

令和6年 3月 7日

令和5年度 学生自主研究成果報告書

教 育 本 部 長 様

学生自主研究グループ名	ロボ班	
研究課題名	ロボットの遠隔操作データを用いた機械学習	
研究代表者 (学生)	学籍番号	B25N008
	氏 名	植 松 夕 奈
指導教員	学 科	知能メカトロニクス学科
	氏 名	伊 藤 亮

学生自主研究の報告書を別紙のとおり提出します。

ロボットの遠隔操作を用いた機械学習

システム科学技術学部 知能メカトロニクス学科

1年 植松 夕奈

指導教員 システム科学技術学部 知能メカトロニクス学科

准教授 伊藤 亮

学生支援スタッフ システム科学技術学部 知能メカトロニクス学

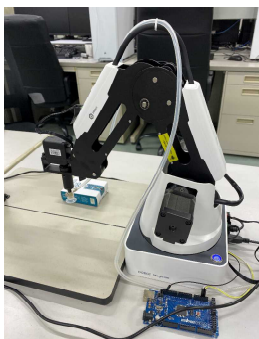
4年 小堀 滉斗

1. はじめに

近年は機械学習や人工知能などの科学技術が発展している。現代の社会人がかかえる課題の一つとして、時間を効率的に使うために自動化を可能にすると考え、日常生活における単純作業をロボットに学習させたいと考えた。そのために、ロボットの遠隔操作システムを作って遠隔操作で入力したデータを取り、取ったデータをAIに機械学習させて、AIでロボットを動かす。それによって、ロボットの動きを労力なく学習できるのではないかと考える。

2. 遠隔操作データの蓄積対象とするタスク

この研究で行うタスクは、右側にある物体を左側に移動させることである。使用するロボットは DOBOT Magician (写真1,2) である。DOBOT Magician の操作として平面上での右, 左, 上, 下と, 高さとしての上, 下と, ロボットの先端にある吸盤の吸引と放出という8つの操作ができるようにプログラミングで設定した。ロボットの上に設置したカメラの画像情報 (写真3) から機械学習をもとに8つの操作を自動で判断させる。右側にある物体の上にロボットの先端部分が来るように操作し, 下がって吸引し物体をつかみ, 上がって左側へ操作して, 下がって放出して物体を放してから上がるという一連の操作によってタスクを自動で行えるようにする。ただし, カメラはロボットの上に設置してありロボットは高さとしての上下, 吸引と放出がわからないため, カメラ画像の左下には下がっている場合は赤, 下がっていない場合は白の「DOWN」, 右下には吸引している場合は赤の「ON」, 吸引しておらず放出している場合は白の「OFF」という文字を示している。



DOBOT Magician の全体像 (写真1)



DOBOT Magician を上から見た
ときの様子 (写真2)



カメラ画像の一例 (写真3)

3. 遠隔操作データの蓄積

今回の研究で機械学習を行うには、画像とその画像に対しどう操作をしたかというデータが多く必要となるため、一連の動作のデータを蓄積する必要がある。物体を右側の様々な場所に置いて、カメラ画像から判断して操作することを何回も行うことによって、画像と操作のデータを蓄積し

た。ただし、ロボットと接続されているパソコンで操作するのではなく、別のパソコンから遠隔操作することによってカメラ画像の「パス」と操作の「タグ」のデータの蓄積を行った。

一連の動作における右や左などのそれぞれの操作の数には偏りがある。数が偏ったまま機械学習をすると、多い数の操作を行えば正解率が上がると学習してしまうため、それぞれの操作が公平である状態での機械学習を行うことができない。そのため、機械学習用にそれぞれの操作の数を同じ数になるようにそろえた。また、機械学習の前に学習用と評価用というデータセットに分けるため、それぞれのデータセットで操作の数が偏らないようにシャッフルした（画像 1）。

1	path	tag
2	./manipulation_image/image_487.png	1
3	./manipulation_image/image_4665.png	3
4	./manipulation_image/image_3051.png	5
5	./manipulation_image/image_2726.png	4
6	./manipulation_image/image_5137.png	1
7	./manipulation_image/image_6287.png	6
8	./manipulation_image/image_3817.png	2
9	./manipulation_image/image_4353.png	3

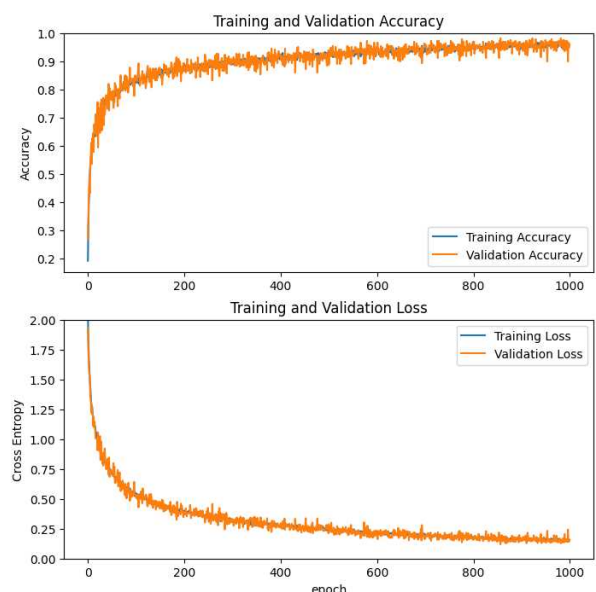
カメラ画像の「パス」と操作の「タグ」
のデータの様子(画像 1)

4. 転移学習について

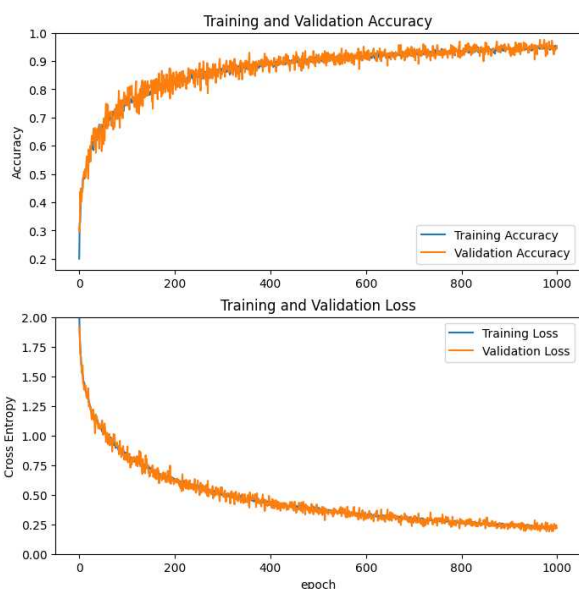
転移学習という機械学習の 1 つで、既に学習済みのベースモデルを呼び出し、学習済みの大部分の層は学習したまま固定して、最後の数層だけ学習するという方法を使った。転移学習の利点であるデータ数が少なくても高い学習精度を目指すことが出来ること、短時間で学習を行うことが出来ることが今回の研究で適していたため、転移学習を用いた。機械学習のライブラリである TensorFlow 上で、Keras の MobileNet というニューラルネットワークを用いて学習を行った。偏りをそろえた 1600 程度のデータをそのまま使って学習することよりもさらにノイズに対応することが出来るように、RandomRotation(0.01)で角度が傾いた画像、RandomBrightness(0.002)で明るさを変えた画像のデータを追加しデータ拡張した。データ拡張後、学習用と評価用のデータセットを 8:2 の割合で作成するように設定する。高い精度の学習が出来るように 1000 エポックで学習させた。ニューラルネットワークの違いによって学習の結果が変わるのか検証するために MobileNet 以外のニューラルネットワークである NASNetMobile と EfficientNet を用いた学習も行った。

5. 転移学習の結果

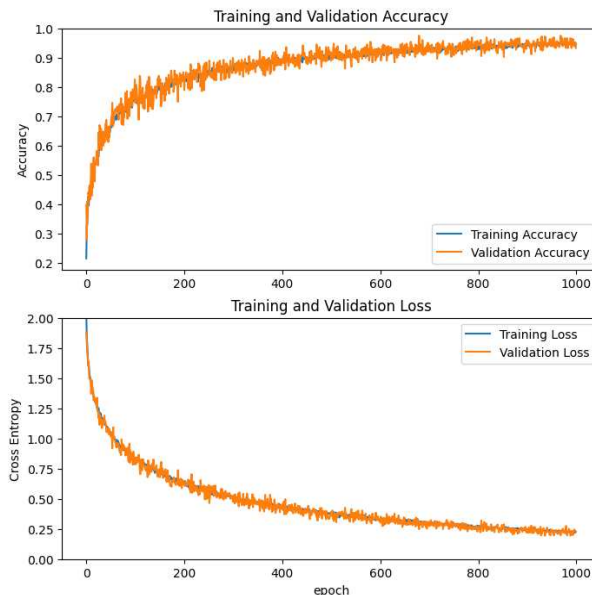
MobileNet での学習の 1000 エポック目の結果は accuracy が 0.9586, val_accuracy が 0.9602 となった。NASNetMobile での学習の 1000 エポック目の結果は、学習用データでの正解率である accuracy は 0.9456, 評価用データでの正解率である val_accuracy は 0.9541 となった。EfficientNet での機械学習の 1000 エポック目の結果は accuracy が 0.9241, val_accuracy が 0.9021 となった。今回の研究におけるニューラルネットワークとしては、精度が最も良い MobileNet が一番適していると考えられる。



MobileNet の学習結果 (グラフ 1)



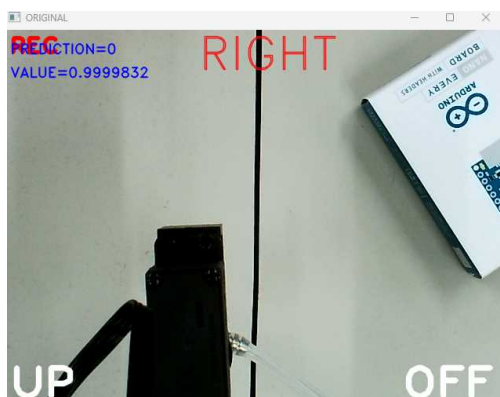
NASNetMobile の学習結果 (グラフ 2)



EfficientNet の学習結果 (グラフ 3)

6. 実装

精度が良い MobileNet で機械学習をして保存したモデルを読み込み、学習と同じ表示に設定したカメラ画像から推論し、推論で一番大きな値であるタグの操作を行う Python のプログラミングを実行することによって実装した。



学習と同じ表示を追加し、推論で一番大きな値も表示したカメラ画像

7. 実装の結果

学習では 9 割以上の成果を得られたのに対し、実装では理想としてある左側まで物体を運ぶという一連の動作を自動で行うことが出来なかった。上手く動作することが出来なかったパターンとして主に 2 つある。

1 つ目は最初の時点で右と左の操作を繰り返すなどの 2 つの操作を繰り返してしまうパターンである。まず、最初の時点で右と左の操作を繰り返すことの詳細である。最初はプログラミングで `dobot.initiate()` を使ってホームポジションに戻るため、最初の時点とは、右側と左側の目安になる真ん中の線の上にロボットのアームがきている状態である。右に操作する時にどのくらい確信して操作を行ったかという推論は、毎回 0.99 や 0.98 などの高い値であった。左に操作する時の推論の表示では約 0.5 という 5 割ほどの値を示している場合もあれば、約 0.8 という 8 割ほどの高い値を示している場合もあった。このような推論の値によって右の操作を行った後、左の操作を行い、また右の操作を行うという右と左の操作の繰り返しがパターンとして見られた。この最初の時点で右と左を繰り返すこと他には、順調に物体の近くまで操作出来た後に平面上の上と下を繰り返すこと、順調に物体のところで下がった後に吸引と放出を繰り返すことがあった。

2つ目は、物体の近くまで操作することが出来た後に物体のある場所から少し離れた場所で下に行き吸引を行ってしまうパターンである。推論の表示は約0.8という8割ほどの高い確信のある値を示している場合が多かった。

8. 実装の考察

実装の結果上手く動作することが出来なかった主なパターンの2つについて原因を考察する。

まず1つ目の、最初の時点で右と左の操作を繰り返すなどの2つの操作を繰り返してしまうパターンである。このような動作になった原因は、機械学習が上手く行えていなかったためではないかと考える。最初に考えたのは、実装のプログラミングに欠陥があるのではないかとということである。そこで、推論の値は表示させながら操作は手動で行うプログラミングを作り調べた。すると手動で操作を行っても推論は同じく右と左を繰り返したため、プログラミングの問題ではないとわかった。右と左を繰り返さなかった場合どうなるか調べるために推論を無視して物体のあるところで下がる操作をすると、吸引をして上がり左側へ移動し下がってから放出して上がるという理想の操作の値が一番大きい推論が続いた。そのため、一連の操作において物体の上で下がる事が出来れば上手く操作できるが、物体までの操作の学習が上手くいっていないのではないかと考えられる。そのため次に考えたことは、物体の初期位置が真ん中に近かったために、右側にある物体をつかみに行く過程と物体をつかんでいて左に持って行く過程のどちらであるか判断できなかったのではないかとということである。しかし、吸引をしている時には判断する画像に「ON」という文字が赤くなっていて、吸引していない時には画像に「ON」という文字が白くなっているため、学習が上手く出来ていれば物体を運んでいるかどうかはわかると考えられる。実際に物体のところで手動で下がった場合は推論によって吸引をして左に操作することが出来ると分かった。よってこのパターンが起こってしまう原因として考えられることは、物体を持った場合では上手く学習が出来ているが、物体を持っていない場合では上手く学習が出来ていないということである。

次に2つ目の、物体の近くまで操作することが出来たが、物体のある場所から少し離れた場所で下に行き吸引を行ってしまうパターンである。このような動作になった原因も、機械学習が上手く行えていなかったためではないかと考える。最初に考えたことはカメラ画像が遅延していてパソコンに表示されるのが遅かったため、ロボットの操作も遅延のせいでずれているのではないかとということである。しかし `time.sleep` をプログラミングで入れることによって遅延を少なくすることが出来たが、変わらずずれた場所で下がってしまったため、遅延が原因ではないと分かった。そしてこれも1つ目のパターンと同じで物体をつかむまでの過程であるため、物体をつかむまでの機械学習がうまくいっていないのではないかと考えられる。

2つの上手く操作できなかったパターンから、物体をつかむまでの学習が上手くいっていないのではないかと考えられる。これを解決するための方法として機械学習の見直しが考えられる。

9. まとめ

実際にデータを取り、MobileNet というニューラルネットワークを用いた転移学習を行った。学習の結果では1000エポックという多いエポック数で学習することで高い精度を目指すことが出来た。しかしながら、ロボットへ機械学習の結果を上手く実装することが出来なかった。原因としては物体をつかむまでの過程での学習が上手くいっていないためではないかと考えられる。今後は機械学習の改善によってロボットへの実装が出来るようにしたい。