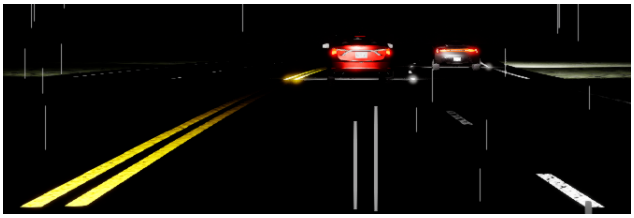
s, f, r, l は太陽高度と霧の濃度, 降水量, ヘッドライトの光量を表している.



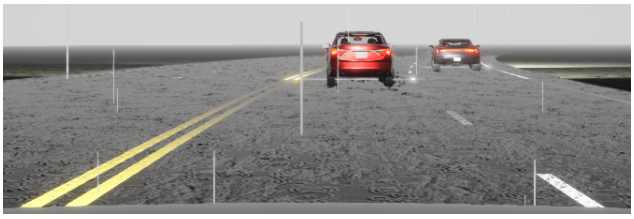
(a) Case1

(b) Case2

(c) Case3



(d) Case4



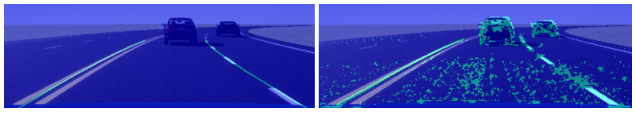
(e) Case5

Fig.3 Test case images (672×224)

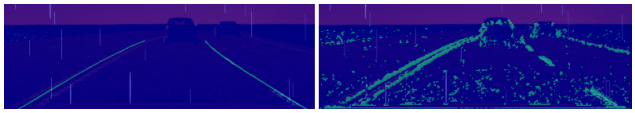
各テストケースのサンプル画像を図 3 に示した. Case1～Case5 について Precision と Recall, F 値の結果を表 4 に示す. また図 4 に各ケースのサンプルにラベルデータを重ねた画像, サンプルに車線認識システムに通した結果を重ねた画像を示す.

Table4 Evaluation results of test cases

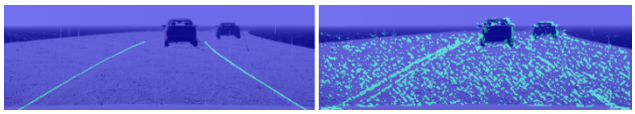
	Case1	Case2	Case3	Case4	Case5
Precision	8.89	7.05	3.60	10.79	4.20
Recall	93.49	74.77	71.25	69.96	81.28
F measure	16.24	12.89	6.85	18.70	7.99



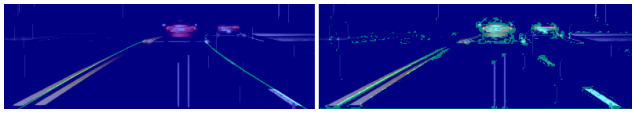
(a) Case1



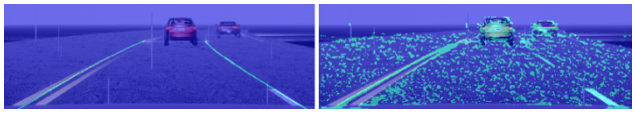
(b) Case2



(c) Case3



(d) Case4



(e) Case5

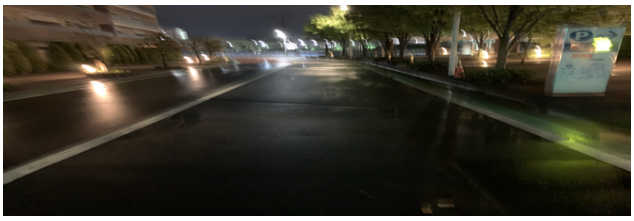
Fig.4 The left image is overlaid label image (672×224), the right image is overlaid result image (672×224)

図 4 の左の画像で青緑色に表示されているものがラベルデータ. 右の画像で青緑色に表示されているものが車線と認識された結果である. 表 4 を見ると Recall は高いものの Precision が非常に低いため Precision と Recall の調和平均である F 値が全ケース共通で低いことがわかる. ここで図 4 の結果と合わせて見てみると, 車線の形状を認識できていることがわかる. しかし, 道路上や車両自体が車線と誤認識されているため, 認識した車線の正確度である Precision が低い結果となり, 車線らしきものを多く車線と認識しているため, 車線を漏れなく認識できたかの指標である Recall が比較的高い結果となった. また, 表 4 の結果は実際に車線認識分野で競われているベンチマークの結果[1][2]と比較すると低いため改善が必要であるとわかる. 次に図 4 の結果を見ると今回作成したラベルデータの形状や位置が適切でないことがわかる. このずれが学習と評価に影響を及ぼしていると考えられる. ラベルデータを修正することで Precision と Recall が改善されそれに伴い F 値

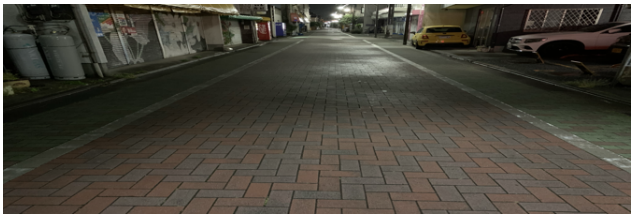
の向上も期待される

5. 2. 現実画像に対して

デジタル画像で学習した車線認識が現実画像に対して機能するかの確認に 3 つの画像 Case6～Case8 を用意した．これは東京理科大学葛飾キャンパスの周辺で簡易的に撮影したものである. ラベルデータは MATLAB のイメージラベラーを利用して手作業で作成した．図 5 に利用した現実の各画像を示した.



(a) Case6



(b) Case7

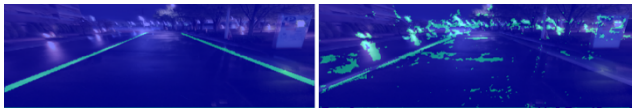
(c) Case8

Fig.5 Real images (672×224)

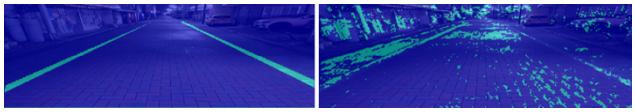
Case6～Case8 に対する結果を表 5 に示し, 現実画像のラベルデータとデジタル画像で学習した車線認識に通した結果を図 6 に示す．図 6 は図 4 と同じく左の画像の青緑色で表示されているものがラベルデータで右の画像の青緑色で表示されているものが車線と認識された結果である. 表 5 を見ると Case8 の Recall を除いた Precision と Recall が低いことがわかり，調和平均である F 値も低い結果となっている．デジタル画像の結果である表 4 と比べると特に Recall が劣っていることがわかる．次に図 6 の結果と合わせ見ると，Case6 と Case7 は左の車線はある程度検出されているが右の車線が全く検出されておらず Recall の結果が約 33%と約 27%となっている．Case8 は車線の形がある程度検出されているため約 68%の結果が得られており，Precision については誤検出が多いため他のケースと同様に低い結果となっている．現実画像に対する結果はデジタル画像の結果と比較して全体的に低いため，Case6～Case8 に似たデジタル画像を加えて学習すると結果がどれほど改善されるか今後確認を行う．

Table5 Evaluation results of real images

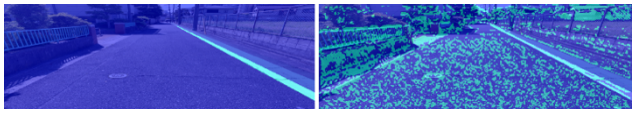
	Case6	Case7	Case8
Precision	11.81	7.47	3.93
Recall	33.51	27.39	68.86
F measure	17.46	11.74	7.44



(a) Case6



(b) Case7



(c) Case8

Fig.6 The left real image is overlaid label image (672×224), the right real image is overlaid result image (672×224)

6. 結 論

本研究では, デジタル空間に交通環境を再現し, そこから得たデジタル画像を用いて CNN で学習を行い車線認識までの一連の流れを行った. また, 作成した車線認識を適切に評価するために Precision と Recall を用いた F 値で評価を行った．学習データと同じ形式のデジタル画像へのテストのみならず, 現実画像への適用度の確認も行った．以上のことからデジタル画像で学習した車線認識は現実の車線の認識に一定の効果があることがわかった. しかし, 実用レベルには至っていないため, 今後さらに改良を行い実用レベルの認識を目指す. 今後取り組むべき項目について以下に示す.

1) ラベルデータの形状の修正
2) データ数やネットワーク構造の修正
3) 車線認識分野のベンチマークでの評価
4) シナリオや道路形状などの追加
5) デジタル空間の再現度向上

参 考 文 献

(1) Zheng, T., Huang, Y., Liu, Y., Tang, W., Yang, Z., Cai, D., He, X.: CLRNet: Cross Layer Refinement Network for Lane Detection, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 898-907 (2022)

(2) TuSimple. Tusimple benchmark. https://github.com/TuSimple/tusimple-benchmark, Accessed August, 2022

(3) MATLAB. MATLAB Simulink. https://jp.mathworks.com/, Accessed August, 2022

(4) Unreal Engine. https://www.unrealengine.com/ja, Accessed August, 2022
(5) CARLA. https://carla.org/, Accessed August, 2022
(6) Autoware. https://www.autoware.org/, Accessed August, 2022
(7) 中川正夫，小林摂，新国哲也：自動運転車の認識性能の正確性評価とその評価手法の妥当性に関する考察，自動車技術会論文集，vol.52, No.3, p.633-638 (2021)
(8) Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation, arXiv preprint arXiv:1511.00561 (2015)
(9) Kim, J., Lee, M.: Robust lane detection based on convolutional neural network and random sample consensus, Neural Information Processing, p.454-461 (2014). Springer International Publishing.
(10) 岡谷貴之：深層学習，東京，講談社，2015，165p
(11) 有水賢吾，毛綱昌弘：深層畳み込みニューラルネットワークを用いたセマンティックセグメンテーションによる森林作業道抽出，森林利用学会誌，vol.35, No.1 p.7-13 (2020)