機械学習を用いたロボット関連製品の制御技術の開発

電子情報科　主任研究員　　堀　江　貴　雄

機械学習の手法であるディープラーニング（深層学習）は近年、物体判別、各種診断、予測など幅広い分野で注目されている。当初は画像診断の事例で注目されてきた技術であるが、本来画像診断に限らず、複雑な関数をif-then ルールでプログラムすることなく導出できる汎用技術である。したがって、十分な計算資源があればあらゆる問題に適用できると考えられている。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。そこで、先ずディープラーニングを活用した画像と方位角による位置推定を行った。次に、画像と方位角から AGV の走行に必要な速度、移動方向、回転方向の走行制御パラメータを推定した。この結果、前処理なしの上記の入力データから AGV の自律走行を制御できることを確認した。

1．緒　言

無軌道による AGV 制御には、距離データや慣性データ、及び車輪回転数などを用いて環境地図の作成と自己位置推定を同時におこなう SLAM（Simultaneous Localization and Mapping）と呼ばれる手法が従来から用いられている。

一方、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。

本研究では、オープンソフトウェアの Neural Network Console（ソニー社製）[1] を用いたディープラーニングにより画像と車両の方位角から 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．Neural Network Console とメカナム AGV

2．1 Neural Network Console

近年、機械学習の一種であるディープラーニングが注目され、画像分類問題からはじまり、自動運転やロボット制御など様々な問題への適用が試みられている。多くはオープンソフトウェアを用いて具体的な開発が行われているが、これまでの if-then ルールによるプログラムと大きく異なるため、初学者には敷居が高いのも実情である。

Neural Network Console は機械学習の専門家でなくてもオリジナルのニューラルネットワークを設計し、その効果を検証できることを念頭に、ソニー社によって開発された機械学習用ソフトである。ネットワークの設計は豊富に用意されたコンポーネントをマウスでドラッグアンドドロップし、これらコンポーネントを線でつなぐことで可能であり、非常に直感的に構築可能である。また、データセットの確認、学習曲線の経過、検証結果についてもボタン操作のみで容易に可能である。そこで、本研究では効率的な開発を実施するため、この Neural Network Console を用いることとした。

2．2 メカナム AGV

試作した AGV はメカナムホイール（14193L,14193R､ヴイストン社製）を有している。車両の前方にはカラー画像を取得するためのカメラ（RealSense Depth Camera D435i、インテル社製）、上部には車両の方位角を測定する磁気センサ（HWT905、WITMOTION 社製）を搭載している（図 1）。

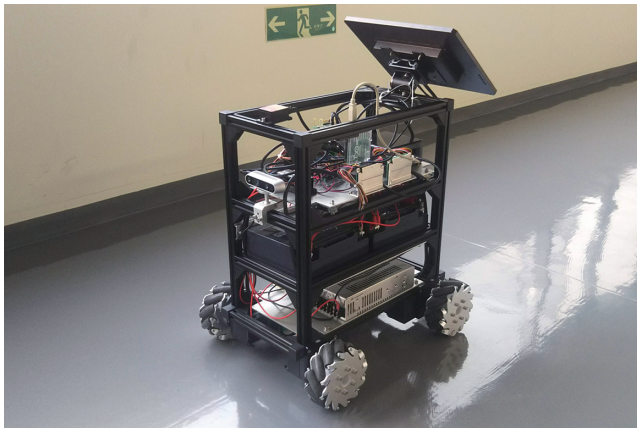

図 1　メカナム AGV

メカナムホイールはギヤヘッド付きブラシレスモータの出力軸に固定されている。各ブラシレスモータは各モータドライバによって速度制御可能であり、マイコン（Arduino UNO、Arduino 社製）を中継して制御用パソコンと接続されている。

AGV への指令は PC から USB を用いてマイコンへ各モータ回転速度と回転方向を指示することで可能である（図 2）。

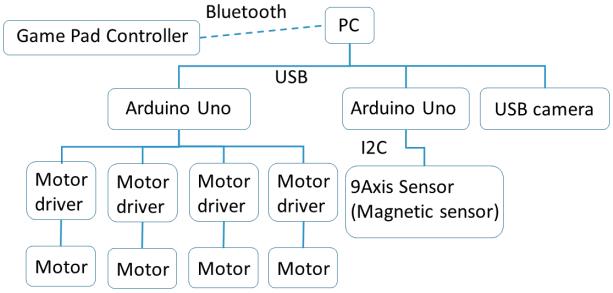

図 2　システム構成

また、市販のゲームパッドコントローラーで操作できるようあらかじめプログラムを作成した。左スティックレバーの 2 軸入力で、スピード、および平行移動方向を指示でき、右レバーの左右方向入力でヨーイング方向の回転速度を指示できる。2 つのレバーを組み合わせることで、全方位への移動制御が可能である。

磁気センサの更新頻度は 50 Hz、画像の更新頻度は 5 Hz、コントローラ読み込み頻度は5 Hz である。また、下記の 3．ネットワーク設計と学習結果で述べる学習データセットを様々な条件下で効率的に生成するためのコントロール画面（図 3）を作成した。

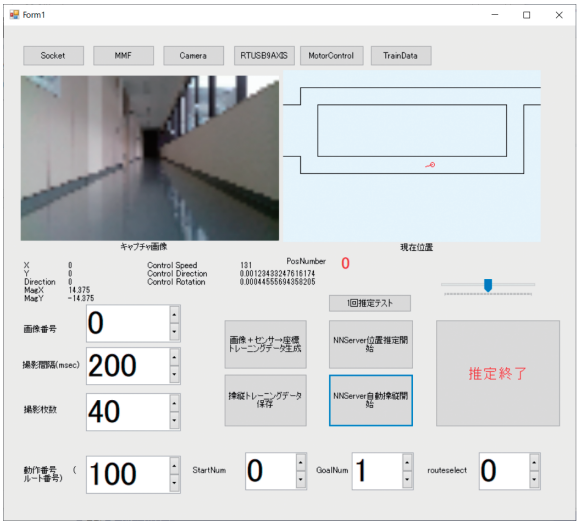

図 3　コントロール画面

3．ネットワーク設計と学習結果

3．1 位置認識用データセット

ディープラーニングの活用事例には画像を提示すると猫や犬、自動車、人などカテゴリを推定する事例が多く報告されている。画像データの細かな特徴をうまくとらえて分類する機能を活用し、画像と方位角から屋内廊下の位置を推定することを試みる。

入力する位置認識用データセットは画像、磁気センサ X 値、磁気センサ Y 値とする。出力値は位置番号とする（表 1）。

表 1　位置認識用データセット

画像データ	磁気センサ X値	磁気センサ Y値	位置番号
pic0-0.png	0.6	-0.095	0
Pic1-0.png	-0.03	-0.555	1
Pic2-0.png	0.8	-0.230	2
Pic79-0.png	0.75	-0.153	79

画像は 50×75 の RGB 画像を用いる。車両前方方向が磁気センサ Y 値、右方向が磁気センサ X 値となり、車両の方位角は -π～π として算出される。ロボットをコントローラで操作し、所定の位置で 360 度回転させながら、画像とその瞬間の二つの磁気センサ値、及び位置番号をデータセットとして csv ファイルに保存した。

屋内位置の推定を例として通路の起点から、1.2 m 間隔で番号を割り当て、各位置で 360 度の画像と磁気センサ値、位置番号からなるデータセットを生成した。AGV の移動経路はビル 2 階の周回経路とし、スタート（位置番号 0）から時計回りに周回してゴール（位置番号 79）までの間で 18,475 個のデータセットを収集した。

3．2 位置推定ネットワーク

次に Neural Network Console でネットワークの設計をおこなった。画像による位置推定はスタートを 0，ゴールを 1 と出力する回帰問題で考えることもできるが、ここでは 0～79 の位置番号のうち 1 か所を選択する分類問題として解くこととした。

畳み込み層 9 層、全結合層 4 層のネットワークを設計し学習させたところ、99.38 % の精度で位置を推定することができた。さらに学習済みのネットワークで未学習のデータを推定したところ、92 % の確率で位置を推定することができた。

3．3 自動操縦用データセット

画像と車両の方位角のみで高精度に位置を推定できることから、この位置推定ネットワークの畳み込み層は十分な環境画像判別ができていると考えられる。そこで、この畳み込み層を生かして、位置に応じたコントローラ操作量を推定できるように転移学習させれば、自動操縦ができると考えた。また、スタートとゴールに一意的な番号を振り、複数ルートの中から選択したルートを自動操縦できるようにした。

AGVへの入力値は画像データ、2つの磁気センサ値、スタート番号（0から9までのどれかを選択）、ゴール番号（0から9のどれかを選択）、ルート番号（-1,0,1から選択）とした。出力値はコントローラから読み取った速度、移動方位角、車両回転速度とする。なお、ニューラルネットに入力するため、各コントローラからの出力値は-1～1に正規化して記録した（表2）。

表2　自動操縦学習用データセット

input	output
X1：画像（50×75RGB），x2：磁気センサX（-1.0～1.0），x3：磁気センサY（-1.0～1.0），x4_0：スタート位置0（0or1），x4_1：スタート位置1（0or1），…，x4_9：スタート位置9（0or1），x5_0：ゴール位置0（0or1），…，x5_9ゴール位置9（0or1），x6：ルート選択番号（-1or0or1）	y1：スピード値（-1～1），y2：平行移動方向値（-1～1），y3：yaw回転速度値（-1～1）

ビル2階の屋内に目標地点を0～4番まで5か所設定し、地点0から地点1、2、3、4までの往路、復路を手動操作し、データを収集した。実験環境に選択した屋内廊下は太陽光が入るため、日差しの強い日中と暗くなる夕方で入力画像に大きな差がある。このため、制御を確実に行わせるには一日を通して網羅的に学習データを収集する必要があった。

また、直線走行時の細かな修正舵を学習させるためには、走行させながら車両に大きな外乱を与えて、その瞬間にコントローラで修正操作を与えて必要な操作データを収集する必要もあった。

学習データを収集し、図4で示すネットワークで学習を行い、得られたネットワークで自動走行実験をしながら、誤操作が発生すれば、リアルタイムでコントローラによる手動修正操作をおこなう。この時の入力データと操作データは追加データセットとして記録される。この作業を繰り返すことで229,811個のデータセットを作成して精度を向上させたが、作成したデータ収集用ツールを活用することで本作業の労力を大幅に低減することができた。

3．4 自動操縦用ネットワーク

位置推定ネットワークの構造を基本として、畳み込み層の最後にスタート、ゴール、ルート番号を結合するように改良した。位置推定ネットワークでは位置番号の分類問題としていたが、自動操縦では、-1～1の推定値を3つ出力する回帰問題とした。

Neural Network Consoleには、基本となるネットワーク構造を設定しておくと、自動的に最適なネットワーク構造を探索する機能が用意されている[2]。最終的に得られた畳み込み層13層、全結合層6層を有するネットワークを図4に示す。

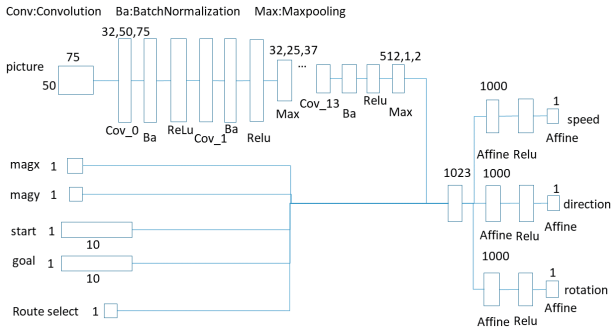

図4　自動操縦用ネットワーク

Neural Network ConsoleのPythonコードのexport機能を用いて学習済みネットワークをPythonコードに変換し、推論サーバーを実装した。C#言語でインタフェースを作成し、インタフェースから入力値をUDP通信によりサーバーに送ると、推論結果を返信するように実装をおこなった。試作したシステムのスループットは5.5 Hzとなった。

自動操縦の例として、地点0から地点1～4への各移動時における、T字路での車両回転速度（Yaw speed）を図5に示す。また各経過時間に対応した入力画像を図6に示す。

このT字路は地点1、2に向かう場合には直進、地点3、4に向かう場合には左折するべき分岐点である。学習データに含まれないデータとして、新たにこのT字路を手動操作で左折走行させて入力データとした。

車両回転速度の値は、AGV直進方向が0、左回転が正、右回転が負の値をとる。

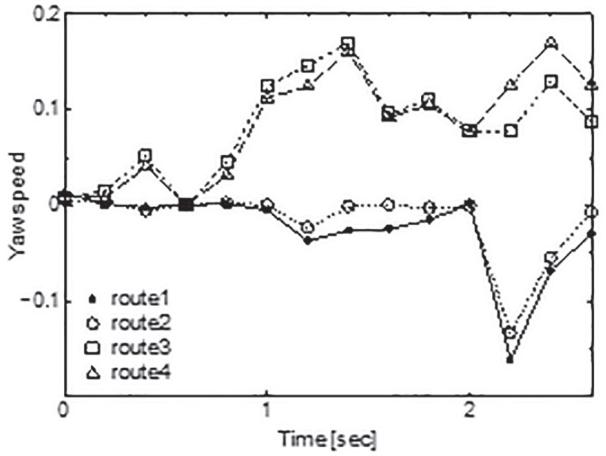

図５　４つの移動経路における車両回転速度の推定結果

図６　検証用入力画像

地点３、４、５をゴールに設定した場合の車両回転速度は 0.8 秒時点から明確に正の値、すなわち左折の推定結果を出力している。

一方、地点１、２をゴールに設定した場合は 0、すなわち直進の制御を行っている。1.2 秒時点からは、AGV 車体が左折方向に回転していることを察知し、直進に修正しようと右に操舵しようとしていることも読み取れる。2.2 秒時点で最大の右回転の出力を行っている。最後の 2.6 秒時点では、向きが大きく左に向いてしまい、事前に学習した入力データの範囲から外れたことで、修正操作が行われなくなっていると考えられる。

以上の結果から、学習したニューラルネットワークがうまく移動制御操作を推定できていることが分かる。各ルートのスタートからゴールへの走行実験においても良好な結果が得られた。また目標地点に到着した時の停止機能についても確認した。地点 1 から地点 4 への移動を指示し、自動走行をおこなわせた結果、目標地点に到達すると速度指令値は減少し、おおよそ 1 m 以内で停止できることを確認した。

４．結　言

オープンソースのニューラルネットワーク設計ツールである Neural Network Console を用いて、ネットワークの設計を行い、画像、車両の方位角、スタート番号、ゴール番号、ルート番号を用いて、メカナム台車を自立的に制御可能なことを確認した。

試作した AGV は、従来手法のように内部で地図を作成することなく、自動的に選択した経路を走行して目標地点に到達することができる。

今回の AGV 試作によって、ディープラーニングの汎用性を実感できた。今後はエレベータを活用した多層階間での移動や、指定された物資を自動で搬送するシステムなどより複雑な自立走行への応用を検討する。また、今回得られたノウハウを活用し、県内企業からの要望が多い産業機械の自動化への技術支援にも活用していきたい。

参考文献

[1] 小林由幸、“AI のもたらす産業の変革とソニーの取り組み事例”、http://www.tec-lab.pref.gunma.jp/info/download/files/2018ai-02.pdf、2018.

[2] 小林由幸、“Neural Network Console 活用テクニックのご紹介”、https://www.slideshare.net/Sony_Neural_Network_Console_Libraries/20171206-sony-neur alnetworkconsole、2017.