

自動運転における深層学習を用いた道路標識の認識

2019SC039 森田航生 2019SC054 下小瀬勇人

2019SC061 竹内祐登

指導教員：中島明

1 はじめに

近年、自動運転の著しい発展に伴い実用化の期待が高まっている。海外では移動サービスや輸送サービスなどで自動運転レベル4(表1)の実用化が進んでおりサービス展開を図っているように、自動運転化は発展し続けている。自動運転化が進むことで、ドライバーの運転負担軽減、交通事故の低減、省エネルギー、高齢化社会での移手段の確保などの効果が期待出来る。

実際に自動車を運転する際には、周囲の状況の認識、なにを行えば良いかの判断、必要な操作を行う必要があり、自動運転でもこれらが制御システムに置き換わる。本研究では、自動車の周囲の環境のうち道路標識を深層学習を用いて認識することを考える。特に、道路上にあり誤認識し易い白線、すなわち停止線とダイヤモンドの認識を試みる。

表1 自動運転レベル

自動運転レベル	概要	運転操作の主体
1 運転支援	アクセル・ブレーキハンドルの操作を支援	ドライバー
2 部分運転自動化	アクセル・ブレーキとハンドルの操作を支援 (ハンズオフ)	ドライバー
3 条件付運転自動化	決められた条件下でシステムが運転。緊急時手動 運転 (アイズオフ)	制御システム
4 高度運転自動化	決められた条件下でシステムが運転 (ブレインオフ)	制御システム
5 完全運転自動化	条件なく自動運転	制御システム

2 深層学習の概要

本章では物体認識の学習にあたって必要となる深層学習について示す.[1]

2.1 機械学習の概要

深層学習とは、大量のデータからルールやパターンを自動で学習する機械学習の手法の一つである。教師あり学習とは、人間がAIに対して問題と答えをセットで教えて、問題をこなしていく中で解き方を学ばせる方法である。教師なし学習とは、人間が答えを出すのではなく、システム自身が学習データを分析することで規則性などを見つける学習方法である。強化学習とは、試行錯誤を繰り返す、より価値の高い行動を学習する方法である。

2.2 深層学習の有用性

従来の機械学習では、画像特徴抽出アルゴリズムや分類ルールの学習アルゴリズムの設計方針が重要となっていた

が、深層学習では別々に行われていた特徴抽出と分類ルールの学習を同時にできる点で有用性がある。

2.3 ニューラルネットワークの学習方法

ニューラルネットワークを学習するための基本的な要素として、損失関数、確率的勾配降下法、誤差逆伝播法の3つが挙げられる。以降で3つについて詳しく説明する。

2.3.1 損失関数

損失関数とは、正解データと対応するニューラルネットワークの出力との差異を計算する関数の事である。ここでよく用いられるものとして、『2乗損失』や『交差エントロピー』が挙げられる。

(1) 2乗損失

$$\mathcal{L}(\mathcal{D}, \Theta) = \frac{1}{2} \cdot \frac{1}{N} \sum_{n=1}^N \|\mathbf{y}_n - \hat{\mathbf{y}}_n\|^2$$

*N: データ数

\mathbf{y}_n は正解データ、 $\hat{\mathbf{y}}_n$ は出力データである。

(2) 交差エントロピー

$$\mathcal{L}(\mathcal{D}, \Theta) = -\frac{1}{N} \sum_{n=1}^N \sum_{k=1}^K y_{n,k} \log \hat{y}_{n,k}$$

*K: カテゴリ数

Nはデータの個数、Kはクラスの数。もし服のサイズS,M,Lだったら場合K=3となる。正解データは、正解レベルの時「1」、その他の時「0」のベクトルである。

2.3.2 確率的勾配降下法

損失関数は、各データに対する損失の平均として以下のように定義される。

$$\mathcal{L}(\mathcal{D}, \Theta) = \frac{1}{N} \sum_{n=1}^N \mathcal{L}(\mathbf{x}_n, \mathbf{y}_n, \Theta)$$

大前提としてモデルの予測値と実際の値との誤差(損失)をなくすことが目的なため、この損失関数を最小にしたい。これを実現するために以下の勾配降下法(バッチ学習法)がある。

$$\mathbf{W}_{t+1}^{(l)} = \mathbf{W}_t^{(l)} - \frac{\alpha}{N} \sum_{n=1}^N \frac{\partial \mathcal{L}(\mathbf{x}_n, \mathbf{y}_n, \Theta_t)}{\partial \mathbf{W}_t^{(l)}}$$

* α : 学習率, \mathbf{W} : 重み行列, t : 更新ステップ数

しかし問題点として、多くの時間を必要としたり、局所解や鞍点から抜け出せなくなるということが挙げられる。その解決策として確率的勾配降下法 (ミニバッチ学習法) がある。データ範囲を絞ることにより、勾配の計算コストを削減できるのである。

$$\mathbf{W}_{t+1}^{(l)} = \mathbf{W}_t^{(l)} - \frac{\alpha}{B} \sum_{b=1}^B \frac{\partial \mathcal{L}(\mathbf{x}_b, \mathbf{y}_b, \Theta_t)}{\partial \mathbf{W}_t^{(l)}}$$

* B : ミニバッチサイズ ($1 \leq B \leq N$)

確率的勾配降下法の収束速度や解の精度は、ステップ幅に依存していることが分かります。これによりステップ幅を適応的に変えることで、モメンタム法, Ada-grad, RMSprop, Adam などの代表的な改良手法もある。

2.4 誤差逆伝播法

誤差逆伝播法とは、具体的に勾配を計算する方法の1つである。損失関数をパラメータで微分した値を求めると、勾配法の更新が可能になり、各層の微分さえ分かれば、勾配が求められる。

3 YOLO を用いた物体検出

以上に示した、深層学習を用いた物体検出に YOLO がある、本章では、YOLO の概要について示す。[2]

3.1 YOLO の概要

深層学習を用いた物体検出手法として YOLOv5 (You Only Look Onse) がある。YOLO とはリアルタイムオブジェクト検出アルゴリズムであり、画像を一度ニューラルネットワークに通すことにより、オブジェクト検出とクラス分類のプロセスを同時に行うことが出来る。また、処理が早い、画像全体を見て予測することがきるというメリットがある。

3.2 YOLOv5 を用いた物体検出の手法

YOLO の最新のものに YOLOv5 がある。YOLOv5 を用いた物体検出を実際に行った手法を簡単に説明する [2]。

1. 教師データとなる画像を大量に (300 枚程度) 集める。
2. 教師データの画像に対して物体がある位置とクラスに注釈を付ける。
3. 上記で作成した教師データを YOLOv5 に学習させる。
4. 学習させた教師データを重みファイルとして使用し、物体検出を行う。
5. 実行後、フレームとラベル (どのくらいの確率で何というオブジェクトであるか示したもの) 付きの画像ファイルが保存される。以上が今回行った物体検出の手法である。

3.3 YOLOv5 を動作させた環境

YOLOv5 を動作させた環境を以下に示す。

使用 PC スペック

Os : windows10

CPU : Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz

2.70 GHz

メモリ : 8.0GB

GPU : Intel(R) HD Graphics 620

python 環境

python : 3.9

3.4 認識精度の評価指標

YOLO の評価指標として Precision, Recall, AP (Average Precision), mAP (mean Average Precision) が用いられる。本章でもこれらの評価指標を用いて認識精度の評価を行う。

4 YOLOv5 を用いた停止線、交通標識の認識

本研究では実際の停止線、交通標識を用いて自動車における認識を行った以下に内容を示す。教師データの作成では図 1, 2 のように、停止線、ダイヤモンドに分けてラベリング (座標とクラスをタグ付け) を行った。



図 1 停止線



図 2 停止線

4.1 認識を行った手順

実際に認識を行った順序は以下の通りである。

- ・エポック回数固定でバッチサイズを変化させた場合
- ・バッチサイズが固定 (バッチ 8) でエポック回数を変化させた場合
- ・明るさを変化させた場合
- ・擦れがあるものないもので分けた場合
- ・角度変更させた場合
- ・全ての場合分けを合わせた場合

ここでエポック回数とは、勾配を繰り返し計算して誤差が

一定値に収束するまでの学習を1エポックと言い、それを何回繰り返すかの数のことである。バッチサイズとは、画像をいくつかのまとまりに分ける正方形のサイズのことである。

4.2 エポック回数固定 (エポック 50) でバッチサイズを変化させた場合

実際の道路にある停止線、交通標識 (ダイヤモンド) の写真を撮り、学習・認識を行った。

4.2.1 目的

今後の実験で用いるバッチサイズを決めるために事前実験としてバッチサイズを変更し、適切なものを選ぶことが必要だと考えたためである。

4.2.2 実験結果

エポック回数固定でバッチサイズを変化させた場合の信号機の認識結果をまとめた表を表2とすると以下のようになる。また、学習の設定について以下に示す。

学習データの枚数:710
エポック回数:50
学習率:0.01

表2 エポック回数固定 (エポック 50) でバッチサイズを変化させた場合

バッチとエポック	dia 昼	夜	停止昼	夜
エポック 50 バッチ 4	0.82	0.71	0.74	0.82
エポック 50 バッチ 8	0.74	0.78	0.69	0.66
エポック 50 バッチ 16	0.76	0.72	0.62	0.81

4.2.3 評価

バッチサイズを変化させても精度が向上するわけではないため、今後の実験ではバッチサイズは8とする。

4.3 バッチサイズが固定 (バッチ 8) でエポック回数を変化させた場合

4.2と同様の写真を用いて、学習・認識を行った。

4.3.1 目的

今後の実験で用いるエポック回数を決めるために事前実験としてエポック回数を変更し、適切なものを選ぶことが必要だと考えたためである。

4.3.2 実験結果

バッチサイズが固定でエポック回数を変化させた場合の信号機の認識結果をまとめた表を表3とすると以下のようになる。ここで4.2で行った結果と比べてエポック回数以外は変えていない。また、学習の設定について以下に示す。

学習データの枚数:710
バッチサイズ:8
学習率:0.01

表3 バッチサイズが固定 (バッチ 8) でエポック回数を変化させた場合

バッチとエポック	dia 昼	夜	停止昼	夜
エポック 10 バッチ 8	0	0	0	0
エポック 20 バッチ 8	0	0.74	0	0
エポック 30 バッチ 8	0.62	0.89	0.5	0.37
エポック 40 バッチ 8	0.72	0.7	0.54	0.71
エポック 50 バッチ 8	0.74	0.78	0.69	0.66
エポック 100 バッチ 8	0.82	0.79	0.89	0.87
エポック 150 バッチ 8	0.73	0.73	0.91	0.85
エポック 200 バッチ 8	0.87	0.9	0.94	0.96
エポック 300 バッチ 8	0.89	0.86	0.96	0.96

4.3.3 評価

エポック回数が10~20回の場合は、認識出来ていない。エポック回数が30~150回の場合は、0.9を超えている箇所はほぼない、エポック回数200回、300回で比べた際300回のが精度が良い。以上の結果からこれ以降の実験では、エポック回数は300回とする。

4.4 明るさを変化させた場合

明るい場合と暗い場合の2パターンで写真を撮り、学習・認識を行った。明るい場合は昼、暗い場合は夜として区別した。

図3のような明るい場合、暗い場合において認識精度の評価、比較を行った。



図3 左:明るい (昼), 右:暗い (夜)

4.4.1 目的

実際に自動運転を行う際には、明るい時間帯(昼)、暗い時間帯(夜)と存在する。暗い時間帯は交通標識が見にくい場合もあるので、明るさを変更させて学習・認識させるべきだと判断したからである。

4.4.2 実験結果

損失は0に、precision,recall,mApは1に収束しているため、正しく学習させることが出来たことを示している。

明るさを変更させた場合の認識結果をまとめた表を表4とする以下のようになるまた、以下に学習設定を示す。

学習データの枚数:824
 検証データの枚数:211
 バッチサイズ:8
 エポック回数:300
 学習率:0.01

表4 明るさを変化させた場合の認識結果

認識に使用した画像/条件	昼	夜
ダイヤモンド(昼)1	0.99	0.97
ダイヤモンド(昼)2	0.98	0.99
ダイヤモンド(夜)1	0.97	0.99
ダイヤモンド(夜)2	0.97	0.44
停止線(昼)1	0.96	0
停止線(昼)2	0.93	0.78
停止線(夜)1	0	0.95
停止線(夜)2	0.86	0.94

4.4.3 評価

教師データが昼の場合の方が全体的に認識の精度が高い。また、昼の学習データだけを学習させた時は夜の停止線、夜の学習データだけを学習させた時は、昼に停止線の認識が出来ていない。

これらの結果から、明るい方が交通標識の形が認識しやすいため認識の精度が上がったのではないかと考えた。また、昼夜を同時に学習させることでどちらかの状況の停止線が認識出来ないという現象が起きないと考えた。

そこで交通標識の形に着目し、線が擦れているもの、完全に形になっているものと場合に分けて実験を行う。

4.5 線が擦れていない場合と線が擦れている場合

交通標識の線が擦れていない場合と擦れている場合の2パターンで写真を撮り、学習・認識を行った。認識画像5枚で平均と分散を集計した。

図4のような擦れていない場合、擦れている場合において認識精度の評価、比較を行った。



図4 左:擦れていない、右:擦れている

4.5.1 目的

4.4の実験より、交通標識の形を認識しているのかを判断するために線が擦れていない、擦れている状態の認識結果を調べ、線の形が認識にどのような影響を与えるのかを確認するためである。

4.5.2 実験結果

損失は0に、precision,recall,mApは1に収束しているため、正しく学習させることが出来たことを示している。

擦れていない場合、擦れている場合の認識結果をまとめた表を表5とする以下のようになる。また、以下に学習設定を示す。また、表6は認識の画像を5枚にして平均と分散を求めた表である。

学習データの枚数:637
 検証データの枚数:182
 バッチサイズ:8
 エポック回数:300
 学習率:0.01

表5 交通標識が擦れていない、擦れている場合の認識結果

認識に使用した画像/条件	擦れなし	擦れあり
ダイヤ擦れあり1(昼)	0.92	0.98
ダイヤ擦れあり2(夜)	0	0.98
ダイヤ擦れなし1(昼)	0.98	0.96
ダイヤ擦れなし2(夜)	0.98	0
停止線擦れあり1(昼)	0.93	0.97
停止線擦れあり2(夜)	0.85	0.96
停止線擦れなし1(昼)	0.95	0.88
停止線擦れなし2(夜)	0.97	0.95

表 6 交通標識が擦れていない, 擦れている場合の認識結果 (平均と分散)

認識に使用した画像/条件	擦れあり平均	分散	擦れなし平均	分散
ダイヤ擦れあり 1(昼)	0.59	0.29	0.18	0.17
ダイヤ擦れあり 2(夜)	0.51	0.24	0	0
ダイヤ擦れなし 1(昼)	0.19	0.18	0.82	0.06
ダイヤ擦れなし 2(夜)	0	0	0.96	0.0003
停止線擦れあり 1(昼)	0.61	0.15	0.48	0.22
停止線擦れあり 2(夜)	0.76	0.18	0.72	0.17
停止線擦れなし 1(昼)	0.90	0.007	0.88	0.03
停止線擦れなし 2(夜)	0.71	0.16	0.91	0.01

4.5.3 評価

ダイヤマークは教師データが擦れありのみの時の夜の擦れなし, 教師データが擦れなしのみの時の夜の擦れありの認識結果が 5 枚とも認識なし. 停止線の場合はダイヤマークと比べて認識出来ている,

これらの結果からダイヤマークは条件によって大きな影響を受けると考えた. また, 停止線は擦れによる条件の影響をあまり受けないのではないかと考えた.

そこで, 交通標識の角度に着目し正面, 正面以外と場合に分けて実験を行う.

4.6 角度を変更させた場合

交通標識の教師データが正面と正面以外の 2 パターンで写真を撮り, 学習・認識を行った. 認識画像 5 枚で平均と分散を集計した.

図 5 のような正面の場合, 正面以外の場合において認識精度の評価, 比較を行った.



図 5 左:正面, 右:正面以外

4.6.1 目的

4.5 の実験より, 交通標識の形以外を認識しているのかを判断するために正面, 正面以外の認識結果を調べ, 線の形が認識にどのような影響を与えるのかを確認するためである.

4.6.2 実験結果

損失は 0 に, precision, recall, mAp は 1 に収束しているため, 正しく学習させることが出来たことを示している.

正面からの場合, 正面以外の場合の認識結果をまとめた表を表 7 とする以下ようになる. また, 以下に学習設定を示す. また, 表 8 は認識の画像を 5 枚にして平均と分散を求めた表である.

学習データの枚数:301

検証データの枚数:86

バッチサイズ:8

エポック回数:300

学習率:0.01

表 7 交通標識の角度を変更させた場合の認識結果

認識に使用した画像/条件	正面	正面以外
ダイヤ正面 1(昼)	0.98	0.88
ダイヤ正面 2(夜)	0.99	0.93
ダイヤ正面以外 1(昼)	0.90	0.96
ダイヤ正面以外 2(夜)	0.91	0.97
停止線正面 1(昼)	0.72	0.45
停止線正面 2(夜)	0.93	0.93
停止線正面以外 1(昼)	0	0.65
停止線正面以外 2(夜)	0.85	0.95

表 8 交通標識の角度を変更させた場合の認識結果 (平均と分散)

認識に使用した画像/条件	正面平均	分散	正面以外平均	分散
ダイヤ正面 1(昼)	0.58	0.28	0.18	0.16
ダイヤ正面 2(夜)	0.57	0.27	0.18	0.17
ダイヤ正面以外 1(昼)	0.52	0.23	0.52	0.24
ダイヤ正面以外 2(夜)	0.48	0.21	0.57	0.27
停止線正面 1(昼)	0.51	0.12	0.18	0.06
停止線正面 2(夜)	0.56	0.25	0.64	0.15
停止線正面以外 1(昼)	0.37	0.14	0.33	0.12
停止線正面以外 2(夜)	0.36	0.25	0.84	0.01

4.6.3 評価

検証用画像が 1 枚と 5 で数値が大きく異なり, 分散も大きい. このことから, 1 枚のみで結果を集計するのは偏った結果になりやすいと分かった. また 0.9 を超えた結果がなく, 認識精度が低い. 角度による場合分けは学習効果が低いのではないかと考えた.

今回の実験と実験 4.4 から対になる条件のものを同時に学習させることで認識の精度を更に上げることが出来るのではないかと考えた.

4.7 全ての場合分けを合わせた場合

最後に場合分けを行ったものを全て学習させた。認識画像 5 枚で平均と分散を集計した。

4.7.1 目的

実際の自動運転には様々な条件が存在する。そのためどんな時でも正しく認識出来るように様々な想定をした学習が必要となる。ここまで行った場合分けを全て合わせることで認識がどのような結果か調べるためである。

4.7.2 実験結果

損失は 0 に、precision, recall, mAp は 1 に収束しているため、正しく学習させることが出来たことを示している。

正面からの場合、正面以外の場合の認識結果をまとめた表を表 9 とする以下ようになる。また、以下に学習設定を示す。また、表 10 は認識の画像を 5 枚にして平均と分散を求めた表である。

学習データの枚数:1610
検証データの枚数:460
バッチサイズ:8
エポック回数:300
学習率:0.01

表 9 全ての場合分けを行った時の認識結果

認識に使用した画像/条件	ダイヤモンド	停止線
昼擦れあり正面	0.97	0.97
昼擦れあり正面以外	0.91	0.87
昼擦れなし正面	0.98	0.97
昼擦れなし正面以外	0.98	0.93
夜擦れあり正面	0.98	0.94
夜擦れあり正面以外	0.97	0.96
夜擦れなし正面	0.98	0.97
夜擦れなし正面以外	0.97	0.95

表 10 全ての場合分けを行った時の認識結果 (平均と分散)

認識に使用した画像/条件	ダイヤ平均	分散	停止線平均	分散
昼擦れあり正面	0.95	0.0009	0.94	0.003
昼擦れあり正面以外	0.18	0.17	0.32	0.20
昼擦れなし正面	0.96	0.001	0.77	0.19
昼擦れなし正面以外	0.43	0.19	0.95	0.0002
夜擦れあり正面	0.76	0.18	0.76	0.18
夜擦れあり正面以外	0.19	0.19	0.72	0.17
夜擦れなし正面	0.98	0.00002	0.94	0.001
夜擦れなし正面以外	0.47	0.24	0.88	0.01

4.7.3 評価

1 枚の時に 0.9 以上出ていた結果が 5 枚の時に大きく精度が下がっている。新しく撮った画像が認識し辛いのでは

と考えた。教師データが全ての場合分けを合わせた場合でも認識なしの結果になる場合もある。0.9 を超えている箇所は今までの実験のなかで最も多いため様々な場合分けを同時に学習させることで認識の精度をあげることが出来ると分かった。

また、全ての場合分けをまとめて実験を行った場合でも認識なしになることがあった。

5 おわりに

本研究では物体検出の基本となる深層学習や、ニューラルネットワークについて学習し、実際に最新の物体検出の手法 YOLO v5 を用いて物体検出を行った。今後の課題としては、擦れがある時の正面以外の時の精度の向上、リアルタイム認識などがあげられる。また、全ての場合分けをまとめて実験を行った場合でも認識なしになることがあった。自動運転において認識なしの結果になるのは事故につながってしまう。そのためさらなる精度の向上が求められる。

参考文献

- [1] 日本ロボット学会 監修, 香月 理絵 編著, 荒井 幸代 etc 共著『自動運転技術入門』. オーム社, 東京, 2021.
- [2] 株式会社アルファシステムズ:『YOLOv5 を使った物体検出』. アルファテックブログ, 2021-08-20, https://www.alpha.co.jp/blog/202108_02.