2019 年度卒業

卒業論文

深層学習を用いたロバストな車線補完

Robust White Line Completion

by Using Deep Learning

所属	新潟大学工学部情報工学科・林隆史研究室
在籍番号	T16I273C
氏名	小松 耀人

概要

近年高齢者による事故が増え自動運転への期待も高まっている. しかしながら現在の研究で最も高い精度を出している自動運転技術は高精度地図を必要とし，高精度地図が整備されている道路は先進国全体でも 1%に満たない. 高精度地図を全道路に整備するのは作成に多大な人的コストがかかる点や，一般道路は更新頻度も高いという点で非現実的である. ゆえに，出発地から目的地まで人の手を一切介さない完全な自動運転の実現には高精度地図に依存せず，自車の周りの状況を把握し，その情報をもとに適切な進路をリアルタイムに選択することが不可欠である. 特に路面上の情報を画像などから計測し，自車が走行すべき車線を計算することは，乗車している人の安全を保証する意味で非常に重要である. しかし，一般道路には車線が途切れていたりかすれていたりする場所が散見され，現在の技術では線がない部分の認識はできず，それらが道路状況を把握することの障害となっている. そこで本研究では敵対的生成ネットワーク (GAN) を用いて，車線の途切れやかすれを自動補完する手法を提案する.

Abstract

In recent years, the number of accidents by the elderly has increased, and expectations for automatic driving have been increasing.

However, the automatic driving technology that has produced the highest accuracy in current research requires a high-accuracy map, and roads with high-accuracy maps are less than 1% of all developed countries. It is impractical to develop a high-precision map on all roads because it requires manual intervention and regular roads are frequently updated.

Realization of fully automatic driving without any human hand from the departure point to the destination does not rely on a high-precision map, but grasps the situation around the own vehicle and determines the appropriate course in real time based on that information. In particular, it is very important to measure the information on the road from images and calculate the lane in which the vehicle should travel, in order to guarantee the safety of the occupants. However, white lines are interrupted or faded on general roads.

However, current technology cannot recognize areas without lines, which is an obstacle to understanding road conditions. In this study, we used a hostile generation network (GAN). In this paper, we propose a method for automatically complementing the discontinuity and blurring of white lines.

目 次

第1章　はじめに

近年高齢者による事故が増えているが，地方では車がないと生活が困難な地域もあり免許を返納してもらうことが非現実的な地域も多くある．また，健常者が移動手段として使う車でも事故のリスクを少しでも減らし安全な社会を実現すべく運転支援技術の高度化や出発地から目的地まで一切人の手を介することのない完全な自動運転の実現が急がれている．

現在の研究で最も完全な自動運転に近い運転支援を実現しているのは通常のカーナビゲーションシステムに使用されている 2D の俯瞰図ではなく，数センチメートル以内の誤差しかない超高精度の 3D 地図にさらに道路上の構造物，車線情報，路面情報などの「静的情報」，道路工事やイベントなどによる交通規制などの「準静的情報」，信号の現示情報などの「動的情報」，観測時点の渋滞状況などの「準動的情報」などを付加したものでこの地図が整備されているのは先進国全体で 1%未満である．このような地図の整備状況が未だに狭い理由としては，そもそもの作成に人力で CAD を用いた細かい調整が必要なことに加え，一般道路は更新の頻度も高く新しい道路が頻繁に増えていくことや地点ごとの車線や路面の整備状況の差異が非常に大きく手動で逐一更新していくのは非現実的である．様々な機関が自動的に高精度地図を整備する技術を提案しているが，未だに一般道路を自動的に高精度な地図に起こすまでには至っていない．[1][2]

このような背景から自車の周囲の車線や路面の情報を観測し，活用していくことが完全な自動運転の実現には必要不可欠である．また，仮に高精度地図が普及したとしても搭乗者の安全を守り正確な自動運転を行うために自車からの周辺観測は非常に重要である．

現在製品化している運転支援技術は先行車がいてかつ高速道路上で限られた速度帯でのみなど非常に限られた状況でしか作動せず，完全な自動運転には程遠い．限られた状況でしか作動しない原因の一つに一般道路は高速道路などの高規格幹線道路とは異なり道路整備の頻度も低く舗装や塗装の基準も異なるので場所によって車線や横断歩道などの路面上の情報が不完全になっているものも多くこれらを含む路面情報を認識するには前処理として本来必要であるのに欠落している情報を補完する必要がある．[3]

以上の点を踏まえて本研究では深層学習を用いた画像補完技術を用いて本来必要であるのに車線が消えたりかすれたりしている部分を補完することを目的とする．画像補完には Generative Adversarial Network(GAN) を用い，ニューラルネットワークを用いて車線があるべき領域を重点

的に学習し，補完を試みる．

第 2 章では機械学習の一つであるニューラルネットを用いた学習法の全体概要を説明した後，分類モデルでよく用いられる CNN と，生成モデルで用いられる GAN について細かく説明する. 第 3 章では実験手法として利用した Globally and Locally Consistent Image Completion(GLCIC) について説明する．第 4 章では提案手法について述べる．第 5 章では提案手法の性能評価について述べる．はじめに学習と検証に用いたデータセットについて述べ，次に実際にネットワークを構築し実際に学習した結果を示す．第 6 章では実験結果や学習について考察を述べる．

第2章　ニューラルネットを用いた機械学習

本章では機械学習の中の一つであるニューラルネットについて基本的なことから説明し，次にCNN，GANの仕組みについて詳しく説明する．

2.1　深層学習

深層学習は，図 **2.1** に示すような生物の脳の神経細胞をモデル化したニューロンを基にして，図 **2.2** のようにニューロンを多層に結合したモデル (ニューラルネットワーク) を用いる学習法である．個々のニューロン間の結合には重み $\boldsymbol{w}$ というパラメータが与えられており，$\boldsymbol{w}$ が更新されていく (学習する) ことで，問題にあった最適解を導く．この重み $\boldsymbol{w}$ の学習法として，誤差逆伝播法を用いる．誤差逆伝播法は教師信号値 $\boldsymbol{t}$ と出力結果値 $\boldsymbol{h(x)}$ の誤差の大きさを表す損失関数 E を定義し，E に対し各層の $\boldsymbol{w}$ の微分係数 (勾配) を求めることで $\boldsymbol{w}$ を更新していく．

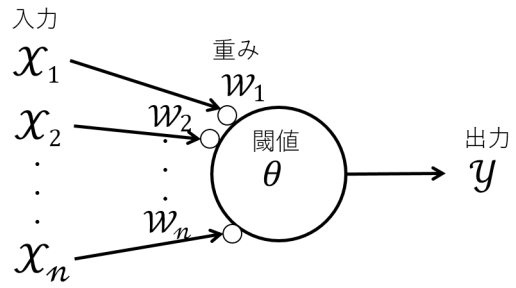

図 2.1: ニューロンモデル

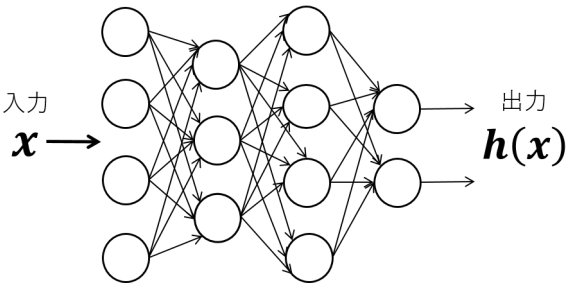

図 2.2: 多層パーセプトロン

2.2　損失関数と重み最適化

2.2.1　損失関数

損失関数 E は様々な種類があり，一般的によく使われるものとして式 (2.1)〜式 (2.3) の，交差エントロピー誤差，平均 2 乗誤差，平均絶対誤差などがある．

$$E = -\sum_{k}^{n} t_k log(h_k(\boldsymbol{x})) \tag{2.1}$$

$$E = \frac{1}{n}\sum_{k}^{n} (t_k - h_k(\boldsymbol{x}))^2 \tag{2.2}$$

$$E = \frac{1}{n}\sum_{k}^{n} |t_k - h_k(\boldsymbol{x})| \tag{2.3}$$

2.2.2　重み最適化

$\boldsymbol{w}$ を更新する手法は様々あるが，基本となっているものは式 (2.4) の勾配降下法であり，損失関数の勾配が減少する方向に学習率 η を乗算した値を加えていくことで $\boldsymbol{w}$ の最適値を見つけていく．

$$\boldsymbol{w_{t+1}} = \boldsymbol{w_t} - \eta\frac{\partial E}{\partial \boldsymbol{w}} \tag{2.4}$$

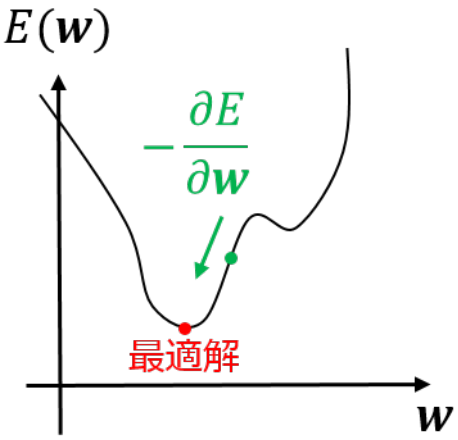

図 2.3: 勾配降下法

2.2.3 Adam

Adam は式 (2.4) を派生させた $\boldsymbol{w}$ の更新方法で現在よく用いられている最適化アルゴリズムである [4]．2015 年に Diedeik P. Kingma らが提唱した手法であり，式 (2.5) のように学習ステップごとに過去の勾配の値から勾配の重みつき平均と重みつき分散を推定している．これにより，更新が多い重みの学習率を低く，更新が少ない重みの学習率を高くするように設定され，学習の収束が早くなることが期待できる．式 (2.6), 式 (2.7) における β_1,β_2 はハイパーパラメータを表し，実装する側が指定する値である．

$$\begin{aligned} \boldsymbol{w_{t+1}} &= \boldsymbol{w_t} - \eta \frac{\hat{\boldsymbol{m}}}{\sqrt{\hat{\boldsymbol{v}}} + \epsilon} \\ \hat{\boldsymbol{m}} &= \frac{\boldsymbol{m_{t+1}}}{1 - \beta_1^t} \\ \hat{\boldsymbol{v}} &= \frac{\boldsymbol{m_{t+1}}}{1 - \beta_2^t} \end{aligned} \tag{2.5}$$

$$\boldsymbol{m_{t+1}} = \beta_1 \boldsymbol{m_t} + (1 - \beta_1) \frac{\partial E}{\partial \boldsymbol{w_t}} = (1 - \beta_1) \sum_{i=1}^{t} \beta_1^{t-i} \boldsymbol{m_i} \tag{2.6}$$

$$\boldsymbol{v_{t+1}} = \beta_2 \boldsymbol{v_t} + (1 - \beta_2) (\frac{\partial E}{\partial \boldsymbol{w_t}})^2 = (1 - \beta_2) \sum_{i=1}^{t} \beta_2^{t-i} \boldsymbol{v_i} \tag{2.7}$$

2.3　畳み込みニューラルネットワーク

畳み込みニューラルネットワーク (CNN : Convolutional Neural Network) は，人間の視覚野の神経細胞の二つの働きである「画像の濃淡パターンを検出する (特徴抽出)」，及び「物体の位置が変動しても同一の物体であるとみなす (位置ズレの考慮)」を組み合わせたものとなっており，画像分野において高い評価を持つニューラルネットワークとなっている [5]．また，入力データのパターンをうまく学習できるという点で，画像だけでなくを様々な問題設定で CNN が広く用いられ，高い精度を出している．

図 2.4 は画像分類問題を例にとった CNN のモデルを示している．入力画像は特徴抽出部で特徴が抽出され，その特徴をもとに識別部でパターン分類を行う．特徴抽出部では数層の畳み込み層とプーリング層から構成され，識別部は全結合層から構成されている．

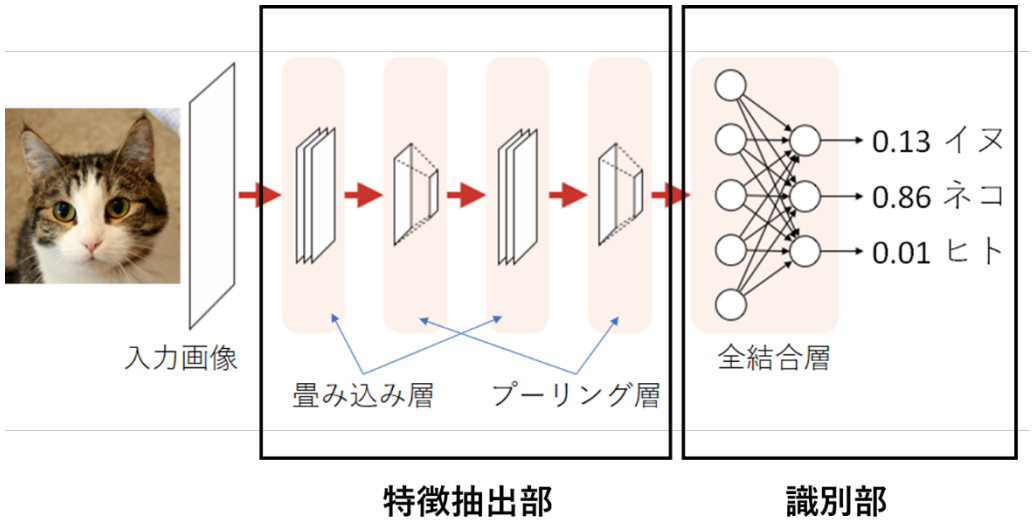

図 2.4: 畳み込みニューラルネットワーク

2.3.1　畳み込み層

畳み込み層は画像の濃淡パターンを検出するための層に相当する．図 **2.5** に示すように，入力画像に対し各ピクセル値にフィルタを適用し，フィルタをスライドさせながら画像を圧縮し，特徴マップを作成する．学習時にはフィルタの値が更新される．

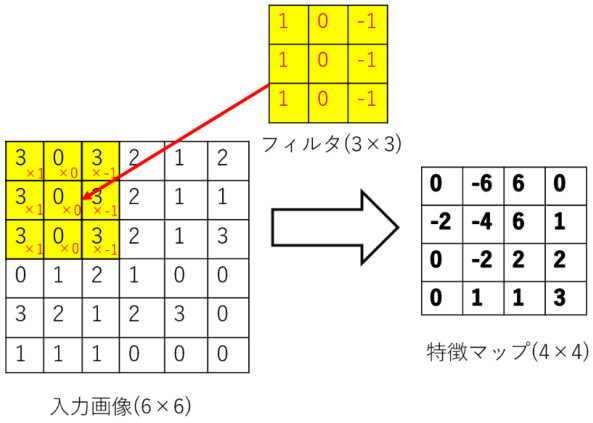

図 2.5: 畳み込み

2.3.2　プーリング層

プーリング層は位置に対する感度を低くする代わりに，位置変化に対する認識能力を上げるための層に相当する．畳み込み層で得られた特徴マップに対しさらに圧縮をかけることで位置ズレの変化に対応する仕組みとなっている．圧縮のかけ方としては，最大プーリングと平均プーリングがある．図 **2.6** に示すのは最大プーリング (max pooling) であり，特徴マップの小領域の中から最大のピクセル値を得る操作となっている．対して平均プーリング (average pooling) は小領域の中の値を平均した値を得る操作となる．プーリング層においては学習時に更新されるパラメータは存在しない．

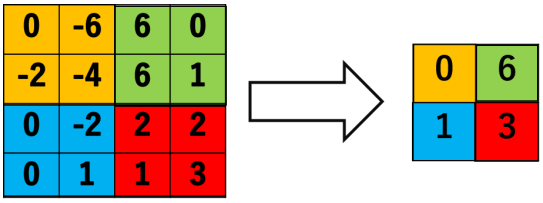

図 2.6: 最大プーリング

2.3.3　全結合層と活性化関数

全結合層は，特徴抽出部から得られた特徴マップの値を入力とし，図 **2.2** のような多層パーセプトロンのニューラルネットワークによりパターン分類が行われる層である．最後のニューロンの出力数は問題設定に応じて変更される．分類問題の場合は分類したいクラス数であったり，画像生成問題の場合は画像のピクセルサイズ分の出力を持ったりなどをする．

各層のニューロンの出力値は活性化関数が通された後の値となっている．これは，ニューロンの出力が線形であるため，多層パーセプトロンと等価な 1 層のパーセプトロンの対が必ず存在してしまうことを避けるためである．活性化関数には，ランプ関数 (ReLU)，ソフトマックス関数，シグモイド関数などといったものが存在する．

Relu 関数

中間層のニューロンの出力を非線形にするためによく使用されている関数である．非負の値を出力する．

$$f(x) = max(0, x) \tag{2.8}$$

ソフトマックス関数

ニューロンの出力を確率として扱う場合に使用される関数である．分類問題において最終層のニューロンの活性化関数として使用され，式 (2.9) で表される．a_k は k 番目のニューロンの出力値であり，n は最終層のニューロンの出力数である．

$$f_k(a_k) = \frac{e^{a_k}}{\sum_{i=1}^{n} e^{a_i}} \tag{2.9}$$

シグモイド関数

出力値を 0〜1 に収める関数である．ニューロンの出力値を非線形したいときや，確率とみなしてマルチクラス分類をする際にも使用されている．

$$f(x) = \frac{1}{1 + e^{-x}} \tag{2.10}$$

2.4　敵対生成ネットワーク

敵対生成ネットワーク (GAN : Generative Adversarial Network) は，学習データと似たような新しいデータを生成する生成モデルの一種であり，言い換えれば生成データの分布を学習データの分布に近づけていくように学習するモデルである．GAN の構造を図 **2.7** に示す．GAN は図 **2.7** に示すように，生成器 (Generator) と識別器 (Discriminator) から構成され，Generator はランダムノイズからオリジナルと似たデータを生成し，Discriminator は入力されるデータが Generator によって作られたデータか，それともオリジナルのデータかの識別を行う．これら二つのモデルが互いを見抜く・騙すように学習するため，十分に学習が進むと Generatr はオリジナルのデータと見分けがつかないようなデータを生成するようになる．

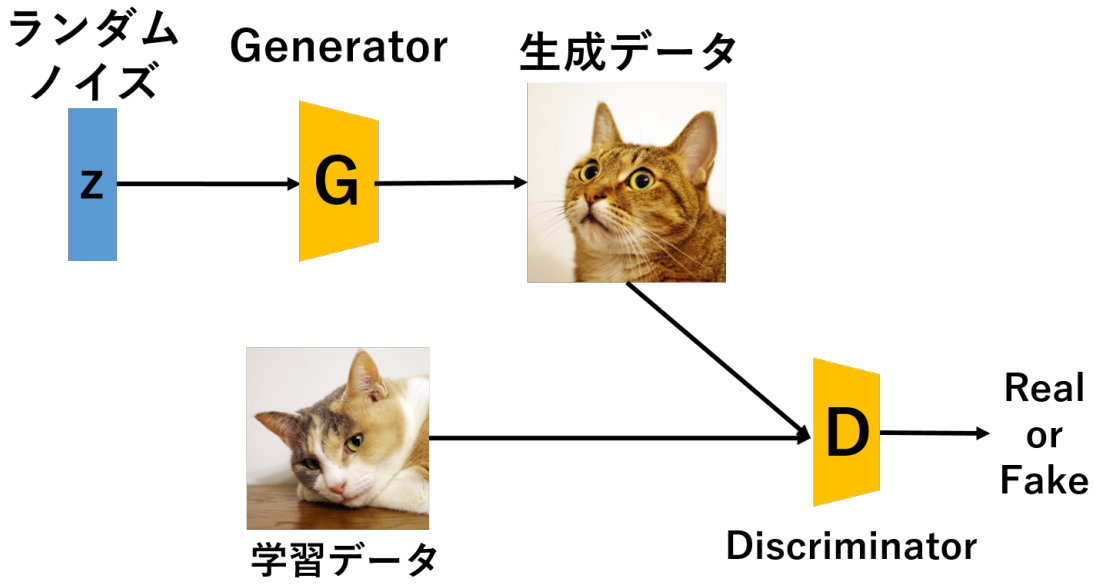

図 2.7: 敵対生成ネットワーク (GAN)

2.4.1 GAN の損失関数

具体的に評価関数を導入して学習を行う場合，式 (2.11) に関して，Discriminator に対して最大化，Generator に対して最小化するミニマックスゲームを考えればよい [6].

$$min_G max_D V(G,D) = \mathbb{E}_{\boldsymbol{x}\sim p_r(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[1 - \log D(G(\boldsymbol{z}))] \tag{2.11}$$

式 (2.11) において，$\boldsymbol{x}$ はオリジナルデータ，$p_r(\boldsymbol{x})$ はオリジナルデータの確率分布，$\boldsymbol{z}$ はランダムノイズ，$p_z(\boldsymbol{z})$ はランダムノイズの確率分布を示す．この時，D がオリジナルのデータを正しく判定できれば $\log D(\boldsymbol{x})$ が大きくなり，D が G の生成データをオリジナルのデータと誤って判定すると $\log(1 - D(G(\boldsymbol{z})))$ が小さくなる．

2.4.2 GAN の最適解

式 (2.11) において，$\boldsymbol{z} \sim p_z(\boldsymbol{z})$ から G が生成するデータの分布を $p_g(\boldsymbol{x})$ とし，$V(G,D)$ を書き直すと，

$$V(G,D) = \int_{\boldsymbol{x}} \{p_r(\boldsymbol{x})\log D(\boldsymbol{x}) + p_g(\boldsymbol{x})\log(1 - D(\boldsymbol{x}))\}\boldsymbol{dx} \tag{2.12}$$

となる．Discriminator の最適解は $V(G,D)$ を最大化することであるため，積分の中身の最大化をすればよい．よって中身を $D(\boldsymbol{x})$ に関して微分し，その時の導関数が 0 になる $D(\boldsymbol{x})$ を $D^*(\boldsymbol{x})$ とすると，式 (2.13) となる．

$$D^*(\boldsymbol{x}) = \frac{p_r(\boldsymbol{x})}{p_r(\boldsymbol{x}) + p_g(\boldsymbol{x})} \tag{2.13}$$

またこのとき，Generator の最適化を考える．式 (2.13) の右辺を式 (2.12) に代入し式を整理すると，式 (2.14) となる．

$$V(G,D) = 2D_{JS}(p_r||p_g) - \log 4 \tag{2.14}$$

式 (2.14) を最小化することは，Jensen-Shannon ダイバージェンスの最適化と等しく，$p_g = p_r$ のときに最小値 $-\log 4$ をとる．

第3章　先行研究

本章では本研究で利用した先行研究である Globally and Locally Consistent Image Completion(GLCIC)[9] について紹介する.

3.1　概要

GLCIC は画像を局所的にも大域的にも整合性の取れた画像を生成する画像補完の手法で，従来の GAN と異なり 2 つの Discriminator を使用することによって局所性と大域性を両立する.
以下の 2 つの点で従来手法と異なる

- 拡張畳込み層
- 補完ネットワークと 2 つの識別ネットワーク

3.2　拡張畳み込み層

拡張畳み込み層では η を膨張係数として各層が入力として使用できる領域をパラメータを増やすことなく広げる.
入出力の式は (3.2),(3.2) のとおり

$$y_{u,v} = \sigma(b + \sum_{j=-k'_h}^{k'_h} \sum_{j=-k'_w}^{k'_w} W_{k'_h+i,k'_w+j} x_{u+\eta i,v+\eta j}) \tag{3.1}$$

$$k'_h = \frac{k_h - 1}{2}, \quad k'_w = \frac{k_w - 1}{2} \tag{3.2}$$

それぞれのパラメータについては**表 3.1** に示す

表 3.1: 拡張畳み込み層の各パラメータ

σ	活性化関数
$x_{u,v}$	入力
$y_{u,v}$	出力
$W_{s,t}$	重み
k_w, k_h	カーネルの幅
b	バイアス
η	膨張係数

3.3 補完ネットワーク

補完ネットワークの層構造を次の**図 3.1**, **表 3.2** に示す.
補完ネットワークは画像の中の任意の $\frac{1}{4}$ の領域をマスクし，そこをより正解画像に近く補完できるように学習する．**表 3.2** では 128 × 128 の画像が入力された際の層構造を示す.

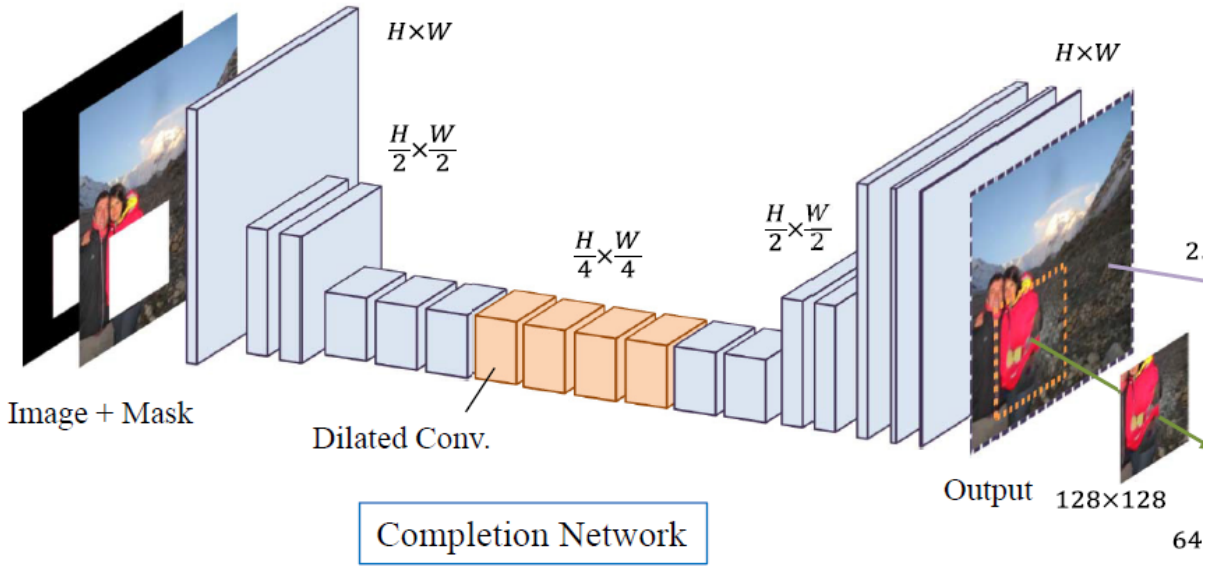

図 3.1: 補完ネットワーク

表 3.2: 補完ネットワークの層

Type	Kernel	Dilation(η)	Stride	Outputs
conv.	5 × 5	1	1 × 1	64
conv.	3 × 3	1	2 × 2	128
conv.	3 × 3	1	1 × 1	128
conv.	3 × 3	1	2 × 2	256
conv.	3 × 3	1	1 × 1	256
conv.	3 × 3	1	1 × 1	256
dilated conv.	3 × 3	2	1 × 1	256
dilated conv.	3 × 3	4	1 × 1	256
dilated conv.	3 × 3	8	1 × 1	256
dilated conv.	3 × 3	16	1 × 1	256
conv.	3 × 3	1	1 × 1	256
conv.	3 × 3	1	1 × 1	256
deconv.	4 × 4	1	1/2 × 1/2	128
conv.	3 × 3	1	1 × 1	128
deconv.	4 × 4	1	1/2 × 1/2	64
conv.	3 × 3	1	1 × 1	32
deconv.	3 × 3	1	1 × 1	3

3.4 識別ネットワーク

識別ネットワークは入力された画像が補完ネットワークによって補完された偽物か本物かを識別できるように学習する.

識別ネットワークは更に大域識別ネットワークと局所識別ネットワークに別れてそれぞれ画像全体での自然さと補完領域のみの自然さに関して識別する. **図 3.2**, **表 3.3**, **表 3.4** に識別ネットワークのそれぞれの層構造を示す.

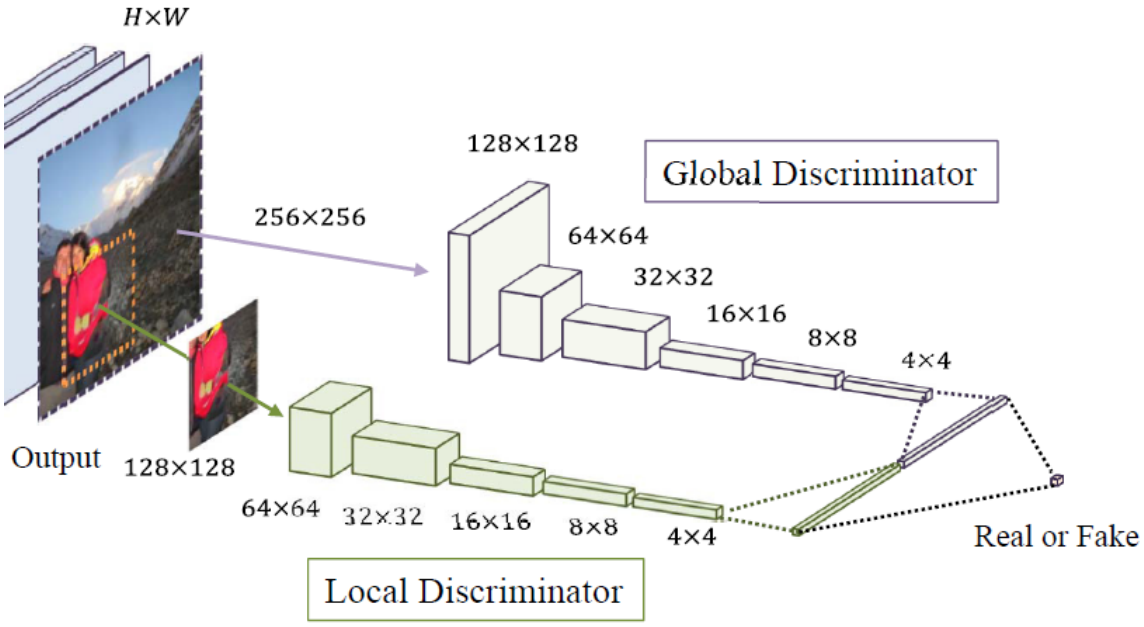

図 3.2: 識別ネットワーク

表 3.3: 大域識別ネットワークの層

Type	Kernel	Stride	Outputs
Conv.	5 × 5	2 × 2	64
Conv.	5 × 5	2 × 2	128
Conv.	5 × 5	2 × 2	256
Conv.	5 × 5	2 × 2	512
Conv.	5 × 5	2 × 2	512
FC	-	-	1024

表 3.4: 局所識別ネットワークの層

Type	Kernel	Stride	Outputs
Conv.	5 × 5	2 × 2	64
Conv.	5 × 5	2 × 2	128
Conv.	5 × 5	2 × 2	256
Conv.	5 × 5	2 × 2	512
Conv.	5 × 5	2 × 2	512
Conv.	5 × 5	2 × 2	512
FC	-	-	1024

第4章　提案手法

前章で示したGLCICは3つのネットワークを組み合わせることによって非常に多くの画像において自然な補完をすることができるが，学習結果が収束し補完できるモデルができるまでに非常に長い時間がかかってしまうという欠点がある．

景色に関して77%の人が自然であると判断する画像を作成するまでにK80GPUを4台用いて約二ヶ月学習を要したという結果も出ており，画像全体で自然さを保ちつつマスク部を生成するのには非常にリソースが必要である

そこで提案手法では**表 4.1**に示すように，この学習の収束までの時間の長さが補完対象の広さ由来だと考え，更に**図 4.1**, **図 4.2**に示すように学習の対象をドライブレコーダーによって撮影された道路のデータのみに絞り車線が出現しやすい部分に重点的にマスクし，かつ同一ネットワークでCelebAデータセットですでに学習されたモデルを流用し転移学習を行うことによって学習時間の削減を試みる．

表 4.1: 学習にかかったリソースと使用したデータセットの大きさの比較

	先行研究	提案手法
データセット名	places2	waymo
学習画像の枚数	1,800万	1.1万
学習にかかったリソース	K80 4台× 2ヶ月	GTX1080Ti × 1週間

図 4.1: 車線部分をマスクした例 (1)

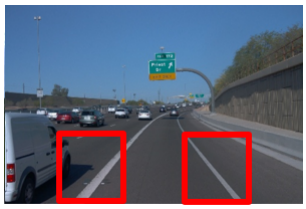

図 4.2: 車線部分をマスクした例 (2)

第5章　提案手法の性能評価

提案手法の性能評価と使用したデータについて述べる．

5.1　評価実験の概要

ドライブレコーダーによって撮影されたデータを教師データとしてネットワークに与え，GLCICではによってマスクした部分を補完した際の精度を比較する．

5.2　評価実験に用いた環境・ツール

5.2.1　実験環境

本実験は表 **5.1** に示す環境で行った

表 5.1: 実験環境

OS	Windows 10 Pro(64bit)
CPU	Corei7-8700K(3.7GHz)
RAM	32GB
GPU	NVIDIA(R) GeForce GTX 1080 Ti 11GB GDDR5X

5.2.2　使用したプログラミング言語とライブラリ

実装のために表 **5.2** に示すツールを用いた

表 5.2: 使用した言語・ライブラリ

使用言語・ライブラリ	バージョン
Python	3.6.8
numpy	1.17.4
Keras	2.3.1
TensorFlow	1.15.0
opencv-python	4.1.2.30

5.3 データセット

本研究では waymo(https://waymo.com/open/) にて公開されている Waymo-dataset[7][8] を使用する.

Waymo-dataset はアメリカ国内の San francisco,Phoenix,Mountrain View の 3 箇所で観測されたデータで以下の図 5.1 に示すようなセンサ類で観測されたデータを内包している.

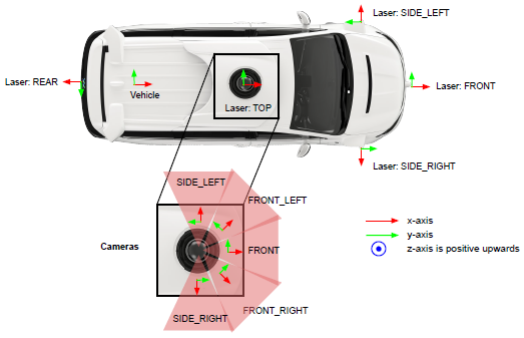

図 5.1: Waymo データセットにおけるセンサーの配置

図 5.1 で示したセンサ類とカメラで撮影した画像のうち，本実験ではカメラによって撮影された FRONT の画像のみを使用し，更にその中で車線がはっきりしているデータのみを抽出し教師データとして与えて学習を行った．もともとの waymo データセットに入っていた FRONT カメラで撮影した画像は 20 万枚だったが，その中から車線がはっきりしている約 1.1 万枚のみを抽出し使用した.

図 5.2-図 5.5 にデータセットの中で教師データとして使用したものと使用していないものの一例を示す.

図 5.2: データセットとして使用した画像の例 (1)

図 5.3: データセットとして使用した画像の例 (2)

図 5.4: データセットとして使用していない画像の例 (1)

図 5.5: データセットとして使用していない画像の例 (2)

5.4 実験結果

先行研究 [9] では，補完ネットワークと判別ネットワークをそれぞれで学習させた後に相互学習を行っているが，転移学習で予め学習済みのモデルを用いたことや GAN は一般に補完側より判別側が強くなりやすく補完側の Loss が発散してしまう場合が多いので，提案手法では補完ネットワークのみの学習を行った後相互学習を行うという流れで実験を行った．

5.4.1 補完ネットワークの評価

補完ネットワークの loss について**図 5.6** と**図 5.7** に示す．
学習が進むに従って loss の減少率は落ちていっているが，**図 5.7** に示されるように loss が発散することなく順調に学習が進んでいることが見て取れる．

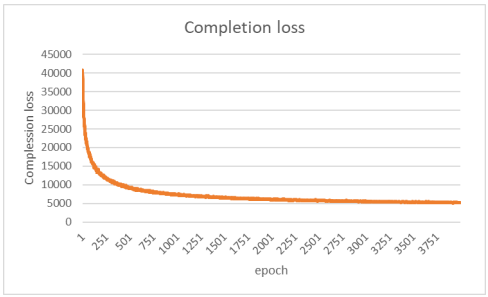

図 5.6: 補完ネットワークの loss

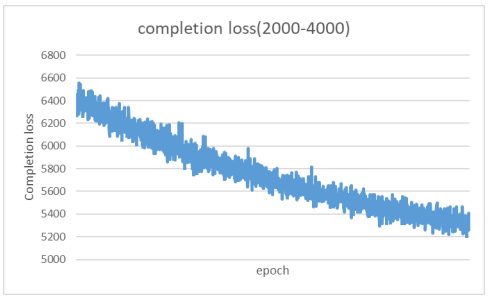

図 5.7: 補完ネットワークの loss(2000 から 4000epoch に関して拡大したもの)

5.4.2　判別ネットワークの評価

判別ネットワークの loss について**図 5.8** と**図 5.9** に示す.
どちらのグラフからも loss が収束せず多少減衰しながらもかなり epoch ごとの値に差異が有ることを示している.

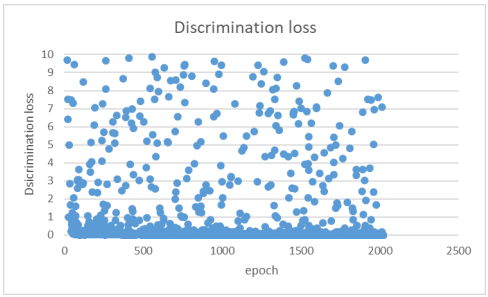

図 5.8: 判別ネットワークの loss

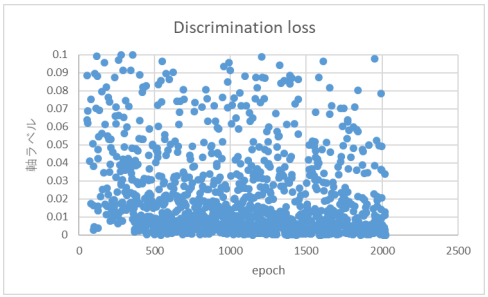

図 5.9: 判別ネットワークの loss(値域を 0.0-0.1 に絞ったもの)

5.4.3　モデルによって車線を補完した結果

車線の復元

すでにつながっていた部分をマスクしてモデルで予測した結果を**図 5.10 図 5.11** に示す.
補完前後を比較するとマスク部分に関してもともとつながっていた直線が確かに復元されていることが確認され，更に補完前は影で黒く潰れ周辺のアスファルトとの差異が小さかった部分に関してよりはっきりと差異が見えるようになったのがわかる.

図 5.10: 補完前

図 5.11: 補完後

車線の補完

図 5.12, **図 5.13** に途切れている車線をモデルで補完した結果を示す.
図 5.10, **図 5.11** に見られるほどの鮮明さはないが，確かに車線が伸びていて本来あるべき位置に車線が補完されているのが見て取れる.

図 5.12: 補完前

図 5.13: 補完後

第6章　考察

一般に GAN の学習においては判別側が強くなり補完側が意味のない画像を生成するようになることが多いが [10]，**図 5.6**, **図 5.7** に示すとおり補完側の loss は発散せず少しづつではあるが下がっているのでうまく補完側の学習ができたと言える．

しかしながら loss が下がりきって安定したわけではなくまだ下がる余地があるので，学習のドメインを絞って更に転移学習を行ったとはいえまだまだ学習の収束には時間がかかる見込みであり，1 週間という時間では学習時間としては不十分であったので今後更に学習をすすめ，より正確に車線を補完できるようにしたい．

学習後の生成画像については**図 5.10**, **図 5.11** に示すようにもともと直線の車線があった部分については正しく復元することができており，学習が成功していることを示している．加えて赤線で囲まれた補完領域に注目すると，補完前より影によって車線が隠れている部分が鮮明に補完されておりこの点でも車線を実際に認識する際に貢献すると考えられる．

実際に途切れている画像をモデルに与えてマスクした場合だが，**図 5.12**, **図 5.13** に示すように，たしかに車線が延長されかつ本来車線があるべき部分に補完されているのが見える．もともと直線があった部分を復元したのと比較すると若干不鮮明では有るが，延長できているので正しく学習できている．今回提案手法ではマスクの範囲を車線が出現する確率が高い領域に限定したが，本来の GLCIC では画像内の任意の領域をマスクするのでその場合アスファルトの部分をマスクする確率が必然的に高くなり今回のような結果は出ず，本来車線がある部分もアスファルト色に補完していたのではないかと考えられる．

第7章　まとめ

高精度地図を用いずに行う自動運転において非常に重要な役割を果たす車線認識で，誤認識の元になる車線のカスレや途切れを車線が直線の場合に補完することができた．
しかし，まだ破線部分についての学習や補完の精度の向上など課題は有るので，今後より広い範囲で正確な保管ができるように，ゆくゆくは他の手法と組み合わせて車線の認識精度を向上できるように更に研究を進めていきたい．

参考文献

[1] TRI-AD
https://www.tri-ad.global/jp/news/20190425 （2020 年 01 月 27 日アクセス）

[2] 自動運転 LAB
https://jidounten-lab.com/u_autonomous-map-company （2020 年 01 月 27 日アクセス）

[3] 第 6 章 道路舗装に関する設計基準
http://www.mlit.go.jp/sogoseisaku/inter/keizai/gijyutu/pdf/road_design_j2.pdf （2020 年 01 月 27 日アクセス）

[4] Diederik P. Kingma, Jimmy Lei Ba, ”ADAM: A Method for Stochastic Optimizer, ”

[5] 畳み込みニューラルネットワーク
https://ml4a.github.io/ml4a/jp/convnets/（2020 年 01 月 27 日アクセス）

[6] ”GAN と損失関数の計算についてまとめた，”
https://qiita.com/kzkadc/items/f49718dc8aedbe8a1bee（2020 年 01 月 27 日アクセス）

[7] Waymo
https://waymo.com/open/ （2020 年 01 月 30 日アクセス）

[8] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla,Aurelien Chouard, Vijaysai Patnaik, Paul Tsui,James Guo, Yin Zhou,Yuning Chai,Benjamin Caine, Vijay Vasudevan, Wei Han, Jiquan Ngiam,Hang Zhao, Aleksei Timofeev, Scott Ettinger, Maxim Krivokon, Amy Gao, Aditya Joshi, Yu Zhang, Jonathon Shlens, Zhifeng Chen, and Dragomir Anguelov,”Scalability in Perception for Autonomous Driving: Waymo Open Dataset”,18 Dec 2019

[9] SATOSHI IIZUKA,EDGAR SIMO-SERRA,HIROSHI ISHIKAWA,”Globally and Locally Consistent Image Completion”,July 2017

[10] Tim Salimans and Ian Goodfellow and Wojciech Zaremba and Vicki Cheung and Alec Radford and Xi Chen,”Improved Techniques for Training GANs”,June 2016

謝辞

本研究を進めるにあたり，ご指導を頂いた林隆史教授に厚く感謝申し上げます．また，日常の議論を通じて多くの知識や示唆を頂いた林隆史研究室の皆様に感謝いたします．

付 録A　プログラムのソースコード

GLCIC を用いた車線補完の際に使用したプログラムのソースコードを示す.

https://github.com/akisen/WhiteLineComplesion

付 録B　卒業研究発表会質疑応答 (2020.02.13)