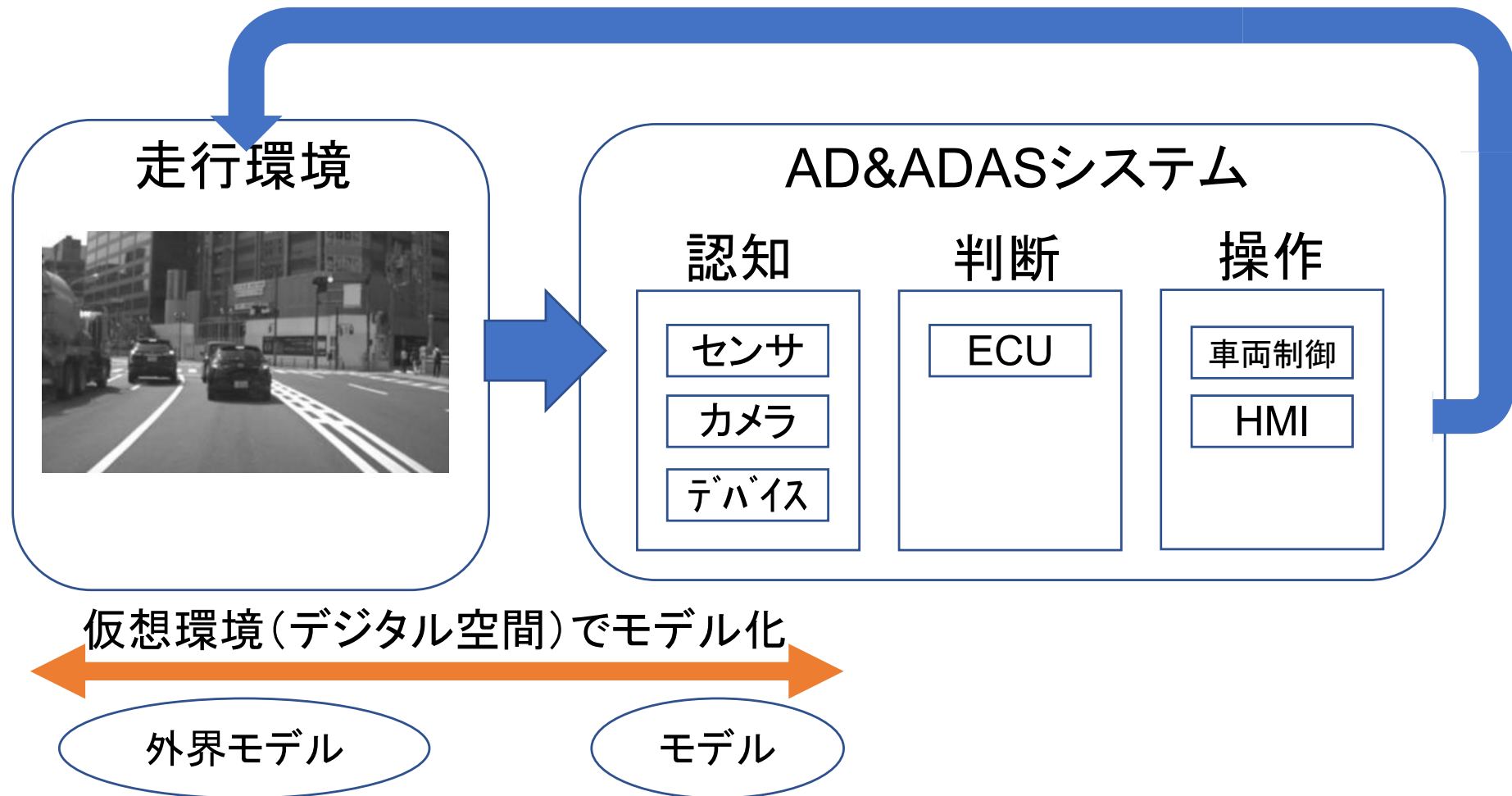


自動車技術会 2022秋季大会 学術講演会

# デジタル空間と深層学習を用いた 道路車線検出の評価

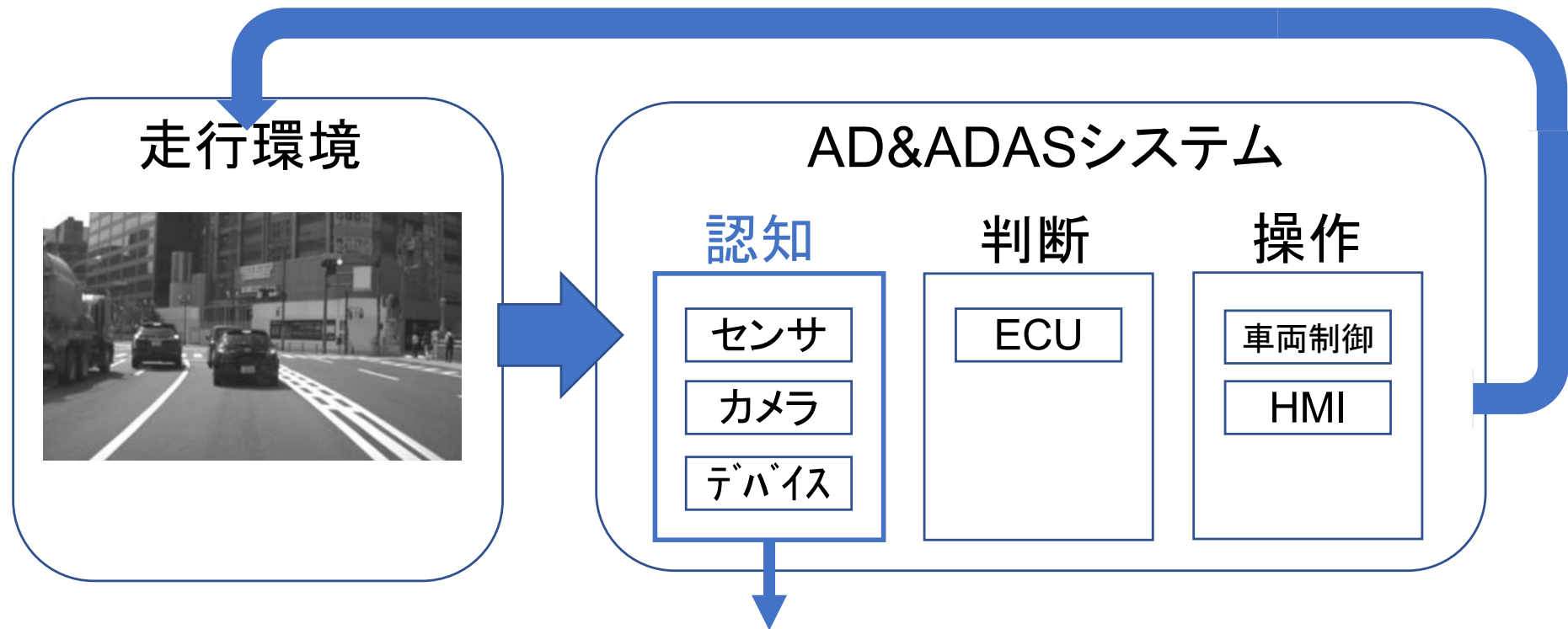
市川竜也, 浅田健吾, 松尾裕一  
東京理科大学 工学研究科 情報工学専攻

2022/10/14



検証システムの網羅性や開発効率を向上

AD=自動運転, ADAS=先進運転アシストシステム



周囲環境を正確に認知する必要  
ex: 道路区画線(車線)検出

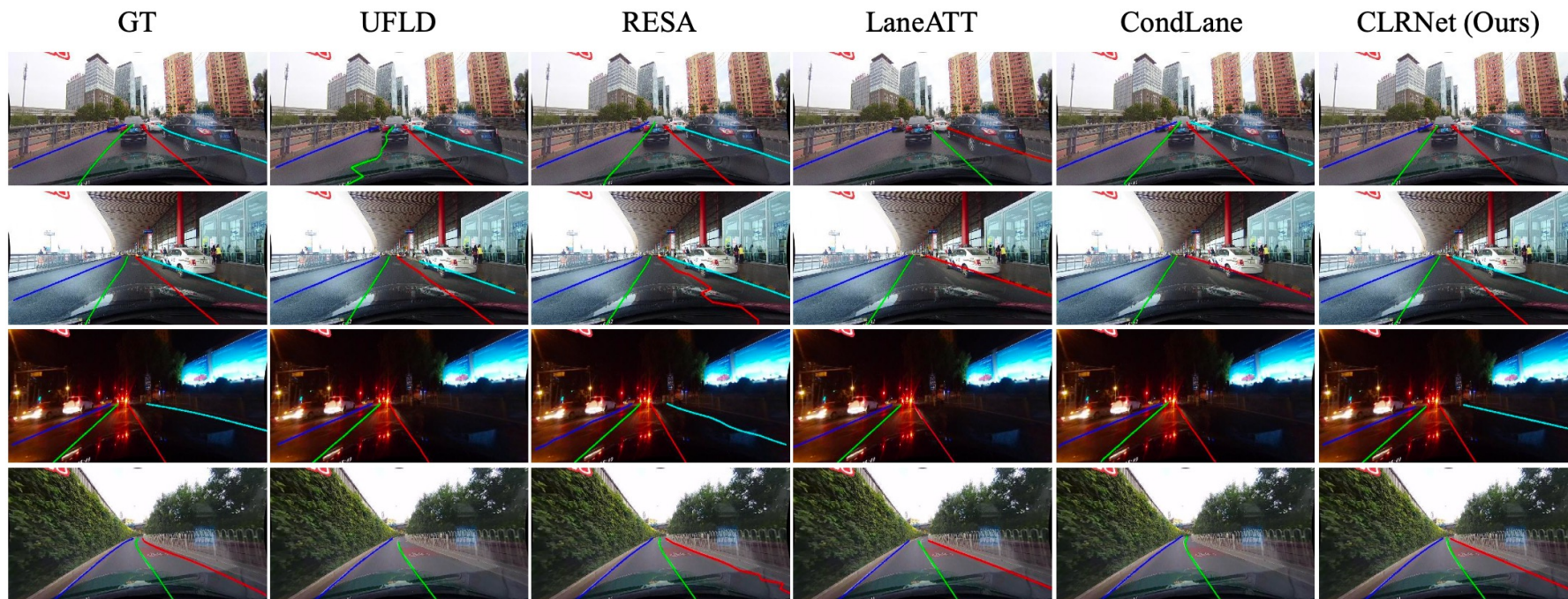
車線検出は機械学習が主に用いられる

学習データは実際に撮影された公開運転画像データセット

➤ 様々な状況の画像はコストと時間がかかる

## ◆車線検出(認識)分野

- CNN(Convolutional Neural Network)を活用した研究が多い<sup>[1]</sup>
- 学習データは公開されているデータセット  
TuSimple<sup>[2]</sup>は高速道路のみの運転画像(2017)  
様々な状況のデータセットが作成されている<sup>[3]</sup>



[1] CLRNet: Cross Layer Refinement Network for Lane Detection, IEEE, 2022

[2] TuSimple. Tusimple benchmark. <https://github.com/TuSimple/tusimple-benchmark>, Accessed August, 2022

[3] CULane. <https://xingangpan.github.io/projects/CULane.html>, Accessed August, 2022

## ◆デジタル空間

実機撮影結果



シミュレーション結果（天空データ:晴れ）



- 実現象と一致性の高いカメラやLidarなどのセンサモデルの構築<sup>[4]</sup>
  - 公開データセットを用いた研究と比較して  
デジタル空間で作成したデジタル画像を用いた  
車線検出の研究は少ない

[4] 仮想空間での自動運転安全性評価環境の構築, 井上秀雄, SIP-adus Workshop 2021 Plenary Session

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  
車線検出の評価

アプローチ

- デジタル空間 : MATLAB/Simulink<sup>[5]</sup>, Unreal Engine<sup>[6]</sup>
- 車線検出 : CNN
- 評価方法 : Precision, Recall, F measure<sup>[7]</sup>

[5] MATLAB. MATLAB Simulink. <https://jp.mathworks.com/>, Accessed August, 2022

[6] Unreal Engine. <https://www.unrealengine.com/ja>, Accessed August, 2022

[7] 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, 2021



① Set traffic scenarios and environmental parameters



MATLAB-Simulink



② Run simulation

Create traffic environment on Unreal Engine

## ◆ 交通シナリオと環境設定

MATLABにある車線認識システムのデモ

Lane Marker Detector Test Benchに備わっている

交通シナリオと天候や時間などの環境設定

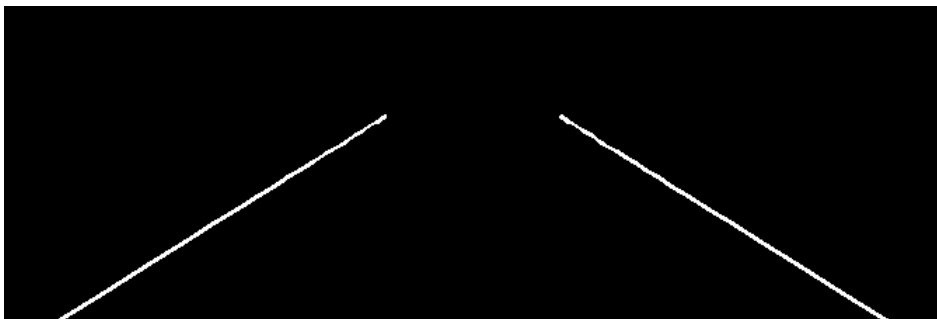


(a) Training scenario1

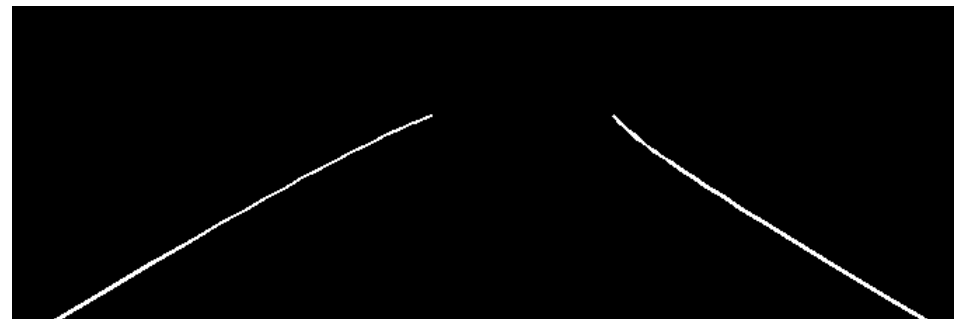


(b) Training scenario2

- 交通シナリオは車線認識のデモから(a)と(b)を利用
- (a)は直線道路の右を走る, (b)は曲線道路の左を走る
- 両シナリオ共に他車両は存在しない



(a) Label data for training scenario1



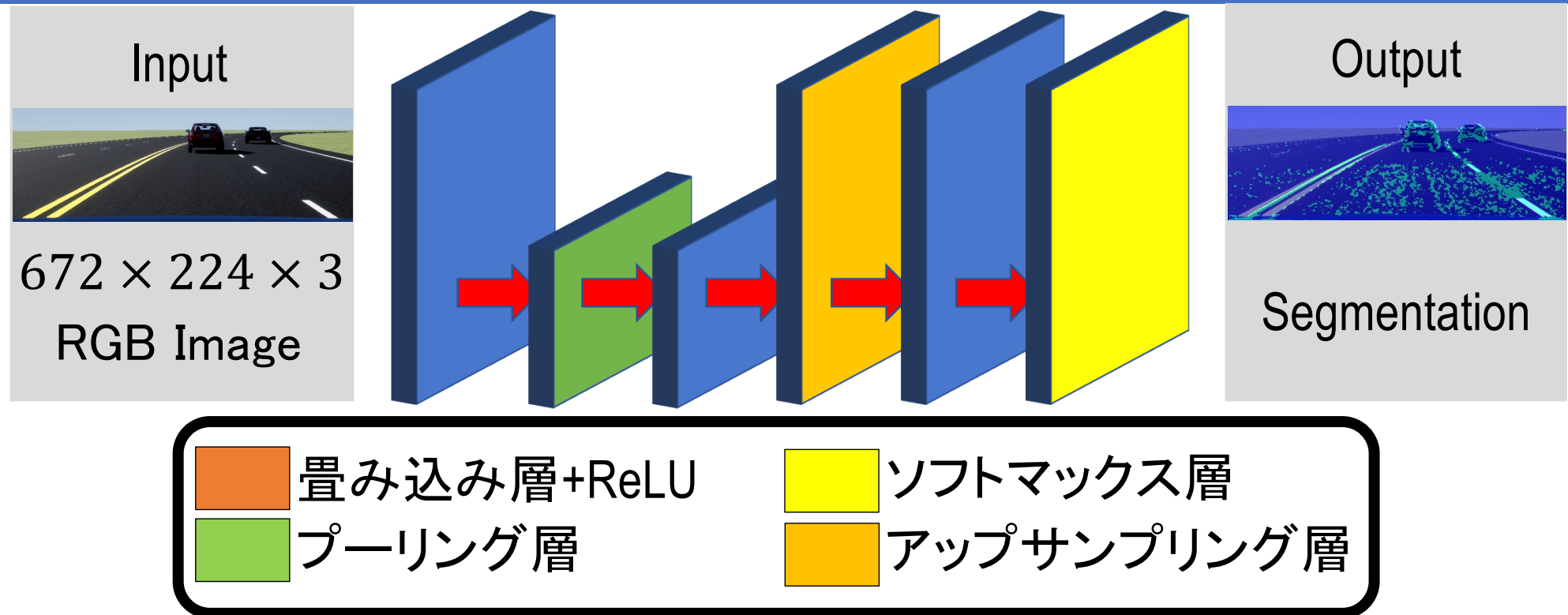
(b) Label data for training scenario2

- MATLABのデモには認識した車線を評価するための正解データがあり, それをラベルデータとして利用



- 太陽高度(s), 霧(f), 雨(r)とヘッドライト(l)の4つのパラメータを用いて様々な状況のデータを作成

太陽高度(s)	昼を40度, 夕方を0度, 夜を-90度
霧(f)	霧がない状態を0%, ある状態を100%
雨(r)	降らない状態を0%, 弱く降る状態を25%, 強く降る状態を100%
ヘッドライト(l)	off を0cd, on を $10^6$ cd



- CNNを用いて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
- セマンティックセグメンテーションはピクセル単位で分類<sup>[8]</sup>
- CNNは雨や霧などのノイズに対してロバストな検出が可能<sup>[9]</sup>
- プーリングはMaxプーリング, 活性化関数はReLU (Rectified Linear Unit)

[8] SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation , arXiv preprint arXiv:1511.00561

[9] Robust lane detection based on convolutional neural network and random sample consensus, Springer, 2014

- バッチは学習データを分割して複数セットにわたり学習をする指標で使用機器のGPUが処理できる最大の数を利用
- エポックは学習が進み学習精度が収束する数を利用
- 初期学習率が大きいと学習がうまく収束しない
- CNNで主に利用される最適化手法

Batch	32
Epoch	100
Initial Learn rate	$10^{-3}$
Optimizer	adam

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

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

		Prediction	
		lane	other
Truth	lane	TP	FN
	other	FP	TN

$$F\ measure = \frac{2Precision \times Recall}{Precision + Recall}$$

- TPは車線であると予測した中で正解であるもの
- FPは車線であると予測したが正解でないもの
- FNはその他であると予測したが車線であったもの

Precisionだけの評価では車線の見落としがわからず、  
Recallだけの評価では認識の正確度がわからないため  
PrecisionとRecallの調和平均F値と合わせて評価する



Test case scenario

テストケースは他車両2台が入り混じるシナリオ

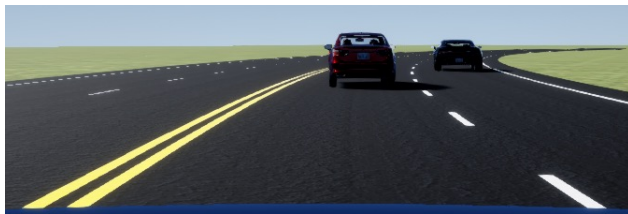
Test case	s	f	r	l
Case1	40	0	0	0
Case2	0	0	100	0
Case3	0	100	100	0
Case4	-90	0	100	$10^6$
Case5	-90	100	100	$10^6$

s, f, r, l は太陽高度, 霧, 雨, ヘッドライトの光量を表す

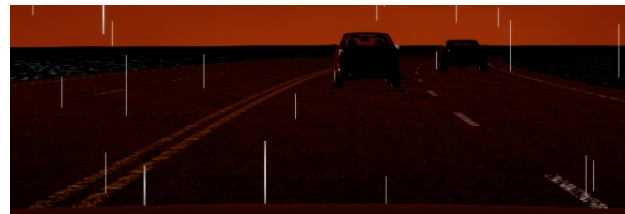
# デジタル画像への結果

13/18

	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



Case1



Case2



Case3



Case4



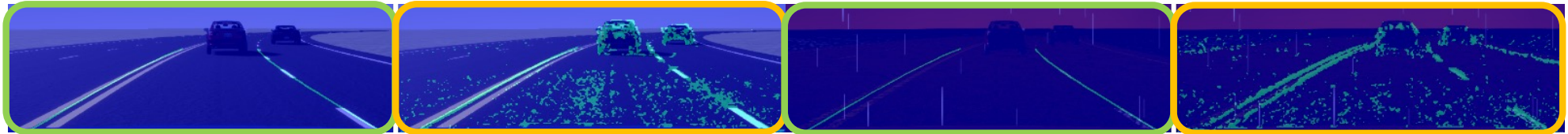
Case5

- 全体的にPrecisionが低いため調和平均のF値も低い
- Recallが高くPrecisionが低いことから車線の誤検出が多い



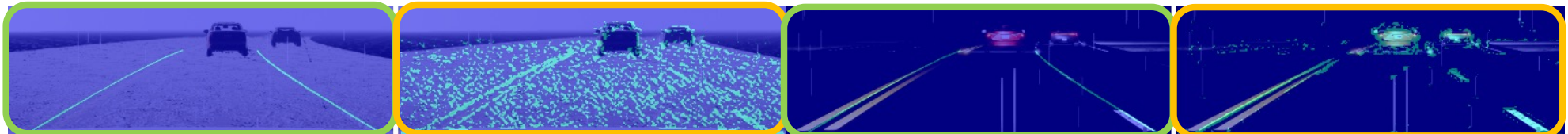
# デジタル画像への結果の可視化 14/18

- 見落としと誤検出の確認 (左はラベルデータ, 右は検出結果)



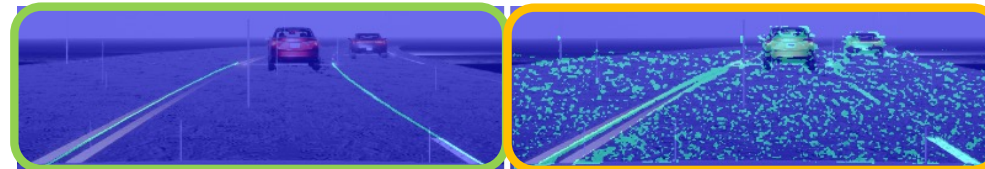
Case1

Case2



Case3

Case4



Case5

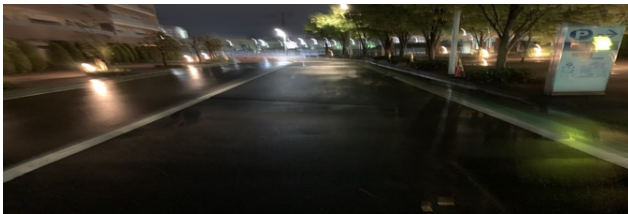
## ➤ 見落とし (Recall)

ラベルデータは破線を直線で作成, 検出結果は破線単位で検出

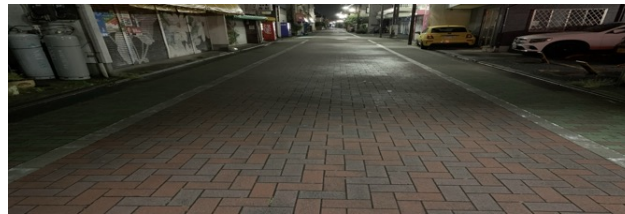
## ➤ 誤検出 (Precision)

車両や道路上に多く誤検出

	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



Case6



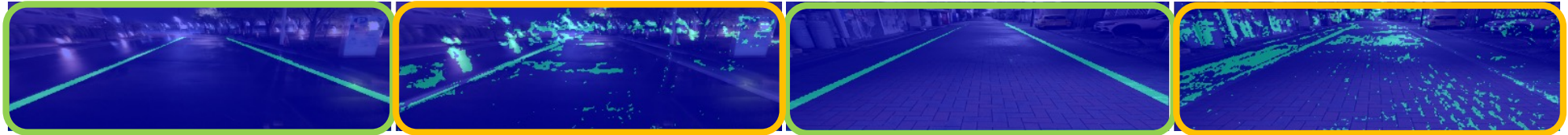
Case7



Case8

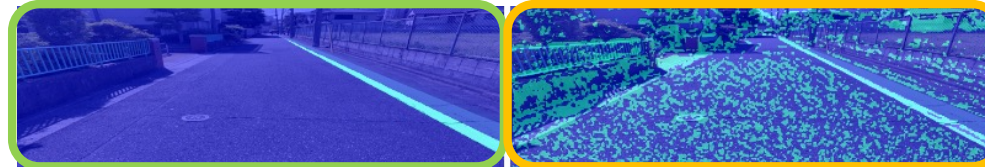
- デジタル画像への結果同様, Precisionが低く Recallが高い傾向
- デジタル画像の結果と比較してRecallの検出精度が落ちている
- Case8のPrecisionとRecallの差が大きい

- 見落としと誤検出の確認 (左はラベルデータ, 右は検出結果)



Case6

Case7



Case8

- 現実画像のラベルデータは手作業で作成

➤ 見落とし(Recall)

Case6, 7は右車線が上手く検出できていない

Case8は大まかな形状が検出できている

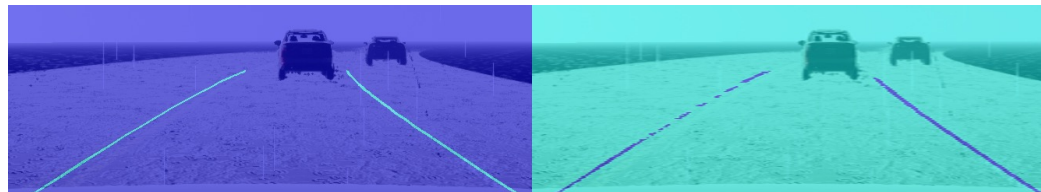
➤ 誤検出(Precision)

デジタル画像と同様に道路上の誤検出が多い(特にCase8)

**ResNet:** 残差(前の層の情報を足し上げるショートカット)を利用して  
多層の劣化問題を解消

## ● デジタル画像

	Case1	Case2	Case3	Case4	Case5
Precision	99.56	99.22	99.33	99.52	99.52
Recall	98.61	96.76	95.75	98.27	98.30
F measure	99.09	97.97	97.51	98.89	98.90



Case3

➤ 結果が白線(破線)単位というよりラベル単位で検出されている

## ● 現実画像

➤ デジタル画像へ過適合し現実画像にうまく適合できなかった

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で学習を行い車線検出までの一連の流れを行った．

1. 学習データと同じデジタル画像へのテスト
2. 現実画像への適用度の確認

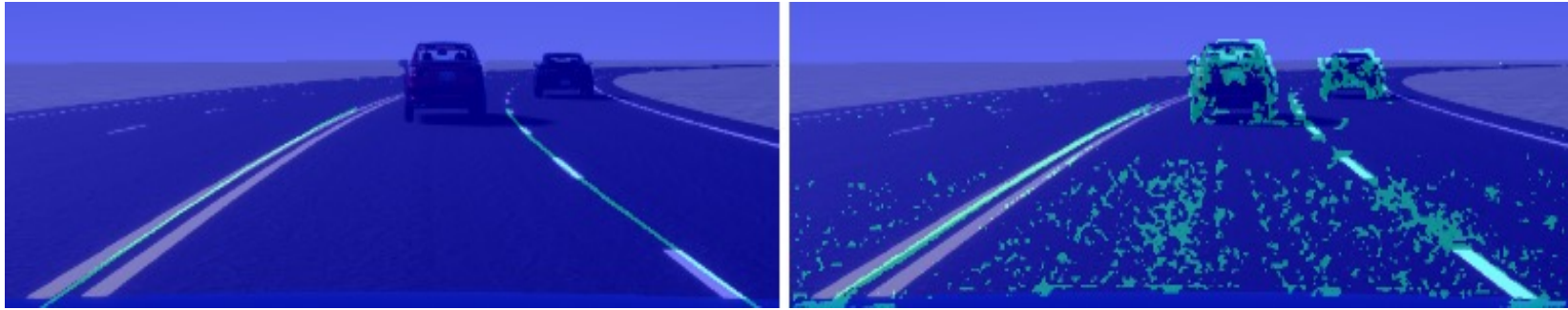
## 課題

- 適したラベルデータの作成環境の構築
- シナリオや道路形状の追加
- 車線検出に限らず画像認識分野の観点を持ってデジタル画像と現実画像のギャップの改善

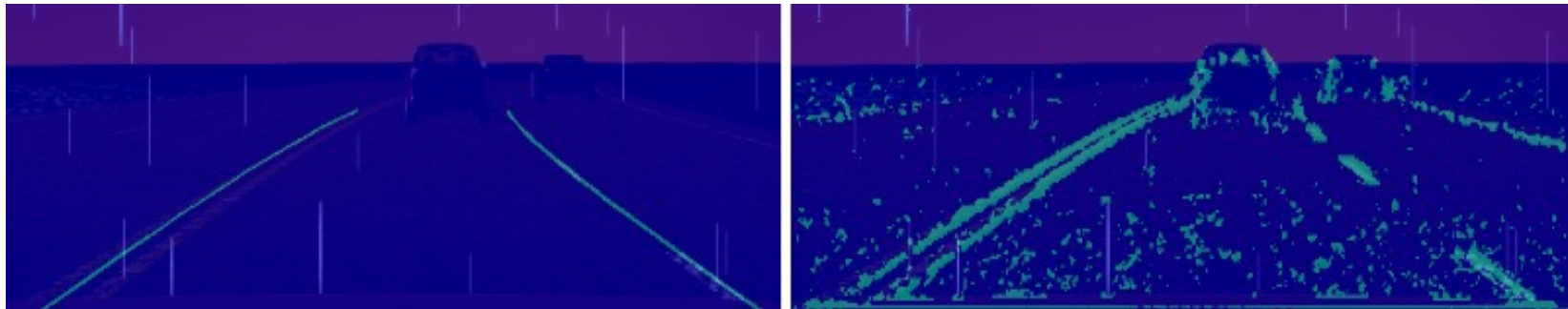
# 付録



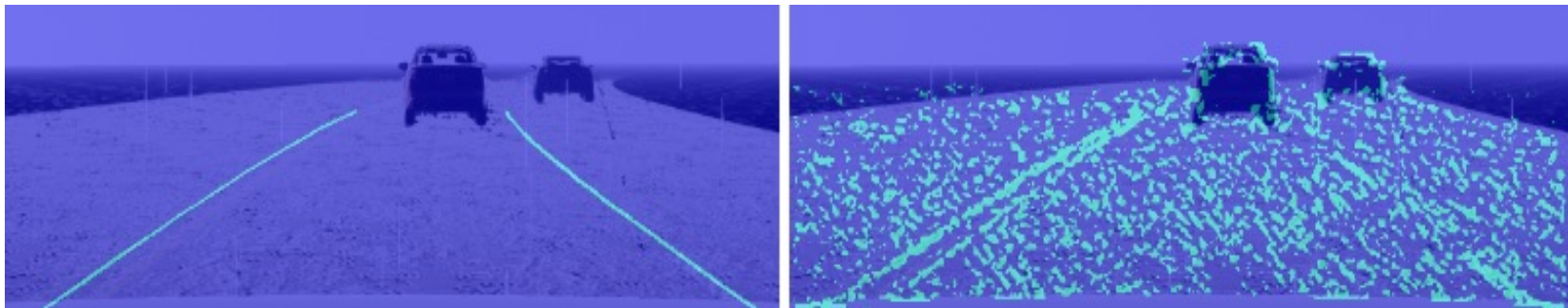
# デジタル画像への結果の可視化 21/18



Case1

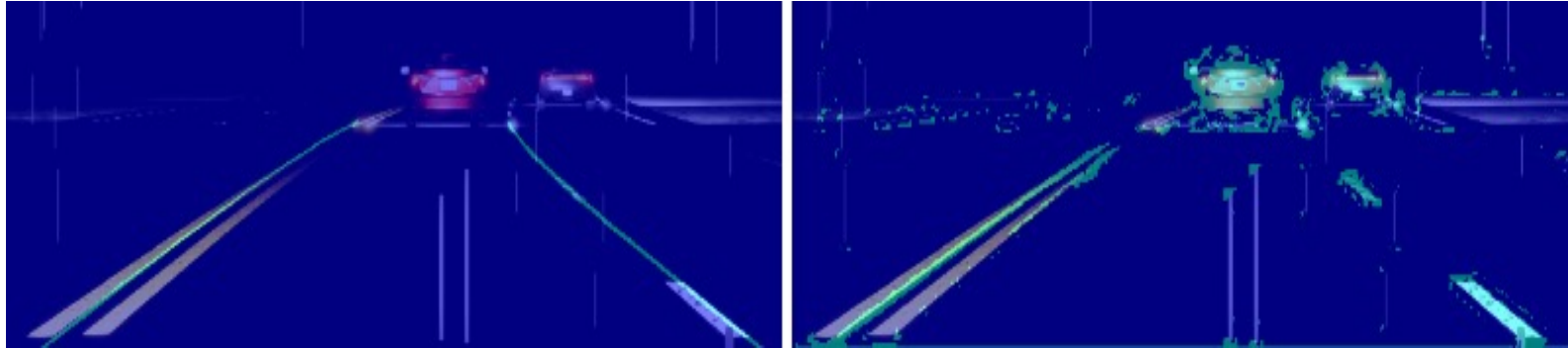


Case2

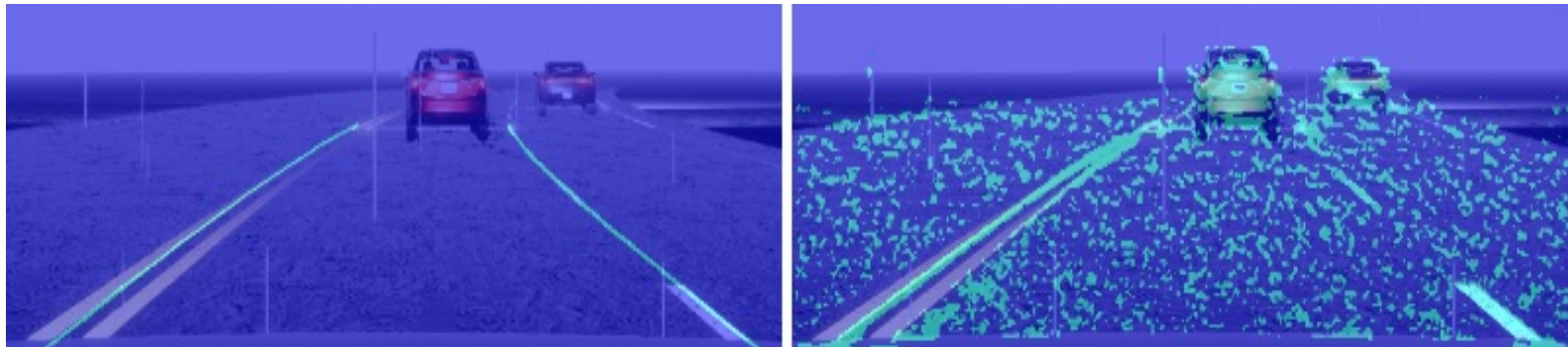


Case3

# デジタル画像への結果の可視化 22/18

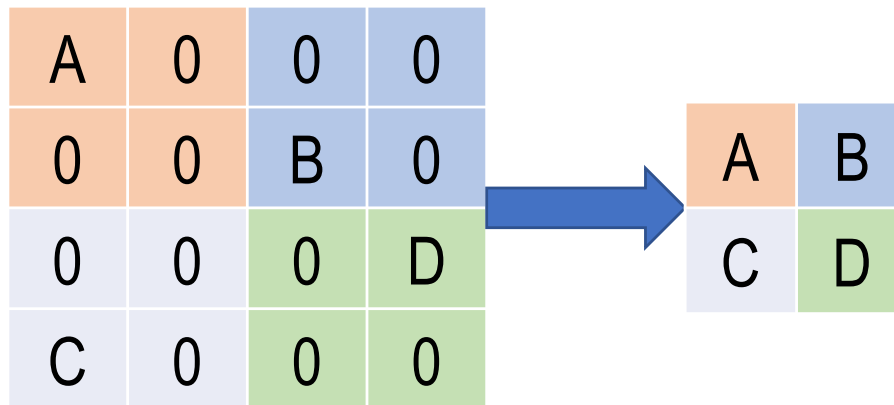


Case4



Case5

- Max Pooling



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、  
この中にある画素の集合 $P_{ij}$ とする。  
チャンネル $k$ ごとに独立に $P^2$ 個ある画素値を使い、1つの画素値 $u_{ijk}$ を求める。

最大プーリングは $P^2$ 個の画素値の最大値を選ぶ

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

プーリング層の出力は入力サイズよりサイズが縮小されるのが普通であり、  
その縮小比を決めるのがストライド $s$ の値である。

- CVPR(ComputerVisionPatternRecognition)2022  
ResNet101を利用した車線検出の研究例<sup>[1]</sup>

学習パラメータ	
Batch	64
Epoch	50
Initial Learn rate	$10^{-3}$
Optimizer	adam

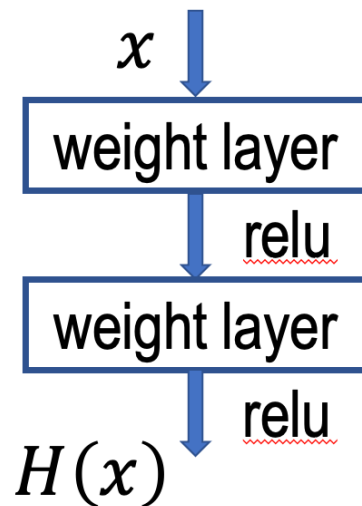
- ResNet101のアーキテクチャを利用した  
セマンティックセグメンテーションネットワークの構築

[1] CLRNet: Cross Layer Refinement Network for Lane Detection, IEEE, 2022

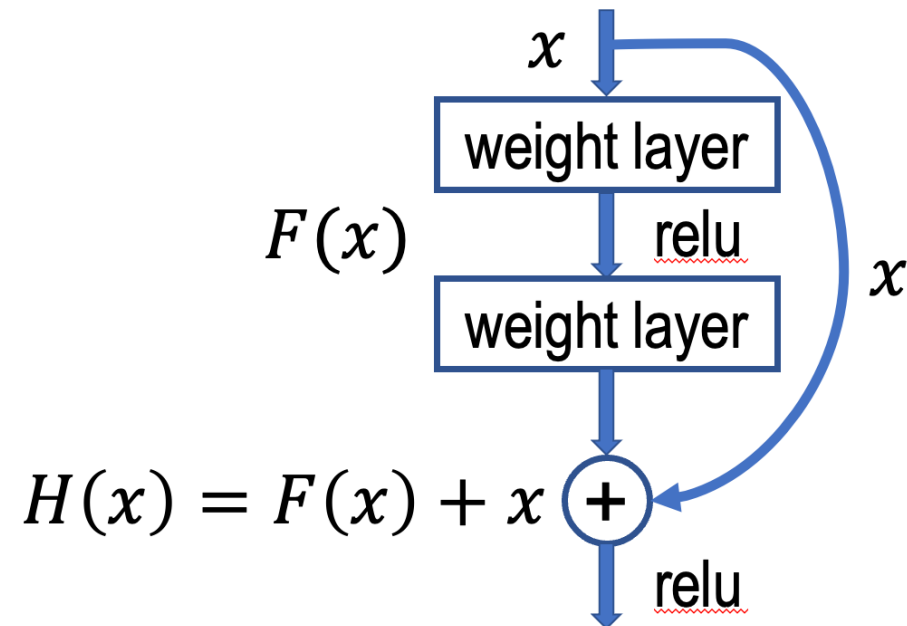
[10] Deep Residual Learning for Image Recognition, arXiv preprint arXiv:1512.03385

- ResNet(Deep Residual Network)<sup>[10]</sup>  
ImageNetテストセットの精度を競うILSVRC2015で優勝  
残差(Residual)の利用により多層での劣化問題を解消

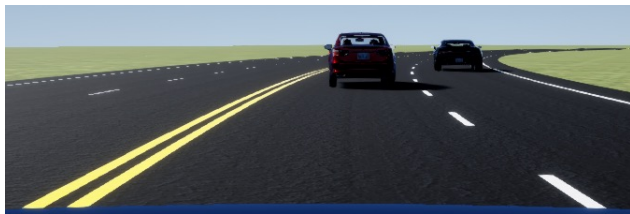
従来ネットワーク



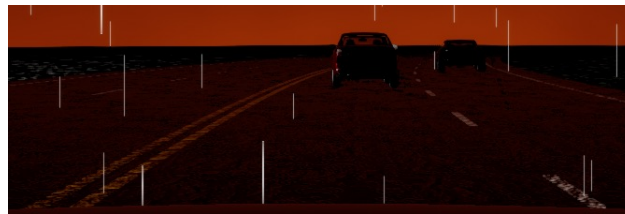
ResNet



	Case1	Case2	Case3	Case4	Case5
Precision	99.56	99.22	99.33	99.52	99.52
Recall	98.61	96.76	95.75	98.27	98.30
F measure	99.09	97.97	97.51	98.89	98.90



Case1



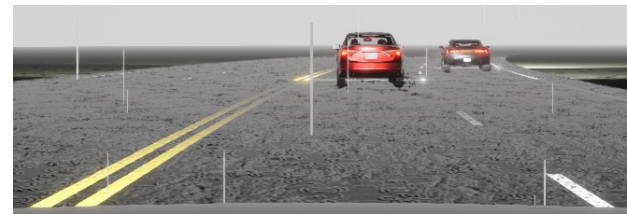
Case2



Case3



Case4



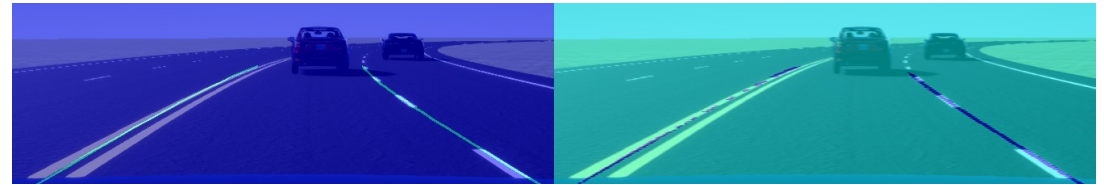
Case5

- 全体的にPrecision, Recall両方の精度が向上

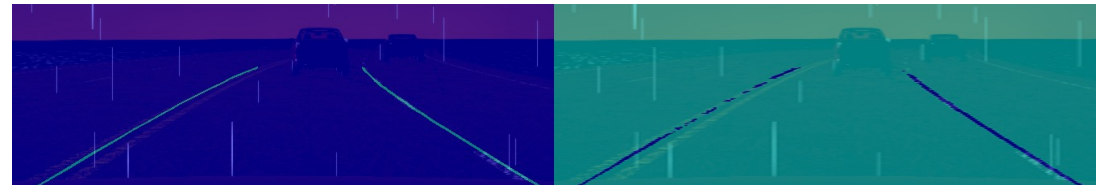


# デジタル画像への結果の可視化 28/18

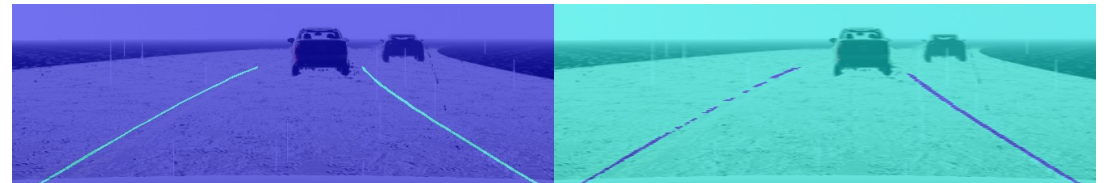
- 左の画像の青緑色はラベルデータを重ねたもの
- 右の画像の青色は車線と検出されたもの
- 白線単位の認識ではない
- ラベルの特徴を捉えている



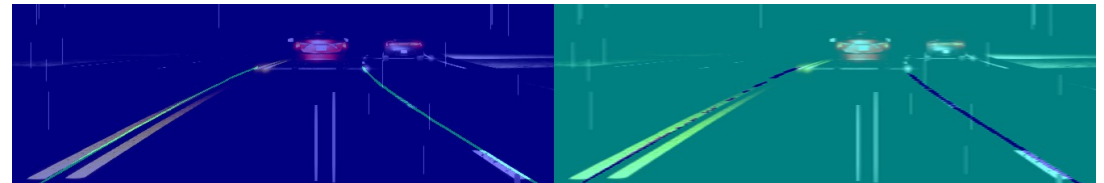
Case1



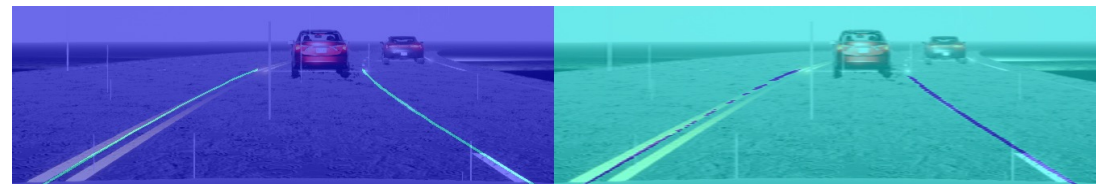
Case2



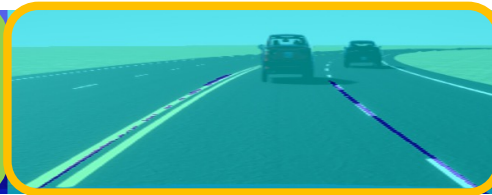
Case3



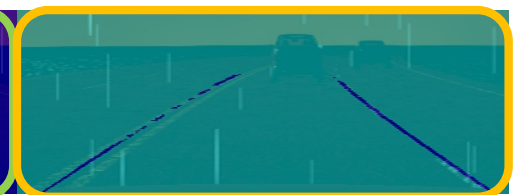
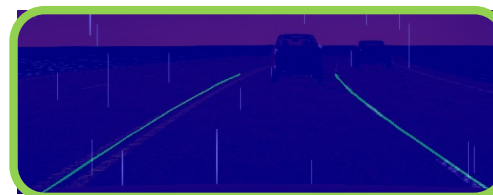
Case4



Case5



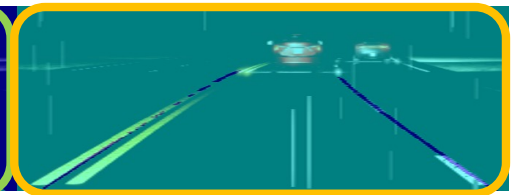
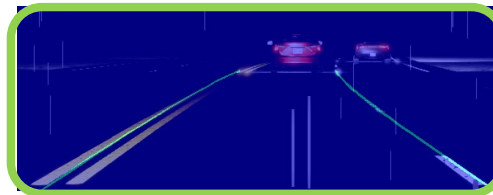
Case1



Case2



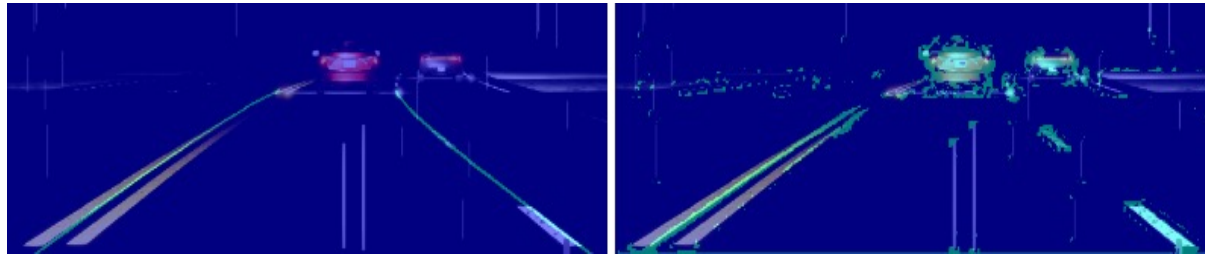
Case3



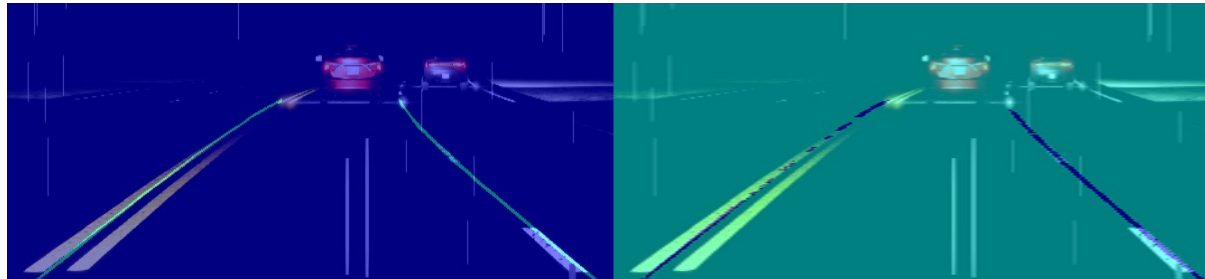
Case4



Case5

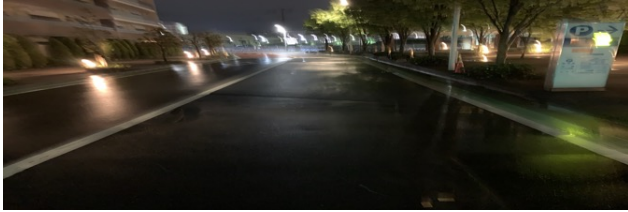


自作CNNによるCase4

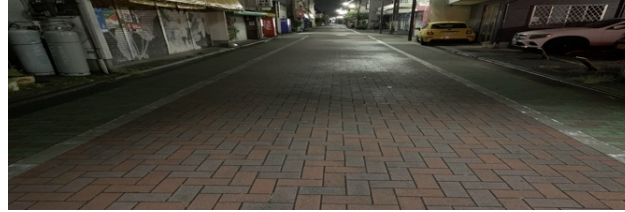


ResNetによるCase4

- CNNの方は白線(破線)単位で検出がされている
- ResNetはラベル単位での検出



Case6



Case7

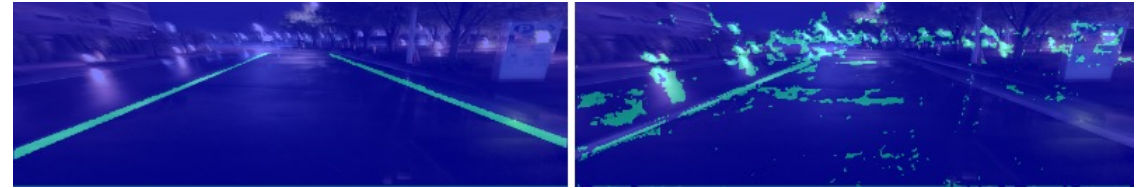


Case8

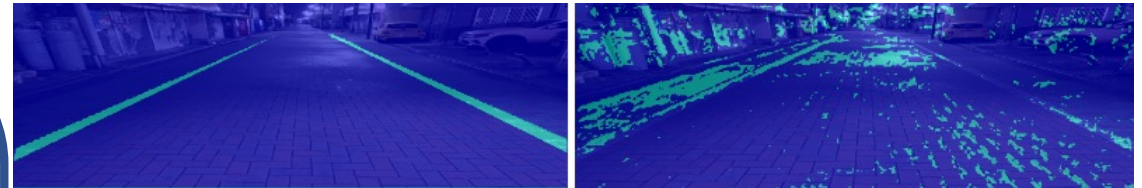
- これらの3ケースには全く適応できなかった
- 最適化手法 (SGDM, ADAM) などのパラメータ調整でも適応できなかった

- Case1～5 (デジタル画像) と似た現実画像には適応できるか
- Case6～8 (現実画像) と似たCaseのデジタル画像を学習した場合適応できるか

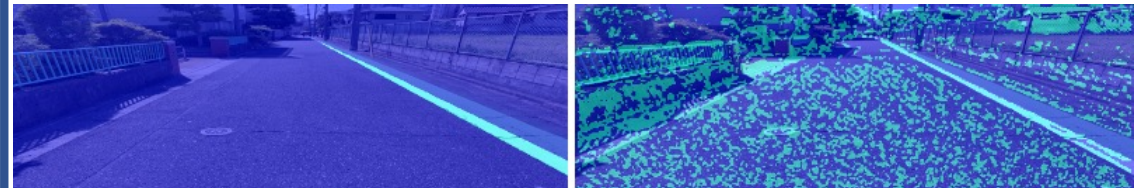
- 現実画像のラベルデータは手作業で作成



Case6



Case7



Case8

- デジタル画像の結果と同様に道路上の誤検出
- Case8について誤検出が多いが、車線の形状自体はある程度認識できている



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で学習を行い車線検出までの一連の流れを行った．

## 車線検出の評価

1. 学習データと同じデジタル画像へのテスト
2. 現実画像への適用度の確認

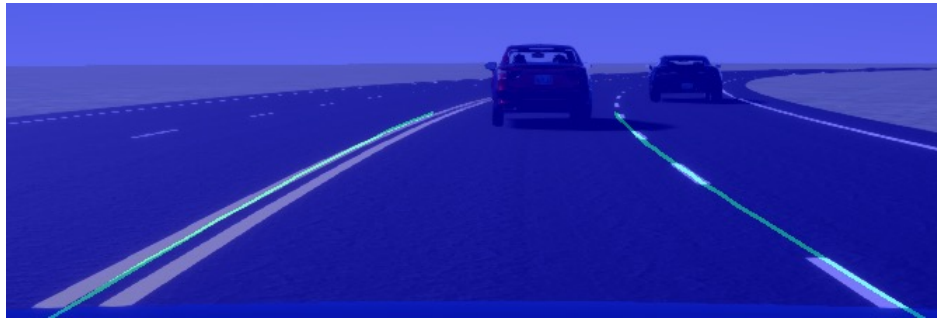
## 車線検出の結果

1. デジタル画像で学習した車線検出は  
現実の車線検出にも一定の効果がある
2. 実用レベルには達していない

## 今後の取り組み

- ラベルデータの形状
- データ数とネットワーク構造
- シナリオと道路形状の追加
- 車線認識のベンチマークテスト

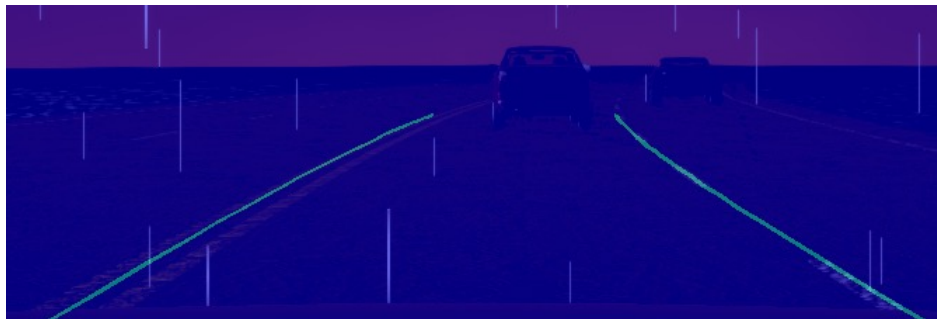




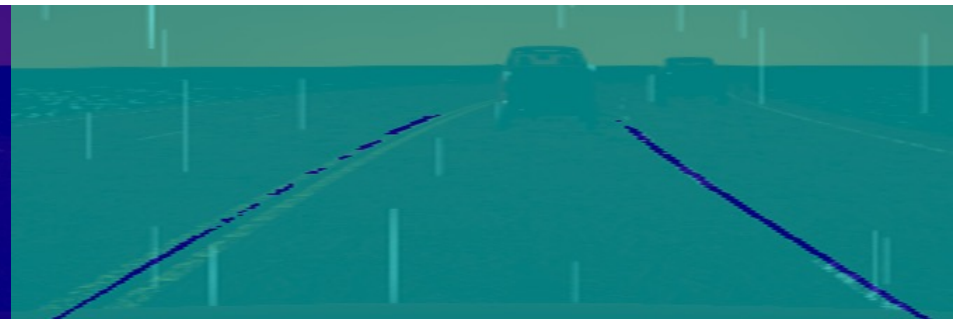
Case1



Case1



Case2



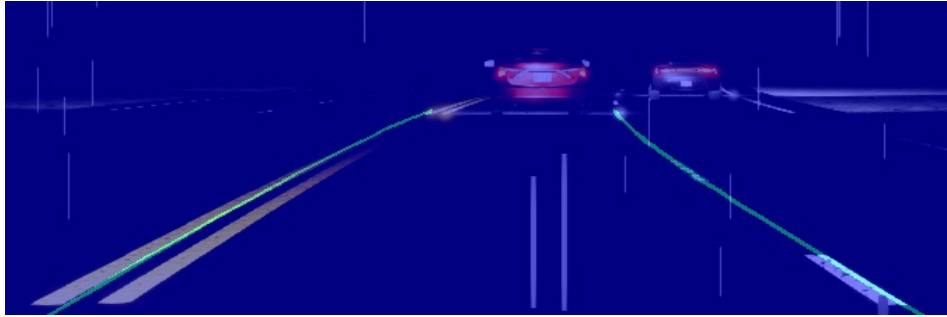
Case2



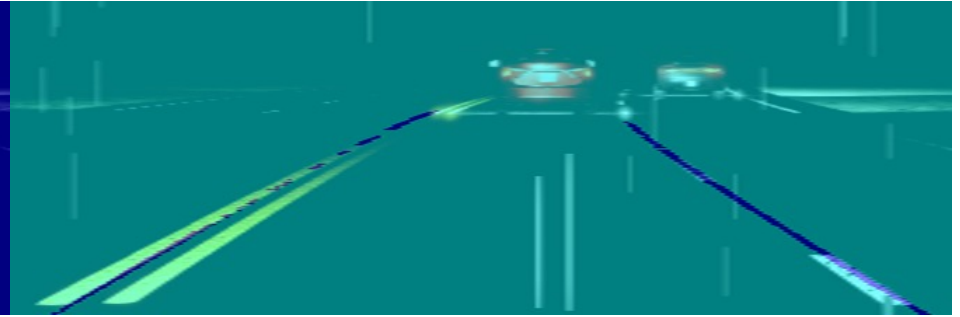
Case3



Case3



Case4



Case4



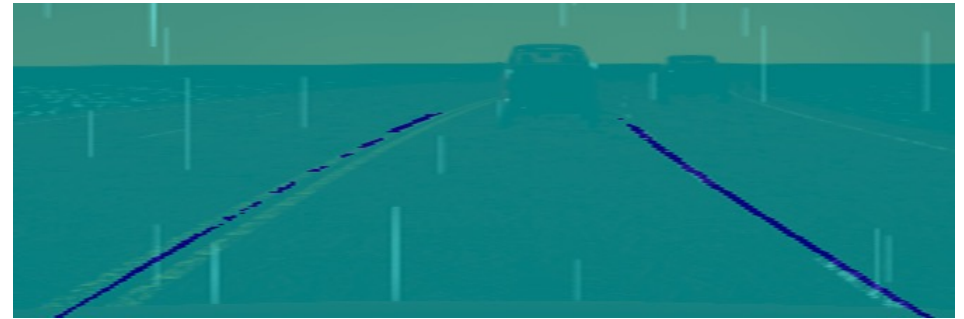
Case5



Case5



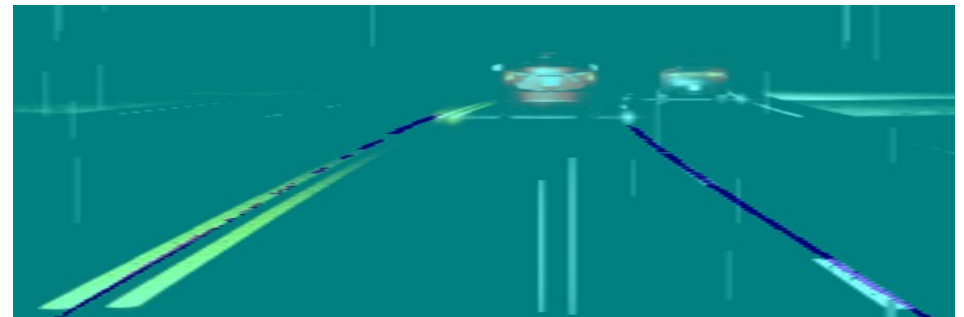
Case1



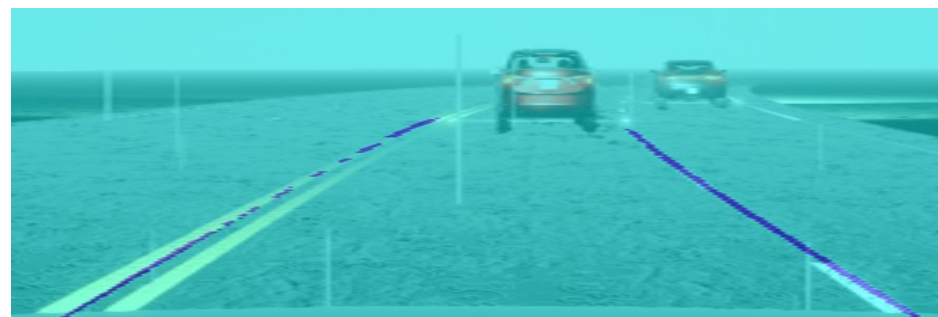
Case2



Case3



Case4



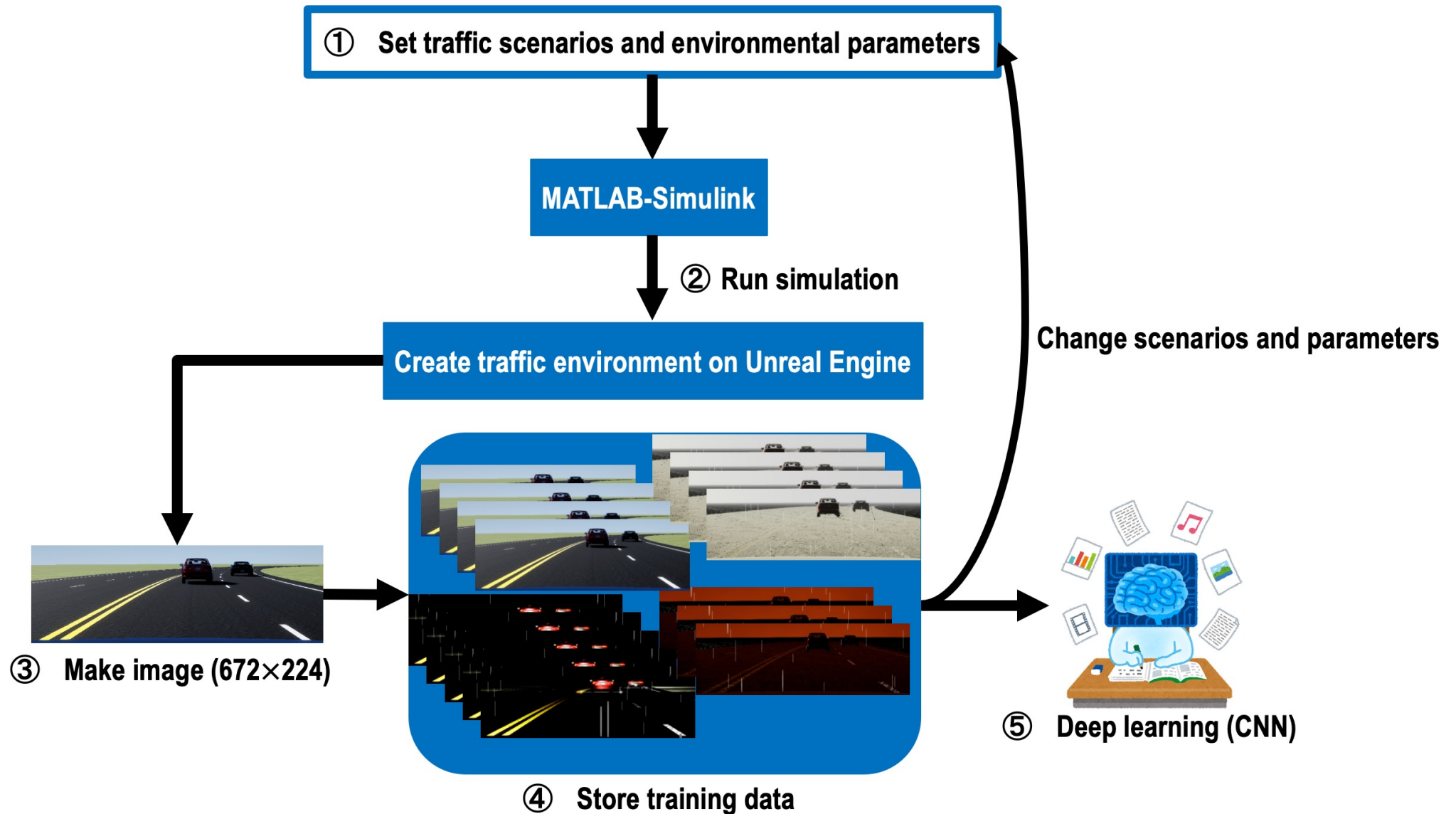
Case5

- ◆ Resnetのデジタル画像への結果  
白線単位での認識がされていない  
→ラベルの影響が大きい
- ◆ 最適化手法  
Resnetの論文<sup>[10]</sup>ではSGDMが最も良い精度であった  
→今回の実験ではADAMの方が良い精度が得られた
- ◆ 学習パラメータの調整  
学習精度の収束が早い段階で見られたためepochの調整  
→過学習が起きている可能性

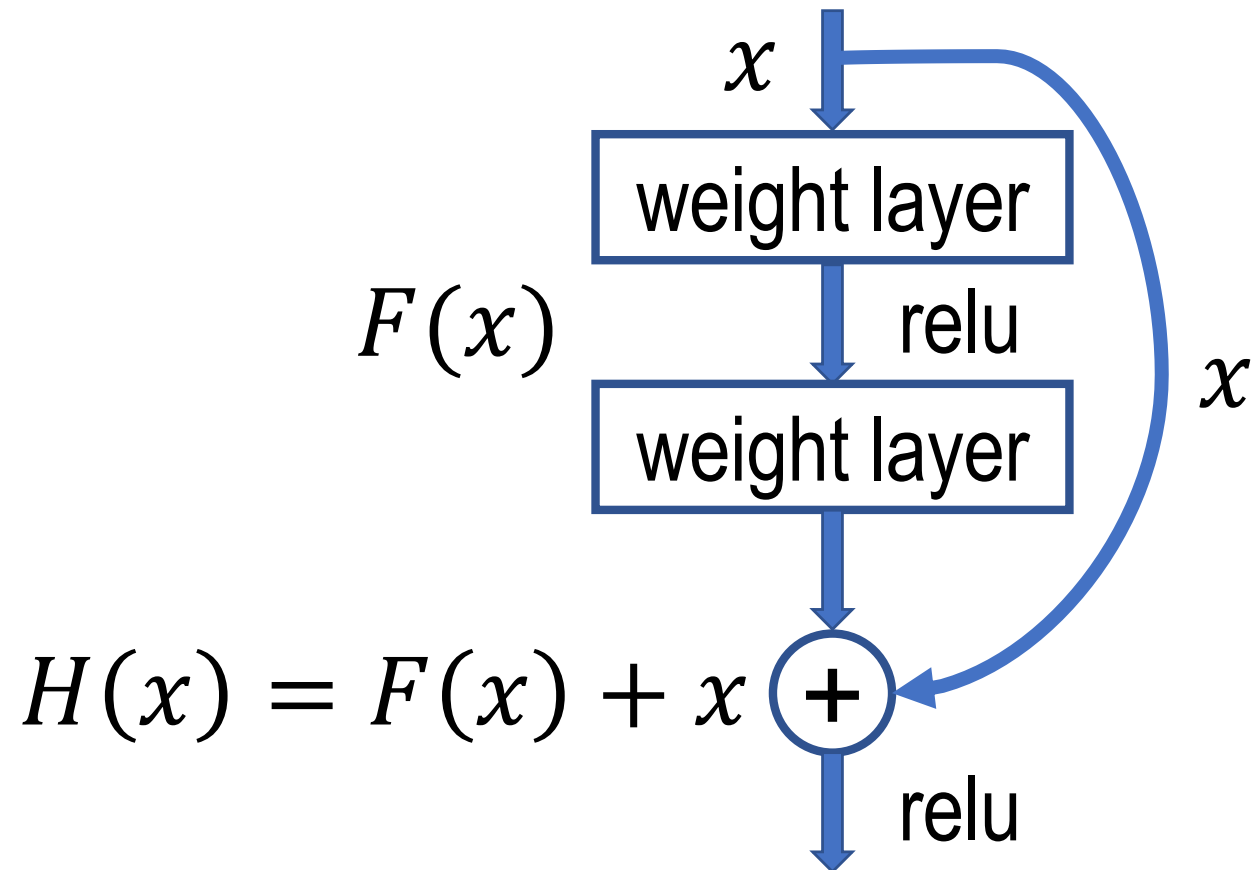
1. Zheng, T., Huang, Y., Liu, Y., Tang, W., Yang, Z., Cai, D., He, X.: CLNet: Cross Layer Refinement Network for Lane Detection, In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 898-907(2022)
2. TuSimple. Tusimple benchmark.  
<https://github.com/TuSimple/tusimple-benchmark>,  
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

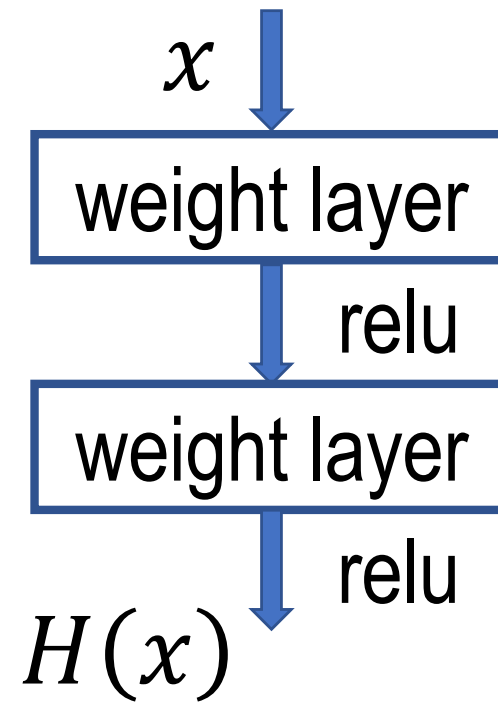
7. 中川正夫, 小林摂, 新国哲也: 自動運転車の認識性能の正確性評価とその評価手法の妥当性に関する考察, 自動車技術会論文集, vol.52, No.3, p.633-638(2021)
8. Badrinarayanan, V., Kendall, A., Cipolla, R.: SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation , arXiv preprint arXiv:1511.00561
9. Kim, J., Lee, M.: Robust lane detection 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)











## ◆ MATLAB/SimulinkとUnreal Engine

- MATLABは豊富な機械学習ライブラリを備えた統合開発環境
- Simulinkはモデルベースでの自動運転技術の開発が可能
- Unreal Engineはゲームエンジンで光環境の再現性が高い
  - Simulinkで作成したモデルをUnreal Engine上でシミュレートが可能

## ◆ 交通シナリオと環境設定

MATLABにある車線認識システムのデモ

Lane Marker Detector Test Benchに備わっている  
交通シナリオと天候や時間などの環境設定

- 発表15分，質疑応答10分の計25分
- 秋季大会は日本語，英語可
- 予稿原稿の内容に沿うもの
- 1ページ目以外に所属機関のロゴや名前を記載しない
- 1枚に一つの事柄のみをまとめ1分/枚程度
- 式や文字の説明は1枚に7~10行程度
- 記号や文字の太さは24pt以上にすると見やすい
- 図表の表題と図表中の用語は英語