

機械学習を用いたロボット関連製品の制御技術の開発

(オープンソースを活用したディープラーニング導入手法の確立と技術支援)

機械システム科 主任研究員 堀江 貴雄

本研究では、各種センサからのデータを用いてディープラーニングにより end-to-end で無人搬送車 (Automatic Guided Vehicle : AGV) を自律制御することを目的とした。令和元年度から2年度にかけて開発してきたニューラルネットでは、前処理なしの時系列データを用いて AGV の自動走行が実現できることを確認しているが、未学習環境における挙動が未知であり、安全面で課題があった。そこで最終年度となる令和3年度では Autoencoder によるカラー画像からの距離画像推定を行い、未学習環境の検出と自動停止機能を実現した。この結果、自動走行中に未知のシーンを検出した時、自動停止可能なことを確認した。

1. 緒言

近年ではディープラーニングによる画像判別や機械制御の事例も多く報告されるようになった。特に、各種データから特徴量を自動的に抽出できることから、これまで人による設計が困難であった高度な認識システムの実現に期待がかけられている。

これまでの研究において、オープンソフトウェアの Neural Network Console (ソニー社製)^[1]を用いたニューラルネットワーク開発により、5 時刻分のカラー画像、距離画像、車両方位角から AGV の移動制御パラメータを制御して複数の移動経路を自律走行できる移動ロボットを試作した^[2-5]。

令和3年度は、ニューラルネットによる制御において問題となる未学習環境での異常動作を抑制するため、Autoencoder を取り入れ、未学習環境の検出と自動停止機能の実装に取り組んだ。

2. メカナム移動ロボット

昨年度までに試作した移動ロボットを基本コンセプトに、前輪2輪をサブフレームに固定し、サブフレームが本体フレームに対してロール方向に動くよう改良した検証用メカナム移動ロボットを作成した (図1)。これによって4輪の接地性が向上し、よりスムーズな全方向移動が可能となっている。

搭載機器の仕様を表1に示す。

表1 メカナム移動ロボット機器仕様

機器名	仕様	用途
画像, 深度画像センサ Intel RealSenseD455	カラー画像 1920×1080, 深度画像 1920×1080	車両前方画像、距離計測
方位センサ WitMotion HWT905	3軸 XYZ 加速度、角速度、磁気 200Hz 出力	車両方位計測
タッチパネル	13.3 インチ USB タッチパネルディスプレイ 1920×1080	制御アプリケーション操作
ブラシレスモータ OrientalMotor BLHM230K	DC 電源駆動、ギヤヘッド付き、デジタル制御	メカナムホイール駆動用
マイコン ArduinoUNO	USB 電源駆動、デジタル IO	ブラシレスモータ同期制御用
メカナムホイール NexusRobot 127mm	ホイール直径 127mm	全方向移動用ホイール
PC Intel NUC10i7FNH	Corei7-1070U (max4.7GHz) RAM32GB	ニューラルネット推論、操作インタフェース、制御
ゲームパッド Microsoft BlueTooth 対応 Xbox コントローラ	Bluetooth 通信対応 左右レバー、十字キーボタン 8 入力	データセット生成用手動操縦、オーバーライド用

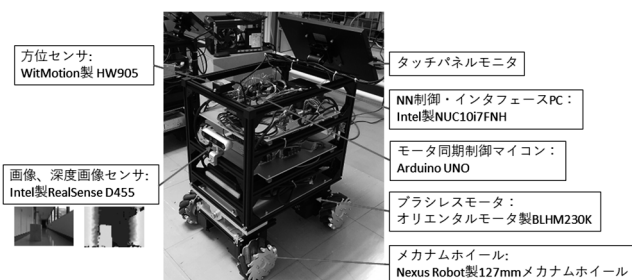


図1 メカナム移動ロボット

データセット収集は、ゲームパッド操作による手動操縦で行う。ニューラルネットの学習は別途 GPU (NVIDIA 製 TITAN RTX) を搭載した PC で実施し、学習結果をロボット搭載の Intel 製 NUC10i7FNH に実装することで動作検証を実施した。

3. Autoencoder

異常データの収集が難しい状況で、正常データのみから、異常を検出するための手法として、自己符号化器 (Autoencoder) と呼ばれるニューラルネットの教師なし機械学習方法がある。

Autoencoder は、入力データを一旦、次元圧縮 (エンコード) し、その後デコードする。このとき出力データが入力データと一致するように学習を行わせる。

このネットワークは、学習データに類似したデータでは高い精度で入力データを再現するが、学習していないデータではうまく入力データを再現出力できない特徴を持つ。したがって入力データと出力データの差分をとり、ある閾値以上の相違があれば、学習していないデータとみなして異常検出とすることができる。

本研究でターゲットとする移動ロボットにおいても、適切な操縦データは大量に含まれるが、一度も走行したことのない環境の操縦方法は含まれず、ロボットがどのような挙動をとるかは想定困難である。このことがニューラルネットベース制御の実用化ネックになっている。そこで本研究では Autoencoder を活用し、これまでに学習した環境かどうかをニューラルネット自身に判定させ、未学習環境と検知した場合、安全面から自動停止する機能を開発する。

3. 1 画像推定 Autoencoder

この手法を今回の例に適用するため、まずは入力データにカラー画像を与え、カラー画像を出力する Autoencoder (以後 PTP Autoencoder と表記する) を学習することとした。学習させるデータセットを表 2 に示す。

表 2 PTP Autoencoder 用データセット

Input	Output
カラー画像 (3×45×80)	Input カラー画像

人や障害物が存在する屋内通路を移動ロボットを走行させて取得したデータセット 82,000 個を用意し、そのうち 74,000 個を学習用とした。残りの 8,000 個の

データは学習に使わず、テスト用の学習済み環境データセットとした。

未学習環境データとして、学習環境データセット 82,000 個には一度も映らない工具箱や階段踊り場などを含むデータを 500 個用意した。

入力するカラー画像が学習済みかどうか判定するためにエラー値を E とし、画像幅 w 、高さ h の入力画像 I_i のある点 $I_i(x, y)$ と、対応する出力画像 I_o の点 $I_o(x, y)$ の差分を合計したエラー値 E は式 1 で求められる。

このエラー値を用いて適切な閾値を設定できれば、ニューラルネットは自身で未学習環境であると判定できる。

$$E(I_i, I_o) = \sum_{x=0}^{w-1} \sum_{y=0}^{h-1} (I_o(x, y) - I_i(x, y))^2 \quad (1)$$

2500epoch 学習させ、式 1 を用いて各エラー値を求めた結果を図 3 に示す。学習済み環境画像 8000 枚に対して平均 109 (もっとも左)、未学習環境画像 500 枚に対しては平均 196 (左から 2 番目) だった。学習済み環境と、未学習環境で似通った値となり、十分な判定ができていないことがわかった。

3. 2 距離推定 Autoencoder

次に入力画像にカラー画像を与え、出力画像が D435 センサの距離画像となる Autoencoder (以後 PTD Autoencoder) を考える。PTP Autoencoder 学習時と同じデータセットのうち、Input としてカラー画像を与え、Output として距離画像を与える (表 3)。学習するネットワークアーキテクチャは図 2 に示す。

表 3 カラー画像から距離画像推定用データセット

Input	Output
カラー画像 (3×45×80)	距離画像 (3×45×80)

2500epoch 学習後に式 1 を用いてエラー値を評価した結果を図 3 に示す。

学習済み環境画像 8000 枚に対して平均 209 (左から 3 番目)、未学習環境画像 500 枚に対して平均 1666 (もっとも右) となった。

PTP Autoencoder と PTD Autoencoder の出力画像比較を図 4 に示す。

検証に用いたデータは学習済、未学習ともにニューラルネット学習時のデータセットには含まれていない。学

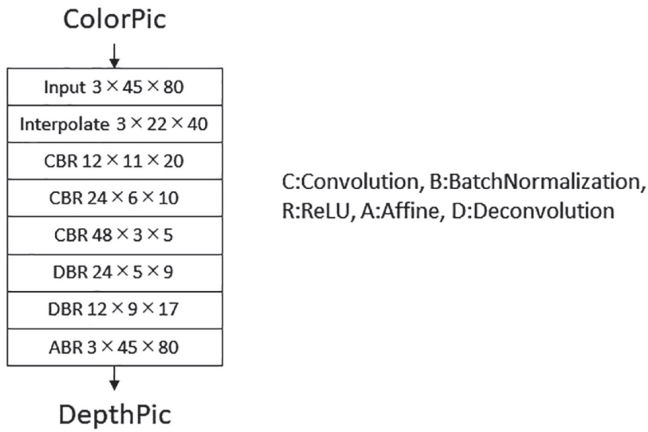


図2 PTD Autoencoder によるカラー画像からの距離推定

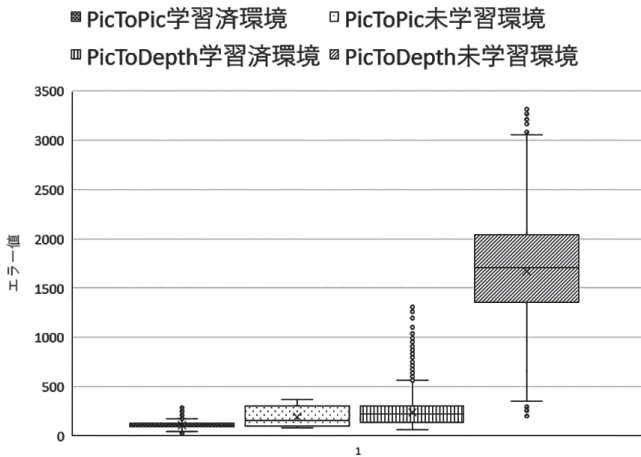


図3 Autoencoder による未学習環境検出

習済み環境のカラー画像を PTP Autoencoder と PTD Autoencoder に与えると、どちらも良好な推定を出力し、エラー値 (E) はそれぞれ 100、200 程度となる。

未学習環境のカラー画像を与えると、カラー画像を出力する PTP Autoencoder は未学習環境であっても一定程度入力画像を再現でき、エラー値は 200 付近を示す。

一方、距離画像を出力する PTD Autoencoder では、未学習環境のカラー画像を与えようまく距離を推定できず、エラー値は 1500 付近を示す。画像 1 枚から 3D 認識を行っており、PTP Autoencoder より困難なタスクを与えられていると考えられ、うまく奥行きを推定できていないと考える。

このことから、同規模のネットワークで未学習環境を検出するためには PTD Autoencoder を用いたほうがより分離度が高く有効であることがわかった。

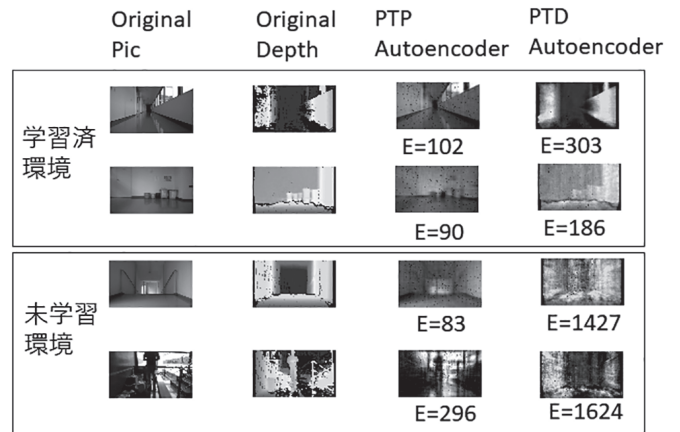


図4 Autoencoder 比較

4. 自動走行ネットワークとの結合と検証

昨年度までに開発したネットワークは、5 時刻分 (最新時刻 T、T-1 (1 秒前)、T-2、T-3、T-4) のカラー画像、距離画像、方位センサデータ、ゴール番号、ルート番号を入力データとし、ゲームパッド操作量であるスピード値、平行移動値、回転速度値を推定する [5]。

このネットワークは RNN [6] アーキテクチャを用いて実装している。カラー画像、距離画像ごとに T-4 時刻分のデータを起点に処理後、その情報を再帰構造で次の T-3 時刻に送り、ここでさらに統合して処理後、T-2 時刻処理に送るといのように、最新時刻 T 処理まで繰り返す。その後、カラー画像処理結果、距離画像処理結果、磁気センサ X、Y、ゴール番号、ルート番号と集計し、速度、平行移動方向、回転速度を推定していた。

今年度開発したネットワークは Autoencoder を結合するため、このままでは計算量が多く、本研究で試作しているロボットに搭載される CPU ではスループットが十分でない。

計算量の削減と推定精度の維持を目的に、ResNet [7] [8] アーキテクチャを用いて改良し、カラー画像処理部を Pic Conv Unit (図 5)、距離画像処理部を Depth Pic Conv Unit (図 6) としてユニット化した。

各時刻において、カラー画像、距離画像をそれぞれ Pic Conv Unit、Depth Pic Conv Unit で処理後、磁気センサ X、磁気センサ Y、ゴール番号、ルート番号と結合し、全結合層で処理した結果を、次時刻の再帰処理へ送るようにネットワークを構成した (図 7)。

さらに 3. 2 の PTD Autoencoder を統合するために、最新時刻 T のカラー画像を入力として、出力に T 時

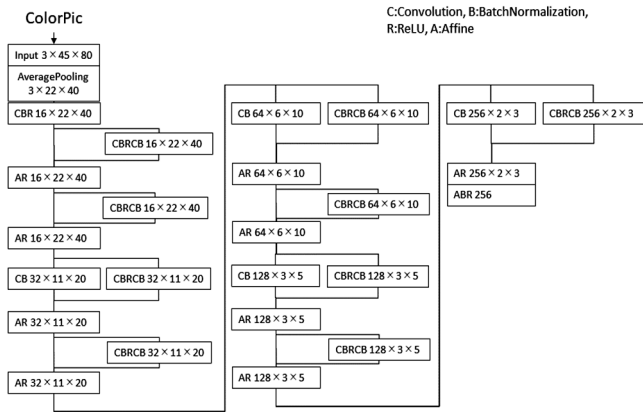


図5 Pic Conv Unit アーキテクチャ

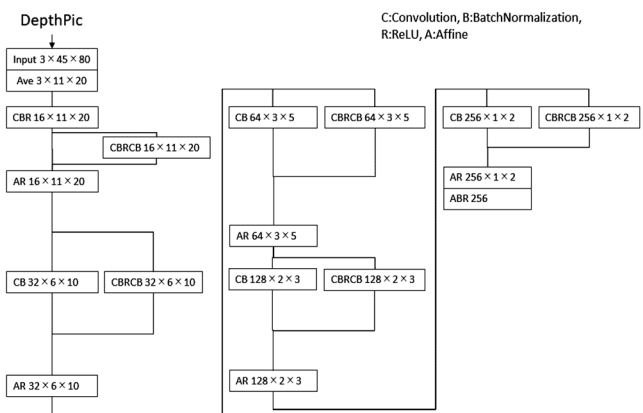


図6 Depth Pic Conv Unit アーキテクチャ

表4 統合学習用データセット

input	output
カラー画像 (3×45×80) ×5	スピード値 (-1 ~ 1)
距離画像 (3×45×80) ×5	平行移動方向値 (-1 ~ 1)
磁気センサ X (-1.0 ~ 1.0) ×5	回転速度値 (-1 ~ 1)
磁気センサ Y (-1.0 ~ 1.0) ×5	
ゴール位置 0 (0 or 1) ~ 9 (0 or 1)	
ルート選択番号 (-1 or 0 or 1)	

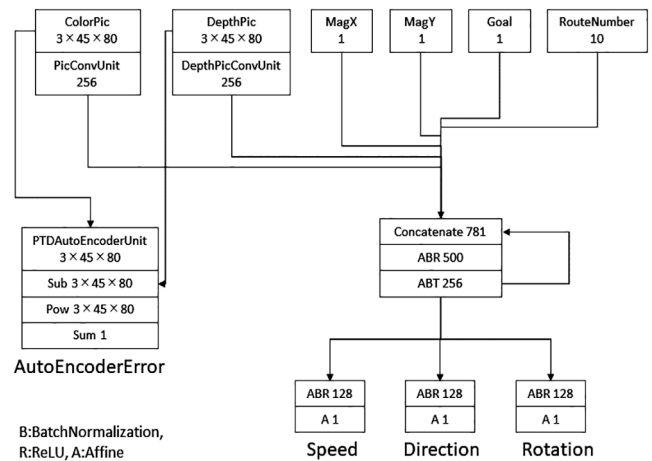


図7 統合推定ネットワーク

刻距離画像を与え同時学習させる。T 時刻推定距離画像のエラー値は、3.1 の 1 式をニューラルネット上で計算し出力する。これによって、スピード値、平行移動、回転速度、Autoencoder の統合した end-to-end 学習が可能である。なお、ネットワーク全体の計算規模を表す全乗加算数は 270,888,064 となった。

統合ネットワークを学習させるためのデータセット構成を表4に示す。このデータセットを 120,000 個収集し、800epoch 学習させた結果、Best Validation は 388.98264 を得た。学習後、自動停止とするエラー値を検討し、1250 を閾値とした。

工業技術センター 2 階の周回約 250 m を対象に自動走行実験を行わせた結果、平均時速 3 km/h で走行可能なことを確認した。

また、階段踊り場など、学習データに含まれない環境では、未学習環境を検知し停止できることが確認できた。想定外の状況に対して一定程度の自動停止が期待できる。以上の結果から、本研究で提案するネットワークが屋内走行機能と、未学習環境での自動停止を

備えた一定の安全機能を備えることが確認できた。

5. 結言

本研究では、カラー画像と距離画像によるメカナム台車の模倣学習を試みた。昨年までの、複数場所への区間移動、障害物回避、人追跡に加え、今年度は Autoencoder による異常停止機能の導入を図った。

カラー画像から距離画像を推定する Autoencoder を学習させることで、未学習環境の検出が可能となった。

この Autoencoder を昨年度までの自動制御ニューラルネットに統合し、ネットワーク全体のコンパクト化調整を実施後、同時学習させた。学習後にロボットへ実装し、工業技術センター通路にて、移動機能と自動停止機能の有効性を確認した。

今後は、屋内だけでなく屋外での自動走行、メカナム台車にロボットアームを追加したうえで、エレベー

タ等を活用した多層階の移動など機能拡張を実施していきたい。

参考文献

- [1] “Neural Network Console”,
<https://dl.sony.com/ja/>, Accessed 2018.4.
- [2] 堀江貴雄：Neural Network Console を使用したメカナム台車制御方法の開発，ロボティクス・メカトロニクス講演会 2020, 1A1-G10, 2020.
- [3] 堀江貴雄：カラー画像と距離画像を用いた模倣学習によるメカナム台車の移動制御，第 38 回日本ロボット学会学術講演会, 3A3-06, 2020.
- [4] 堀江貴雄：機械学習を用いたロボット関連製品の制御技術の開発，長崎県工業技術センター研究報告, No.49, pp.18-21, 2020.7.
- [5] 堀江貴雄：機械学習を用いたロボット関連製品の制御技術の開発，長崎県工業技術センター研究報告, No.50, pp.14-15, 2021.7.
- [6] 小林由幸：Deep Learning 入門 Recurrent Neural Network とは？,
<https://www.youtube.com/watch?v=yvqgQZIUAKg>, Accessed 2020.6.
- [7] 小林由幸：Deep Learning 入門 ニューラルネットワークの多層化テクニック,
<https://www.youtube.com/watch?v=X2KWO1UPqzk>, Accessed 2020.10.
- [8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun : Deep Residual Learning for Image Recognition,
<https://arxiv.org/abs/1512.03385>, Accessed 2020.10.