

2023年6月1日(木)

会場C

OS30 人工知能およびソフトコンピューティングとその近傍領域

[C-08] OS30 人工知能およびソフトコンピューティングとその近傍領域 (1)

座長:入江 寿弘(日本大学)

13:15 ~ 14:15 会場C (2F 中会議室 201A)

[C-08-01] 水難救助用ドローンへの搭載を目的とした深層学習モデルのエッジデバイスへの実装

*三ツ本 大輝¹、小林 伸彰¹ (1. 日本大学)

13:15 ~ 13:30

[C-08-02] 運動想起時脳波の深層学習による分類に関する研究

*柳生 遥人¹、小林 伸彰¹ (1. 日本大学)

13:30 ~ 13:45

[C-08-03] 画像想起時脳波の分類に関する研究

*根本 渉¹、小林 伸彰¹ (1. 日本大学)

13:45 ~ 14:00

[C-08-04] 物体検出用深層学習モデルのIoT向けエッジデバイスへの実装と性能評価

*藤本 彩斗¹、小林 伸彰¹ (1. 日本大学)

14:00 ~ 14:15

OS30 人工知能およびソフトコンピューティングとその近傍領域

[C-09] OS30 人工知能およびソフトコンピューティングとその近傍領域 (2)

座長:小林 伸彰(日本大学)

14:30 ~ 15:30 会場C (2F 中会議室 201A)

[C-09-01] ライドシミュレータ

*西川 智紀¹、松谷 拓実¹、入江 寿弘¹ (1. 日本大学)

14:30 ~ 14:45

[C-09-02] 楕円型展開車輪を用いた階段昇降ロボットの試作—実大機構モデルの作成と制御方法の検討—

*加茂 雄太¹、入江 寿弘¹ (1. 日本大学)

14:45 ~ 15:00

[C-09-03] 展開車輪を用いた階段昇降ロボット-深層強化学習による階段昇降の学習-

*青木 颯大¹ (1. 日本大学)

15:00 ~ 15:15

[C-09-04] データセット作成における自己教師あり学習の有効性の検討

*塩崎 雄晴¹、入江 寿弘¹、小林 伸彰¹、新宮 清志¹ (1. 日本大学)

15:15 ~ 15:30

OS30 人工知能およびソフトコンピューティングとその近傍領域

[C-08] OS30 人工知能およびソフトコンピューティングとその近傍領域 (1)

座長:入江 寿弘(日本大学)

2023年6月1日(木) 13:15 ~ 14:15 会場C (2F 中会議室 201A)

[C-08-01] 水難救助用ドローンへの搭載を目的とした深層学習モデルのエッジデバイスへの実装

*三ツ本 大輝¹、小林 伸彰¹ (1. 日本大学)

13:15 ~ 13:30

[C-08-02] 運動想起時脳波の深層学習による分類に関する研究

*柳生 遥人¹、小林 伸彰¹ (1. 日本大学)

13:30 ~ 13:45

[C-08-03] 画像想起時脳波の分類に関する研究

*根本 渉¹、小林 伸彰¹ (1. 日本大学)

13:45 ~ 14:00

[C-08-04] 物体検出用深層学習モデルのIoT向けエッジデバイスへの実装と性能評価

*藤本 彩斗¹、小林 伸彰¹ (1. 日本大学)

14:00 ~ 14:15

13:15 ~ 13:30 (2023年6月1日(木) 13:15 ~ 14:15 会場C)

[C-08-01] 水難救助用ドローンへの搭載を目的とした深層学習モデルの
エッジデバイスへの実装

*三ツ本 大輝¹、小林 伸彰¹ (1. 日本大学)

13:30 ~ 13:45 (2023年6月1日(木) 13:15 ~ 14:15 会場C)

[C-08-02] 運動想起時脳波の深層学習による分類に関する研究

*柳生 遥人¹、小林 伸彰¹ (1. 日本大学)

画像想起時脳波の分類に関する研究

Study on Classification of EEG during image recall

根本 渉¹⁾, 小林 伸彰²⁾

Wataru Nemoto, and Nobuaki Kobayashi

1) 日本大学理工学研究科精密機械工学専攻 大学院生 (〒274-8501 千葉県船橋市習志野台7-24-1, E-mail: cswa23028@g.nihon-u.ac.jp)

2) 博(工) 日本大学理工学部 准教授 (〒274-8501 千葉県船橋市習志野台7-24-1, E-mail: kobayashi.nobuaki@nihon-u.ac.jp)

In Japan, the birthrate is rapidly declining and the population is rapidly aging. Against this backdrop, a shortage of caregivers for the elderly is becoming an issue. In addition, some people in need of care who are physically unable to move themselves also want to live their lives without inconveniencing. BMI (Brain Machine Interface), which measures neural activity in the brain and operates machines based on the signals obtained, is attracting attention as a solution to these problems. There are BMIs that use various stimuli such as P300 and motor imagery. In this study, I focused on BMIs that use image recalling and achieved high average accuracy of 78.7% in three-class classification.

Key Words : EEG, CNN, brain computer interface.

1. はじめに

現在、日本では少子高齢化が進んでおり、2021年時点で高齢者(65歳以上)の人口は3640万人で総人口の29.1%で過去最多で世界で最も高くなっている。この割合は2065年には38.4%になると見込まれている。こうした背景から高齢者の介護者の人手不足が問題視されている。また、体が動かせない要介護者自身も、他人に迷惑をかけずに生活をしたいという人もいる。こうした問題の解決策として、脳内の神経活動を測定し得られた信号により機械を操作するBMI(Brain Machine Interface)が注目を集めている。BMIは体を動かさず機械を操作できるため、要介護者自身での機器の使用や、周りとのコミュニケーションが可能になるのではないかと考えられており、実際に色刺激を利用して車椅子を操作する研究などがある。[1]

P300、運動想起など様々な刺激を利用したBMIがあり、その中でも本研究では画像想起を利用したBMIに着目した。形や色を認識する際の脳波を分類することが可能なことが知られており[3][4]、画像想起時の脳波も分類可能であると考えられる。今回は事前に準備した画像を指定し、それを想起してもらうことで、基準となる刺激を設けてそれと同じものを想起させることによって被験者が想起しやすくなり、脳波の変化をつかみやすくなった。また、画像を見るだけで済むのでSSVEPなどと違い視覚呈示装置が不要なため、事前の準備が簡単なほか、想起をすることによって脳に刺激を与え記憶を思い出すトレーニングにもなるため、認知症予防にもつながるといった利点もある。

本研究では画像想起時の脳波を測定し、EEGNet[2]という深層学習モデルによって学習を行い、画像想起時の脳波の分類を行った。

2. 実験方法

脳波の測定には8チャンネルまで測定可能なGuger Technology社製のg.Nautilus、電極には同じくGuger Tehnology社製のg.SAHARA Active electrodeを使用した。画像の認識や記憶の呼び出しは側頭葉で行われることや、色彩情報の把握が後頭葉で行われることから、電極は国際10-10法に準拠し、図1のようにT7,TP7,P7,O1,O2,P8,TP8,T8に配置した。グランド電極は左乳様突起、リファレンス電極は右乳様突起とした。脳波は250Hzでサンプリングした。5-60Hzのバンドパスフィルタと商用電源を取り除くために48-52Hzのノッチフィルタを適用した。被験者は事前に同意を得た4人の男女(21-22歳)で、いずれの被験者も既知の神経学的障害や重大な健康上の問題を有していない。

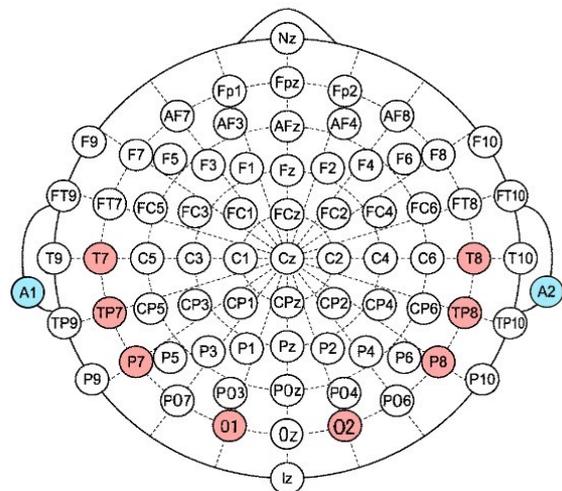


図1 国際10-10法に基づいた電極配置

次の実際の実験パラダイムについて示す。

- (1) 被験者を楽な姿勢で椅子に座らせる。
- (2) 2秒間マスク画像を表示する。
- (3) 2秒間ネコ、ウサギ、ゾウの中からランダムな画像を1枚表示する。
- (4) 3秒間再びマスク画像を表示する。
- (5) 3秒間黒い画面を表示し、(3)で表示された画像を想起する。
- (6) (2)-(5)の手順を10回繰り返す。
- (7) 約1分の休憩を与える。
- (8) (2)-(7)を1セットとし、これを10セット行い、計100試行測定する。

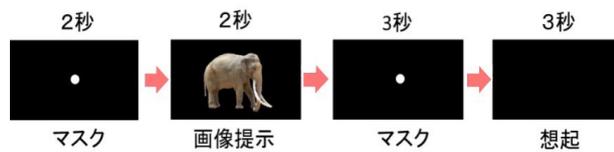


図 2 画像の表示順序

実際の実験の様子を図3に示す。



図 3 実験時の様子

上記の手順で得られたデータをEEGNetによって学習、分類する。EEGNetとはEEGの分類に特化している比較的軽量のCNN(Convolution Neural Network)のことでSeparable Convolution、Depthwise Convolution層を有している。入力データは1秒間(8x250)とする。単純に3秒間のデータを三分割すると学習データが不足するので画像想起部分の脳波データを1秒のデータを0.1秒ずつずらしてデータ拡張を行った。そのうち7割のデータを学習用データ、3割のデータをテストデータとし学習、分類を行った。また、エポック数は200とした。

3. 結果

被験者1-4の分類結果を図4-7に、すべての被験者の結果と平均をまとめたものを表1に示す。

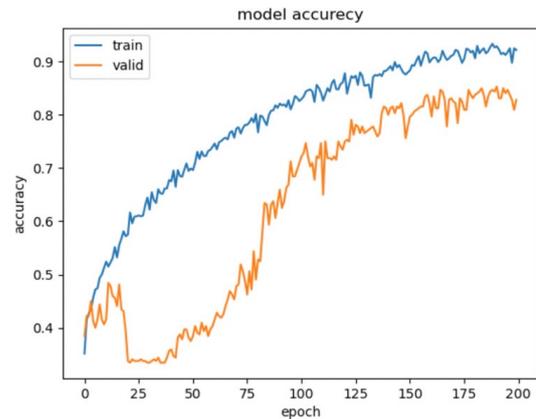


図 4 被験者 1 の分類結果

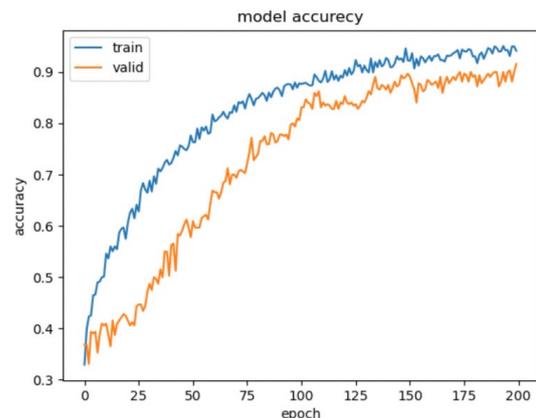


図 5 被験者 2 の分類結果

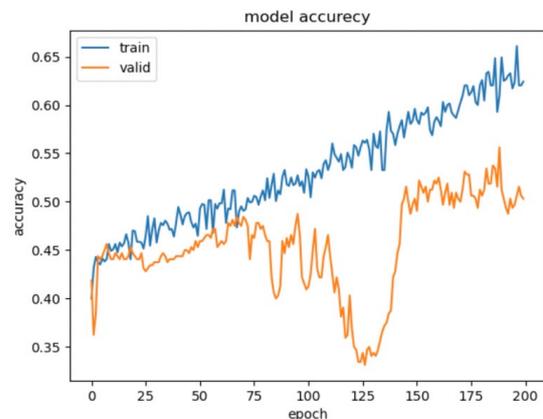


図 6 被験者 3 の分類精度

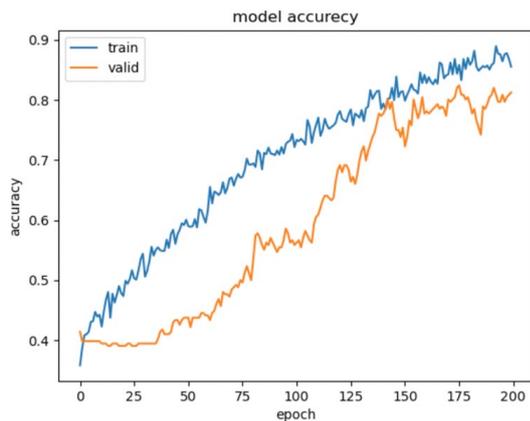


図 7 被験者 4 の分類精度

表 1 各被験者の分類精度と平均

被験者	1	2	3	4	平均
精度[%]	90.2	92.5	54.2	77.8	78.7

4. 考察

EEGNetによって画像想起時の脳波の分類を行ったところ、平均で78.7%という高い精度で分類することができた。中でも被験者4人のうち2人が90%以上という非常に高い精度を出しており、画像想起時の脳波の分類にEEGNetが非常に有効であるとわかる。

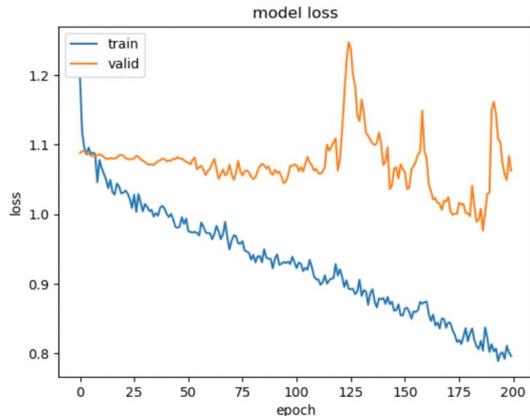


図 8 被験者 3 の損失

しかし、図8からわかるように被験者3は全体的に精度が低いことに加えて、グラフが途中で大きく下降している。図9に被験者3の学習、推論時の損失のグラフを示す。図からは後半部分での学習時の損失の減少と同時に損失の増大の傾向があり、過学習が起きている可能性が示唆されている。そのほかの要因としては、データ数の不足、騒音などにより被験者の注意が他に向いてしまったことなどが考えられる。

今回は3種類の画像で分類を試みたが、分類数を増やした際にも高い精度を維持できるかが今後の課題としていきたい。

参考文献

- [1] Nafiu Hasan, Md, Mahmudul Hasan, Md. Akramul Alim: Design of EEG Based Wheel Chair by Using Color Stimuli and Rhythm Analysis, International Conference on Advances in Science, Engineering and Robotics Technology, pp. 1-4, 2019.
- [2] Vernon J. Lawhern, Amelia J. Solon, Nicholas R. Waytowich, Stephen M. Gordon, Chou P. Hung, and Brent J. Lance: EEGNet: A Compact Convolutional Neural Network for EEG-based Brain-Computer Interfaces, Journal of Neural Engineering, Vol. 15, No. 5, 2018.
- [3] Daniel Zakzewski, Ismail Jouny, and Yih Choung Yu: Statistical Features of EEG Responses to Color Stimuli, Statistical features of EEG responses to color stimuli, Annual Northeast Bioengineering Conference, pp. 1-4, 2014.
- [4] Monalisa Pal, Anwesha Khasnobish, Amit Konar, D. N. Tibarewala, R. Janarthanan: Performance Enhancement of Object Shape Classification by Coupling Tactile Sensing with EEG, International Conference on Electronics, Communication and Instrumentation, pp. 1-4, 2014.

物体検出用深層学習モデルの IoT向けエッジデバイスへの実装と性能評価

Implementation and performance evaluation of deep learning models for
object detection in edge devices for IoT

藤本 彩斗¹⁾, 小林 伸彰²⁾

Ayato Fujimoto and Nobuaki Kobayashi

1) 日本大大学院 精密機械工学専攻 大学院生 (〒274-8501 千葉県船橋市習志野台7-24-1, E-mail: csay23033@g.nihon-u.ac.jp)

2) 博(工) 日本大学 理工学部 准教授 (〒274-8501 千葉県船橋市習志野台7-24-1, E-mail: kobayashi.nobuaki@nihon-u.ac.jp)

A method of rescuing people by using object detection with a camera mounted on a drone, for example, is being considered. However, single-board computers such as edge devices that can be mounted on small and lightweight drones have various limitations, such as power consumption and area size. In this study, we examined the differences in performance depending on the type of CPU (Central Processing Unit) or GPU (Graphics Processing Unit) used in the SoC (System on a chip). We used three commercially available edge devices: the Raspberry Pi 4 Model B, the Jetson Nano 2GB developer's kit, and the Jetson AGX Xavier developer's kit. Object detection was performed using these three models, and edge devices suitable for installation on a drone were selected from the viewpoints of power consumption and inference speed. YOLOv5, a deep learning-based model, was used for object detection.

Key Words : Edge devices, single board computers, deep learning, YOLOv5

1. 緒言

近年, 人命救助や物流にドローンなどの小型軽量なエッジデバイスを用いるための研究が行われている。エッジデバイスとは, 末端の端末で取得したデータをそのままクラウドに送るのではなく, 末端の端末上で必要な情報処理を行い, クラウドや通信回線に対する負荷を低減するエッジコンピューティングを用いる機器のことである。例として, スマートフォンや家電製品, 自動車, 産業用機械などがあげられる。その中でもIoT(Internet of Things)向けエッジデバイスには, 用途によって様々な制限が存在する。

今回はドローンに着目する。ドローンを用いる際の制約条件の筆頭として挙げられるのは消費電力や大きさ(サイズ)である。消費電力, 大きさはどちらも活動可能時間に影響を与え, 必要十分な処理能力を有しながら消費電力, 大きさを抑える必要がある。そこで, ドローンに搭載するシングルボードコンピュータとしてRaspberry Pi 4 Model B, Jetson Nano 2GB開発者キット, Jetson AGX Xavier開発者キットの比較を行う。これらは, 搭載されているCPU(Central Processing Unit), GPU(Graphics Processing Unit)の違いから選定した。既存のドローンに搭載することを想定するためOSはUbuntuとし, これに物体検出モデルであるYOLOv5を実装する。YOLOv5を用いた推論には

カメラから取得した映像や, 既存の画像, 映像を用いることができる。本研究では, カメラの性能や取得した映像のフレームレートなどの影響を取り除くため, COCOデータセットの画像を用いる。ドローンに搭載した際のことを考慮しCUI環境, SSH接続にて推論速度および推論時の消費電力を測定する。これにより, 本研究では, 物体検出モデルをIoT向けエッジデバイスに実装した際の性能評価を行うことを目的としている。

2. 物体検出用深層学習モデルについて

使用する物体検出用深層学習モデル, YOLOは2015年に公開された物体検出モデルである。[1]

本研究で用いるYOLOv5は, YOLO, YOLOv2, v3, v4の後継に当たり, 2020年に公開されたものである。[2~5]

3. 実験

(1) 実験機器

本実験では, 次の機器を用いた。実際にドローン等に搭載する際の環境に近づけるため, CUI環境, ssh接続を用いて測定を行う。

- ・ Raspberry Pi 4 Model B
- OS : Ubuntu 22.04.1 LTS
- SoC : Broadcom BCM2711
- メモリ容量 : 4GB

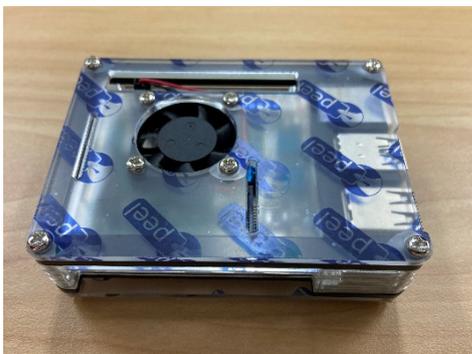


図-1 Raspberry Pi 4 Model B

- ・ NVIDIA Jetson Nano 2GB 開発者キット
- OS : Jetpack R32 7.3(Ubuntu 20.04.5 LTS)
- SoC : Tegra X1
- メモリ容量 : 2GB

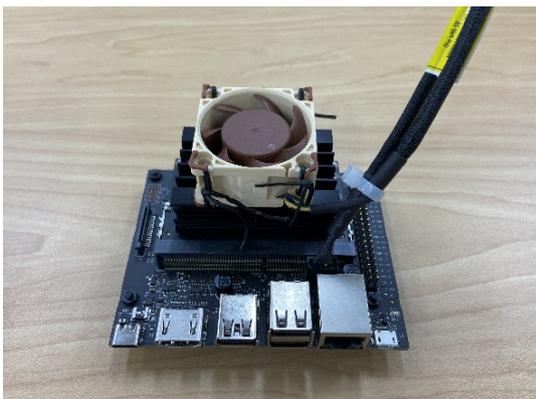


図-2 NVIDIA Jetson Nano 2GB 開発者キット

- ・ NVIDIA Jetson AGX Xavier 開発者キット
- OS : Jetpack R35 1.0(Ubuntu 20.04.5 LTS)
- SoC : Xavier
- メモリ容量 : 32GB



図-3 NVIDIA Jetson AGX Xavier 開発者キット

- ・ USB電流電圧テスター
- ATORCH UD18



図-4 ATORCH UD18

- ・ データセット
- Microsoft COCO 2017 [6]
- val2017.txtに記載されているリストより上部1000枚を用いた。
- ・ 学習モデル
- YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x
- 学習モデルは参考文献[5]よりダウンロードし、使用した。

(2) 測定

USBテスターを各エッジデバイスに接続し、COCOデータセットおよび学習モデルYOLOv5s, YOLOv5m, YOLOv5l, YOLOv5xを用いてdetect.pyを実行した際の消費電力を測定した。また、推論に用いた時間を取得し、推論速度と消費電力から比較を行う。

本研究では時間測定の簡易化のため、detect.pyに対して次のように追記した。

```

30
31 import argparse
32 import os
33 import platform
34 import sys
35 from pathlib import Path
36
37 import torch
38
39 #推論速度の計測用に追加
40 import time
41
42 FILE = Path(__file__).resolve()
43 ROOT = FILE.parents[0] # YOLOv5 root directory
44 if str(ROOT) not in sys.path:
45     sys.path.append(str(ROOT)) # add ROOT to PATH
46 ROOT = Path(os.path.relpath(ROOT, Path.cwd())) # relative
47
  
```

図-5 追記箇所

```

119 # Run inference
120 model.warmup(imgsz=(1 if pt or model.triton else bs, 3, *imgsz)) # warmup
121 seen, windows, dt = 0, [], (Profile(), Profile(), Profile())
122
123 #計測開始
124 time_start = time.time()
125
126 > for path, im, im0s, vid_cap, s in dataset:
127
128     #計測終了
129     time_end = time.time()
130     #計算
131     time_result = time_end - time_start
132     #推論速度を出力。アンダーバーの後に追記すること。
133     print(time_result)
134     with open('yolov5_inf_time.csv', 'a') as t:
135         t.write(f'{str(time_result)}\n')

```

図-6 追記箇所

4. 結果

(1) 推論速度

Jetson Nano 2GBではメモリ不足によりYOLOv5xを実行することができなかつたため、測定不能とした。

各エッジデバイスの推論速度を表-1に、1枚当たりの推論速度を図-7に示す。

本結果より、Jetson AGX Xavier が最も推論速度が速く、次いでJetson Nano 2GB、最後にRaspberry Pi 4 Model Bとなることがわかる。

表-1 各エッジデバイスの推論速度 (n=1000) [s]

モデル	Raspberry Pi 4 Model B	Jetson Nano 2GB	Jetson AGX Xavier
YOLOv5s	1879.87	189.938	63.9607
YOLOv5m	3648.89	376.579	111.8133
YOLOv5l	6232.01	631.417	175.3225
YOLOv5x	10098.74	-	290.7200

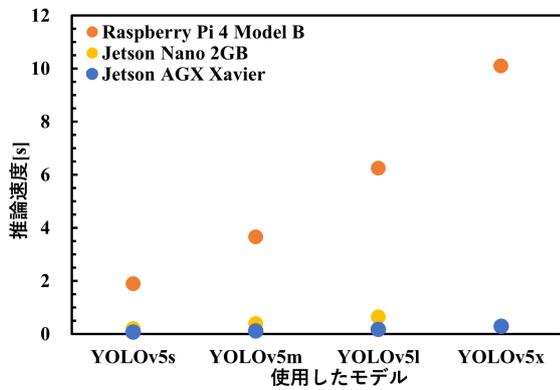


図-7 各エッジデバイスの1枚当たりの推論速度

(2) 推論時の消費電力

推論時の消費電力を測定した結果を表-2、図-8に示す。本結果より、Jetson AGX Xavier が最も推論時の消費電力が高く、Jetson Nano 2GBとRaspberry Pi 4 Model Bは僅差であることがわかる。

表-2 推論時の消費電力

モデル	Raspberry Pi 4 Model B	Jetson Nano 2GB	Jetson AGX Xavier
YOLOv5s	7.90	8.024	17.2394
YOLOv5m	8.10	8.653	20.6703
YOLOv5l	8.39	9.066	20.7778
YOLOv5x	8.54	-	22.3897

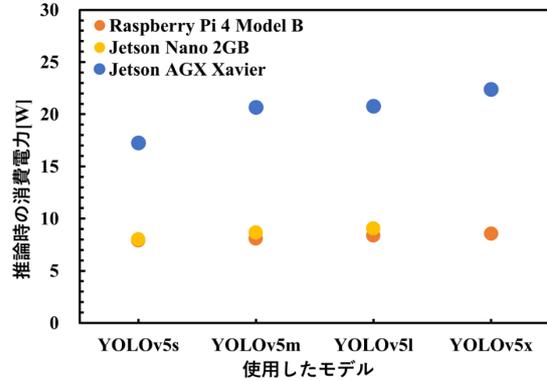


図-8 推論時の消費電力

(3) 推論時の消費電力

以上の結果より各エッジデバイスの推論時の消費電力と推論速度の関係を図にすると図-9のようになる。

Jetson AGX Xavier が最も推論時の消費電力が高く、Jetson Nano 2GBとRaspberry Pi 4 Model Bは僅差であり、Jetson AGX XavierとJetson Nano 2GBの推論速度は僅差となることがわかる。

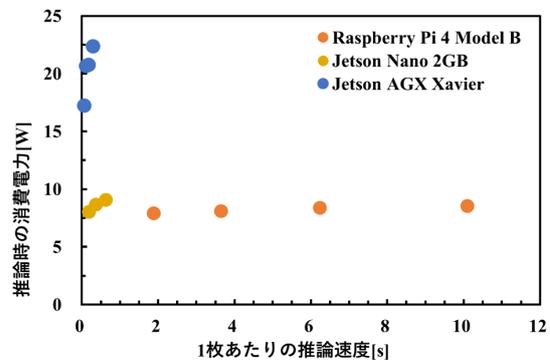


図-9 推論時の消費電力と推論速度の関係

5. 考察

本研究では、YOLOv5を用いてRaspberry Pi 4 Model B, Jetson Nano 2GB, Jetson AGX Xavierの性能評価を行った。

図-7より、推論速度はJetson Nano 2GB, Jetson AGX Xavierが優れている結果となった。また、Jetson Nano 2GBがメモリ不足でYOLOv5xを実装することができないことが分かった。これは、YOLOv5xが最も大きいモデルであるためであるが、量子化を行うことでモデルを縮小化すれば実装することが可能となるのではないかと推測される。

図-8より、推論速度はJetson Nano 2GB で必要十分であると考えられ、消費電力が大きいため、Jetson AGX Xavier はYOLOv5を用いた物体検出を行う上では過剰な性能であると考えられる。また、Raspberry Pi 4 Model B とJetson Nano 2GBの消費電力は同等であることがわかる。GPUを搭載していないRaspberry Pi 4 Model Bは推論速度が非常に遅いため、障害物を検出するなどの検出速度が求められるケースにおいては使用を避ける必要があると考えられる。

以上より、本研究で物体検出用深層学習モデルYOLOv5を実装した三機種において適しているエッジデバイスは、Jetson Nano 2GBであると考えられる。

6. 結言

本研究では、物体検出用深層学習モデルであるYOLOv5をRaspberry Pi 4 Model B, Jetson Nano 2GB, Jetson AGX Xavierに実装し、性能評価を行った。その結果、三機種においては Jetson Nano 2GBがYOLOv5を実装するのに最も適しているという結論が得られた。

7. 今後について

本研究はCPU, GPUの違いが物体検出用深層学習モデルの推論にどの程度影響を及ぼすかを改めて確認するものである。今後FPGAにYOLOv5を実装し、比較を行う際の指標として本研究の結果を用いたいと考える。

参考文献

- [1] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, “You Only Look Once: Unified, Real-Time Object Detection”, arXiv preprint, arXiv: 1506.02640 (2016)
- [2] Joseph Redmon, Ali Farhadi, “YOLO9000: Better, Faster, Stronger”, arXiv preprint, arXiv:1612.08242 (2016)
- [3] Redmon, Joseph, Ali Farhadi, “YOLOv3: An incremental improvement.”, arXiv preprint, arXiv: 1804.02767 (2018)
- [4] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan, Mark Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection”, arXiv preprint, arXiv:2004.10934 (2020)
- [5] YOLOv5, <https://github.com/ultralytics/yolov5>
- [6] COCOデータセット, <https://cocodataset.org/#home>

OS30 人工知能およびソフトコンピューティングとその近傍領域

[C-09] OS30 人工知能およびソフトコンピューティングとその近傍領域 (2)

座長:小林 伸彰(日本大学)

2023年6月1日(木) 14:30 ~ 15:30 会場C (2F 中会議室 201A)

[C-09-01] ライドシミュレータ

*西川 智紀¹、松谷 拓実¹、入江 寿弘¹ (1. 日本大学)

14:30 ~ 14:45

[C-09-02] 楕円型展開車輪を用いた階段昇降ロボットの試作 ー実大機構モデルの作成と制御方法の検討ー

*加茂 雄太¹、入江 寿弘¹ (1. 日本大学)

14:45 ~ 15:00

[C-09-03] 展開車輪を用いた階段昇降ロボット-深層強化学習による階段昇降の学習-

*青木 颯大¹ (1. 日本大学)

15:00 ~ 15:15

[C-09-04] データセット作成における自己教師あり学習の有効性の検討

*塩崎 雄晴¹、入江 寿弘¹、小林 伸彰¹、新宮 清志¹ (1. 日本大学)

15:15 ~ 15:30

ライドシミュレータ

Development of a Riding simulator

西川智紀¹⁾, 松谷拓実²⁾, 入江寿弘³⁾

Tomoki Nishikawa, Takumi Matsutani and Toshihiro Irie

1) 日本大学大学院 理工学研究科 大学院生(〒 274-8501 千葉県船橋市習志野台 7-24-1, E-mail: csto23027@g.nihon-u.ac.jp)

2) 日本大学大学院 理工学研究科 大学院生(〒 274-8501 千葉県船橋市習志野台 7-24-1, E-mail: csta23035@g.nihon-u.ac.jp)

3) 博士(工学)日本大学大学院 理工学研究科 教授(同上, irie.toshihiro@nihon-u.ac.jp)

This paper describes the development of a VR-based simulator, the "Ride Simulator". While research on VR technology has been rapidly advancing and VR simulators have been developed for various purposes, these devices are often limited in their applications and be expensive. To address these issues, we are aiming development of a versatile and cost-effective VR simulator.

Key Words : VR simulators

1. はじめに

近年の VR 技術の発展は目覚ましいものである。[1]しかし、VR 技術を使用した汎用的なシミュレータが開発されたという話は聞えなかった。用途が限られ、VR 技術のほかに外部からの刺激を与える VR シミュレータがほとんどである。そのため、カヌーなどの普段体験できない乗り物やスポーツを疑似的に再現することができる VR シミュレータ、「ライドシミュレータ」を開発する。次の図-1 は先行研究時のものである。[2]



図-1 VR カヌーシミュレータ (2019 年度)

2. 研究背景

スポーツの一種であるカヌーの乗り手の動きを体験することができる機械として、株式会社西川精機製作所にて開発された。カヌーに取り付けた 360 度カメラとスマ

ートフォンに搭載されている加速度センサを用いることで、角度を算出していた。しかし、先行研究の構造では本装置にかなりの負担があり、長時間稼働させ続けた結果モータドライバと制御基板が破損した。私たちはこれが特定の条件でリンクと重心との関係が崩れ一つのモータに過負荷がかかっていたのではないかと考えた。

3. 機構の決定

本研究でのロボットは人を乗せるための安全上負荷容量を高く設計する必要がある。そのためライドシミュレータの平行マニピュレータは、シリアルマニピュレータと比較して作業空間は小さいものの、高剛性、高精度、高速という特徴を有している。[3]以下の表-1 はそれをまとめたものである。

表-1 マニピュレータの特徴比較

	シリアルマニピュレータ	平行マニピュレータ
速度	小さい	大きい
剛性	小さい	大きい
精度	出しにくい	出しやすい
作業空間	大きい	小さい

表-2 平行マニピュレータの特徴比較

	旧型 (平行移動を制限)	新型 (角度変化を制限)
負荷容量	小さい	大きい
乗り手への影響	小さい	大きい

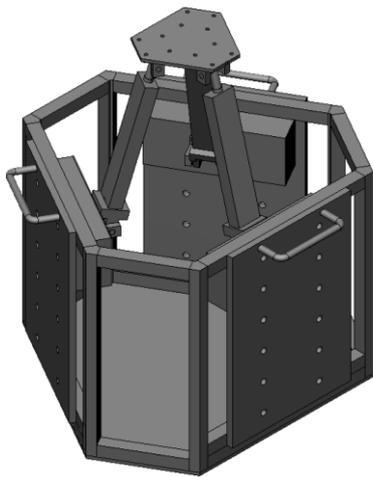


図-2 旧全体図

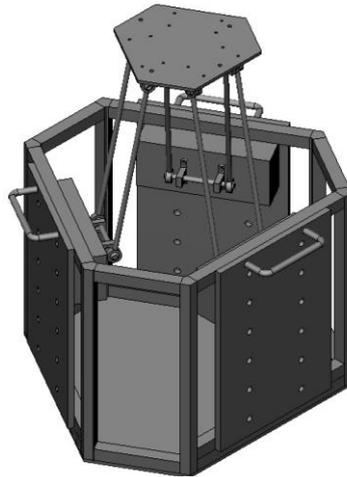


図-3 新全体図

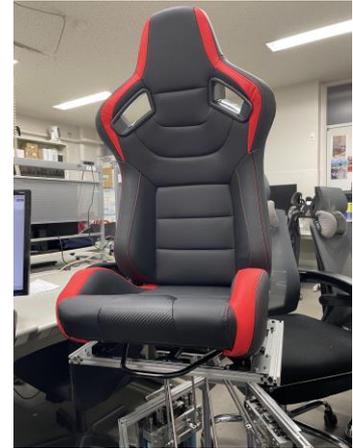


図-4 実物写真

本研究のマニピュレータはボールねじによる直動運動によって制御している。大きく違う点は天板とマニピュレータをつなぐリンク構造にある。図にそれぞれの形状について構造比較図を示す。(図-5) 先行研究にある旧型のパラレルマニピュレータ(図-2)について説明する。中心の移動を制限し角度だけを出力するものとなっており、大きくダイナミックな表現を可能にするが本装置が大きく傾きバランスを崩す恐れがある。また天板に設置するものによって重心自体も大きく変化するためモータへの過負荷が予想される。またその動きを考慮しても身長、座高、体重の違いによって乗り手の安全性が異なるため、シミュレータとして一つの問題になりうると考えた。新規に考案するパラレルマニピュレータ(図-3)は旧型と違い角度変化を制限しテーブルおよび乗り手を平行移動させるものとなっている。近いものとしては地震のようなものを再現することで先行研究とは違った感覚を再現することができる。実際には乗り手を水平に保ちつつ前に投げ出されるような感覚を再現できると考えている。身長差などによる乗り手の違いも大きく出ないと考えたため、これを採用することにした。以下の表-2 にて旧型と新型のパラレルマニピュレータについて特徴の違いをまとめる。

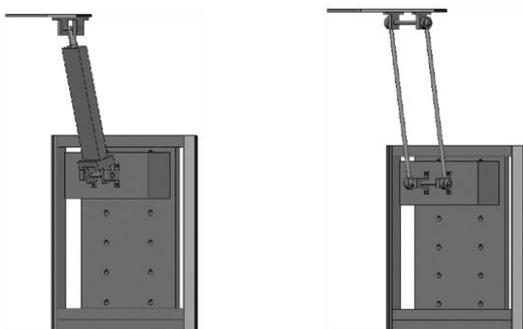


図-5 構造比較

4. XY平面における位置姿勢の近似推定

機械を制御するにあたり平行移動する都合上xy平面から近似させていきたい。(図-6, 図-7)

機械原点から目標値との偏角と距離、駆動限界半径Rから、システムの変数、比重角度 q と比重距離 d を設定する。

本装置の平面移動の限界半径を R 、本装置のテーブル上での中心とリンクをつなぐ距離を P とし目標値と最大値との割合を算出する。

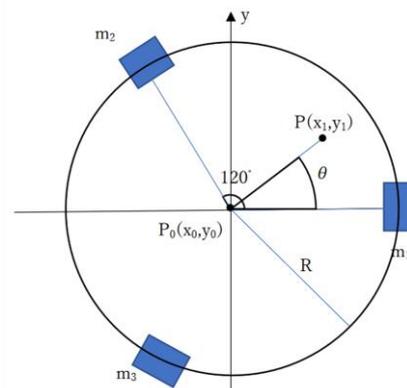


図-6 入力座標とマニピュレータの関係

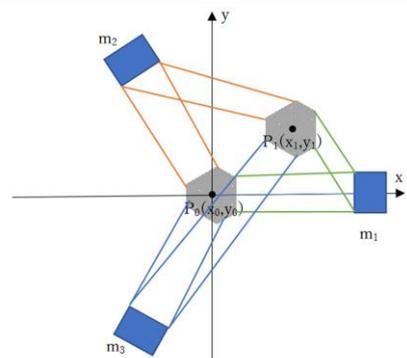


図-7 目標値へ到達後のテーブルとリンクの関係

$$q = \begin{cases} \frac{3}{2\pi} \left(\arctan \left(\frac{y}{x} \right) + \frac{\pi}{2} \right), & 120 \geq \theta > 90 \\ \frac{3}{2\pi} \arctan \frac{y}{x}, & 90 \geq \theta \geq 0 \end{cases} \quad (1)$$

$$d = \frac{\sqrt{x^2 + y^2}}{R} \quad (2)$$

さらに実際の直動システムの最大移動量を L_{max} とし $m_1 \sim m_3$ を計算する. まず, それぞれのリンクの必要な移動量を算出する.

$$m_1 = \frac{L_{max} d q}{2} \quad (3)$$

$$m_2 = \frac{L_{max} d (1 - q)}{2} \quad (4)$$

$$m_3 = \frac{\sqrt{3} (m_1 + m_2)}{2} \quad (5)$$

となる. z 方向の移動に関してはそれぞれのマニピュレータに一定の数値の増減で表現できるため, ここでは割愛する. 以上でパラレルマニピュレータの近似制御値を算出することが出来た. これらの式を用いて, m_1, m_2 区間内のそれぞれのボールねじの移動量と xy 平面の関係を示す等高線を用いてグラフにする. (図-8~10)

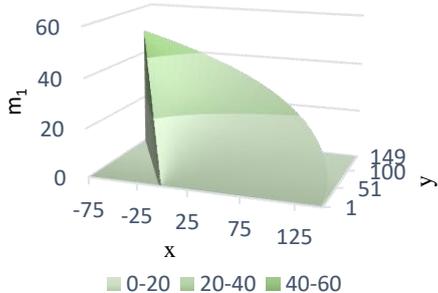


図-8 m_1 の xy 平面上における値

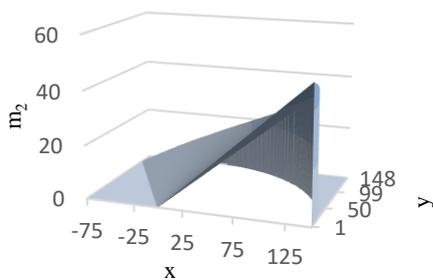


図-9 m_2 の xy 平面上における値

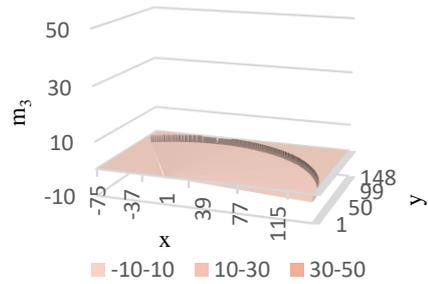


図-10 m_3 の xy 平面上における値

5. おわりに

先行研究にて課題と考えていた構造的な問題点を改善するにあたり, パラレルマニピュレータの運用方法を変更することで解決できるのではないかと考えた. しかし本開発を行うにあたり制御, 駆動方式の検討・安全性の向上について等, 課題の解決に向けてできることは多く残っているため, 次期につながる研究結果となった.

実際の負荷制御の実施はもちろんのこと, 実際にVRを使用した際の動作とその実感をレポートする必要がある. かねてよりパラレルマニピュレータにあった課題を明確化し複数のシミュレートに対応できるように今後は開発していきたい.

付録:

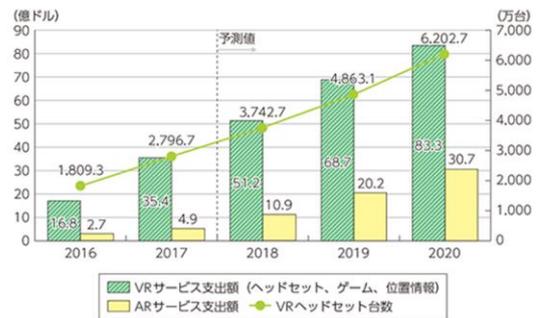


図-11 世界のAR/VR市場規模・VRヘッドセット出荷台数の推移

参考文献

[1] 総務省(2021年9月19日)「平成30年版情報通信白書」第1部特集人口減少時代のICTによる持続的成長
<https://www.soumu.go.jp/johotsusintokei/whitepaper/ja/h30/html/nd111350.html>

[2] しのぎ文化プラザ(2019年10月5日)
<https://www.shinozaki-bunkaplaza.com/staffblog/detail686>

[3] 楠田喜宏:「パラレルメカニズム実用化の展望」日本ロボット学会誌 Vol. 30 No. 2, pp.118~122, 2012
https://www.jstage.jst.go.jp/article/jrsj/30/2/30_2_118/_pdf

楕円型展開車輪を用いた階段昇降ロボットの試作 —実大機構モデルの作成と制御方法の検討—

Experimentally make the mobile robot that climb steps by using elliptic transformable wheels
— Examination of wheel construction and control method to increase the size —

加茂雄太¹⁾, 入江寿弘²⁾

Yuta Kamo and Toshihiro Irie

1) 日本大学大学院 理工学研究科 精密機械工学専攻 大学院生 (〒274-8501 千葉県船橋市習志野台 7-24-1)

2) 博士(工学) 日本大学 理工学部 精密機械工学部 教授 (同上, E-mail: irie.toshihiro@nihon-u.ac.jp)

Japan became a super-aging society, the number of wheelchair users will increase further than now. Although Japan have recently promoted barrier-free public buildings, there are a lot of bumps that prevent them from going their ways, so they often need to assistant in moving outdoors. As the way to solve such problems we study on the mobile robot that can climb steps by using transformable wheels, work on the development of the automatic control technique. In this paper, we experimentally made the mobile robot that have elliptic transformable wheels, consider control method to adjust the pitch of wheel automatically.

Key Words : Transformable Wheel, Wheelchair, 3D-Print, ToF, stereo vision

1. はじめに

我が国では超高齢社会を迎え、車椅子利用者は一層増えることが予想される。近年、公共建築等のバリアフリー化は進んでいるものの、車椅子の走行を妨げる段差は未だに数多く存在し、車椅子利用者は外出時に介助者の支援を必要とすることが多々ある。このような問題の解決策として、著者達は展開車輪を用いた階段昇降ロボットの実用化に向けて機体設計及び制御技術の開発を行っている。

今回、3Dプリンタを用いて階段昇降ロボットの試作機を製作し、階段昇降動作の自動化に向けて、車輪の制御モデルを考案した。

2. 方法

本研究で検討する階段昇降ロボットの移動方法と計測手法を以下に示す。

(1) 楕円型展開車輪

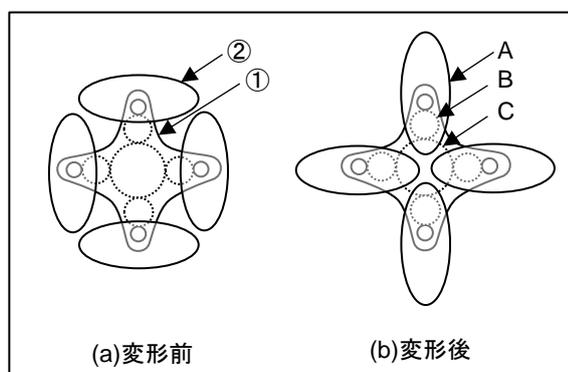


図-1 楕円型展開車輪の構造

展開車輪とは日本大学理工学部精密機械工学科の入江

寿弘教授により発明された車輪構造である。^[1] 楕円型展開車輪の構造を図-1に示す。展開車輪は大きく分けて展開駆動体①と転動爪体②により構成されており、回転軸から円周上に配置された各転動爪体はそれぞれ連携しながらの展開動作を可能としている。楕円型は中心の楕円駆動ギアAの動力をその周辺に配置されたアイドルギアBを介して、4つの楕円型転動爪体Cに伝えることで展開動作を行う機構になっており、これにより車輪は平地での走行において展開前の形状(図-1(a))を維持し、階段昇降時に展開動作によって展開後の形状(図2-(b))に変形することで転動爪体を段差に引っ掛けながら上昇・下降動作を実現することを想定している。

(2) TOF方式

距離計測の方法にTOF(Time of Flight)方式がある。TOF方式はセンサの送信機から送信した信号(レーザ光や超音波)が空气中を伝搬し、対象物の表面で反射して戻ってきた信号を受信器で受信する(図-2)。

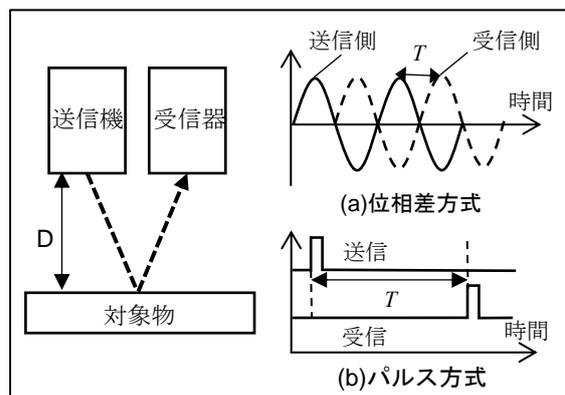


図-2 TOF方式

受信した信号の時間 T [s]から式(1)によって、距離 D [m]を計測する手法である。

$$D = \frac{1}{2} cT \quad (1)$$

ここで c は信号の速度[m/s]であり、レーザ光源の場合は光速、音波の場合は音速である。TOFの計測方法には位相差方式(図2-(a))とパルス方式(図2-(b))がある。位相差方式は信号の変化を用いて、送信側と受信側の位相差から時間 T を計測する。一方、パルス方式はパルス状の信号を送信機で発信してから、戻ってくるパルスを受信するまでの時間 T [s]を計測する。

(2) ステレオビジョン

2台のカメラを用いて異なる2視点から撮影した画像により3次元位置を求める手法をステレオビジョンという。ステレオビジョンでは、左カメラと右カメラの2台のカメラからなるステレオカメラを用いて、左右のカメラ取得した画像の見え方を利用して、画像のみから対象の位置を計測する。ここで内部のパラメータが等しい2台のカメラを用いて、カメラの光軸の方向、光学中心の高さを一致させて平行に設置したステレオカメラを平行ステレオカメラ(図-3)と呼ぶ。

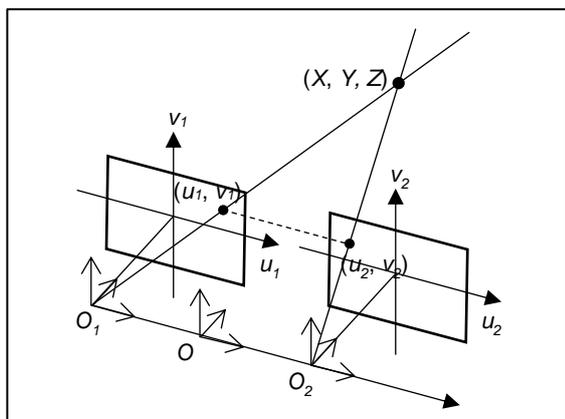


図-3 平行ステレオカメラ

カメラの焦点距離を f 、左カメラの光学中心 O_1 と右カメラの光学中心 O_2 の距離(基線長)を b とおくと、 O_1 と O_2 の中点を原点とした座標系での計測点の3次元位置 (X, Y, Z) は以下の式で求めることができる。

$$X = \frac{b(u_1 + u_2)}{2(u_1 - u_2)} \quad (2)$$

$$Y = \frac{bv_1}{(u_1 - u_2)} = \frac{bv_2}{(u_1 - u_2)} \quad (3)$$

$$Z = \frac{bf}{(u_1 - u_2)} \quad (4)$$

ただし、 (u_1, v_1) は左画像中での対象の位置、 (u_2, v_2) は右画像中での対象の位置である。

3. 結果

今回、実社会の階段を想定して楕円型展開車輪を試作し、階段寸法に対応した車輪の変形動作について制御モデルを考案した。

(1) 製作機体

試作した階段昇降ロボットに搭載した楕円型展開車輪の3Dモデル(図-4)と寸法(表-1)を以下に示す。

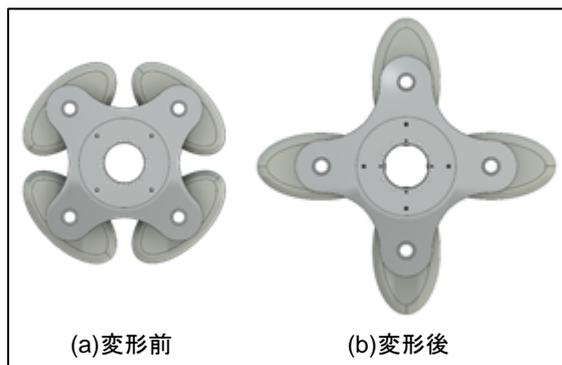


図-4 楕円型展開車輪の3Dモデル

表-1 楕円型展開車輪の寸法

r_1 :(a)車輪状態の半径	190[mm]
r_2 :(b)展開状態の半径	235[mm]

3Dモデルの作成はAutodesk社のFusion360を用いて行い、3Dプリンタにより展開車輪を製作した。楕円型展開車輪を搭載した階段昇降ロボットの試作機(図-5)と機体寸法(表-2)を以下に示す。



図-5 階段昇降ロボットの試作機

表-2 試作機の寸法

L :全長	1100[mm]
W :全幅	790[mm]
H :高さ	360[mm]

(2) 制御モデルの検討

今回、実社会に存在する未知の階段を想定して、深度

カメラを用いた階段寸法の計測により,展開車輪の展開量(楕円型転動爪体のピッチ)制御方法を検討した. ここで階段寸法と展開量の関係(図-6)について以下に示す.

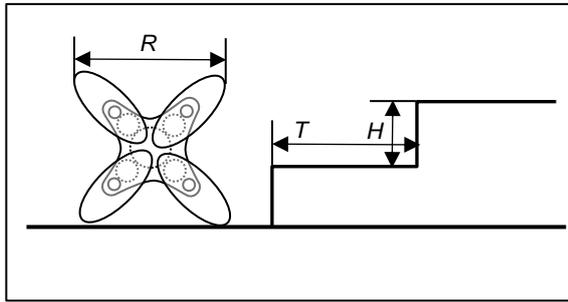


図-6 階段寸法と車輪の展開量^[2]

R は車輪の展開量, H は階段の蹴上寸法, T は階段の踏面寸法を表す. R は以下の式を充たす必要がある.

$$R > \sqrt{H^2 + T^2} \quad (6)$$

この R の自動制御を行うため検討した制御モデル(図-7)を以下に示す.

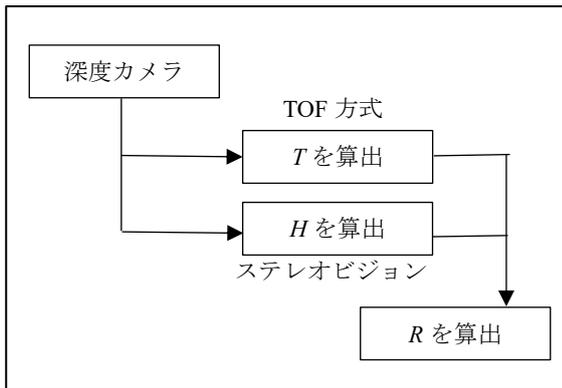


図-7 展開量の制御モデル

4. 結論

実在する階段に対応した大型の展開車輪の製作では,3DCADソフトにより設計データを作成し,3Dプリンタを用いて成形を行い,階段の動作実験可能とする車輪の製作を達成した.

階段昇降ロボットの試作において,製作手段として3Dプリント技術は有用であり,材料費が安価であることや複雑な形状を比較的容易に設計することができるため,今後の開発においても積極的に採用できる.

深度カメラを用いた展開量の自動制御方法について検討を行った. TOF方式により階段の蹴上寸法を既知としているが,計測精度について課題がある.

今後は機械学習を用いて画像計測の精度向上を検討し,展開量の自動制御を行う.

参考文献

- [1] 国立研究開発法人(2013年4月18日)「J-STORE(～展開車輪 入江 寿弘 特願 2011-192839 ～)」, <https://jstore.jst.go.jp/PDFView.html?type=nationalPatent&id=30648&property=openPdf>
- [2] 入江寿弘,新宮清志,“展開車輪を用いた階段昇降ロボットの誘導制御,”第33回ファジィシステムシンポジウム講演論文集, pp.293-296, 2017.

展開車輪を用いた階段昇降ロボット -深層強化学習による階段昇降の学習-

Stair Climbing Robot with Expandable Wheels
-Learning Stair Climbing Movements by Deep Reinforcement Learning-

青木 颯大¹⁾

Hayata Aoki

1) 日本大学大学院 理工学研究科 大学院生 (〒271-0077 千葉県松戸市根本351-1シティテラス松戸311, E-mail: csha19001@g.nihon-u.ac.jp)

Abstract : Expandable wheel is a wheel that can overcome steps and ascend and descend stairs equipped with a deployment mechanism. A two-wheeled robot equipped with a deployment wheel is operated on a simulator to simulate the physics of stair climbing and descending. The theme of this research is to determine the appropriate amount of deployment for stairs of arbitrary dimensions by using deep reinforcement learning in the process. The final goal is to implement the system on a wheelchair and operate it in places that are not barrier-free.

Key Words: Reinforcement Learning, Deep Reinforcement Learning, Expandable Wheel, Stair Climbing Robot

1. はじめに

現在の日本は少子高齢化が進行しており、バリアフリー化が求められている。しかし、スペース不足・拡張困難を始めとして様々な理由がバリアフリー化を妨げている。この際問題となるのが移動支援問題である。ほんの僅かな段差であっても要介護者にとっては大きな障害であり、特に車椅子使用者などが無理に乗り越えようとする重大な事故へつながる。

以上の問題を解決する手段の1つとして、著者が所属する研究室にて、段差乗り越えおよび階段昇降動作が可能である展開車輪が研究されている。展開車輪の研究は平成21年度から現在まで研究されているが、あらゆる寸法の階段に対応できているとは言えない。

従って本研究では、任意の寸法の階段に対して適切な展開量を決定できるシステムを完成させることにより、あらゆる寸法の階段に対応可能な展開車輪の開発を目的とした。その手段として本研究では、物理シミュレータを使用して階段昇降動作をシミュレーションし、その過程を強化学習、特に深層強化学習を適用する。本論文では、その途中過程を示す。

2. 展開車輪とは

展開車輪とは展開機構を搭載した車輪のことである。非展開時には車輪として機能することで、整地走破が可能とし、展開時には段差乗り越えおよび階段昇降動作が可能とする。平成21年度から現在までに楕円型、勾玉型の2種類が考案され、研究されてきた。本研究では楕円型を採用している。なお図1左が勾玉型、右が楕円型である。

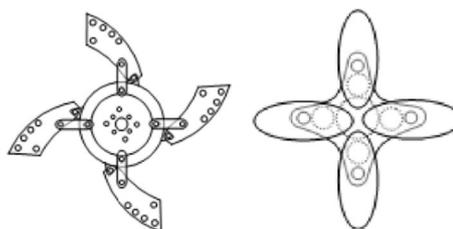


図1. 展開車輪概要図

3. 深層強化学習

本研究で用いる深層強化学習について示す。そのために、まずは強化学習についての概要を示す。

3.1. 強化学習とは

強化学習は機械学習の一種である。行動をする主体である「エージェント」と、そのエージェントが置かれた場所である「環境」と呼ばれるものがあり、このエージェントと環境が、相互にやり取りを行うことが強化学習の枠組みである。具体的なやり取りの流れは以下の通り。

- 1) エージェントが環境の「状態」を観測する
- 2) 環境から観測した状態と、エージェントが持つ「方策」に基づき「行動」を決定する
- 3) 選択した行動により状態が遷移するとともに、エージェントに報酬を返す
- 4) 1)~3)を繰り返す

強化学習は、上記のサイクルを繰り返していく中で、2)で示した方策をより良いものに更新し、最適方策を求めることが目的である。そして、方策を更新する指標として

3)で示した報酬を用いる。報酬の与え方は、良い行動や状態に対してプラスの報酬、悪い行動や状態に対してマイナスの報酬を与えることである。エージェントは環境の中でランダムに行動を行いながら、自らの方策を報酬により更新していくことで、最適方策を求める。しかし実際によく利用される強化学習アルゴリズムでは、最適方策を直接求めることはできず、方策を最適方策へと収束させることで行動の最適化を行う。

3.2. 深層強化学習とは

強化学習アルゴリズムはさまざまあるが、基本的な枠組みは3.1.で示した1)~4)の通りで共通である。しかし通常の強化学習アルゴリズムでは、ロボットの動作などの連続動作制御に向いていない。なぜなら、状態と行動の候補の数が爆発的に増加してしまうからである。この問題を解決する手段として、強化学習に深層化したニューラルネットワークを組み込んだ深層強化学習が挙げられ、本研究でも用いることにした。深層学習を組み込むことにより、莫大な数の状態と行動の候補を近似によりコンパクトに表現することができる。この近似により、現実に近いより複雑な問題への対応を可能とする。

4. 実装

4.1. 物理シミュレータ

展開車輪の適切な寸法も確定していない上に、実機での強化学習は時間がかかるため、実機を製作しての研究は効率が悪いと予想される。よって、物理エンジンを搭載したシミュレータを用いることで、その動作の過程を強化学習により最適化することにした。物理シミュレータを採用することにより、時間的、経済的コストを抑えながら研究を行うことができる。しかしデメリットもあり、強化学習が終了して実機への実装を行う際に、物理シミュレータと現実世界のギャップが考えられるため、物理シミュレータ上でのシミュレーションは、より現実に近い必要がある。

なお物理シミュレータとしては、PyBulletを採用した。PyBulletは当たり判定を厳密に行うことができる上、強化学習および深層強化学習のフレームワークの多くを適用できることから本研究で採用した。

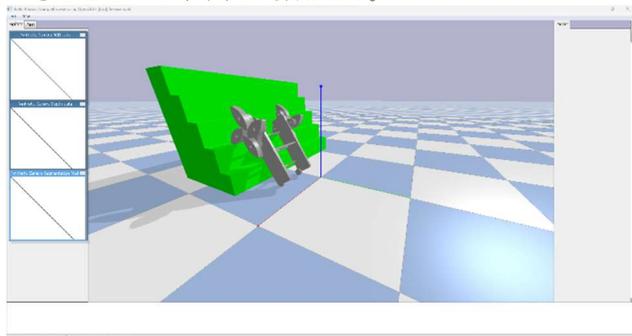


図2. PyBulletでの展開車輪のシミュレーション

4.2. 強化学習アルゴリズム

使用したい強化学習アルゴリズムは、DQN(Deep Q Learning)であるが、その前に Q 学習を行う。そのための定義を以下のように行った。

1) 状態(State)の定義

状態は、ロボットの位置・姿勢や階段の形状・位置などの情報を含むものである。これらの情報を観測することで、エージェントは自身の状態を把握することができる。PyBullet では、`getBasePositionAndOrientation()`関数を使用してロボットの位置・姿勢を取得できる。簡単のため、まず階段の寸法は固定する。ここでは、階段の寸法規格において最も勾配が厳しい、蹴上 230mm、踏面 150mm の寸法で固定する。また階段と機体の初期位置も固定し、階段の正面に機体が設置されるようにする。

2) 行動(Action)の定義

行動は、ロボットがとることができる操作の種類を示す。具体的には、各関節を動かす角度や速度などがある。本問題では、展開車輪の展開量決定動作と、機体を前に進める駆動動作の 2 つの動作に分けられる。展開量決定動作は、非展開時である 0° から 1 回転してもとの位置に戻るまでの 180° の範囲で、 1° 刻みで角度を決定し、その角度で固定する。機体の初期値を階段の正面に設置されていることにより、駆動動作では単純に前進するのみとする。

3) 報酬(Reward)の定義

報酬は、エージェントの行動の良し悪しを評価する指標である。本問題ではまず、単純に機体の位置が高くなったらプラスの報酬、高いところから低くなったらマイナスの報酬を与えることにする。今後は、安定した姿勢を維持しながらの階段昇降を可能とするため、階段の傾きに対して機体重心の軌跡が平行となっているときにプラスの報酬を与えることも考えている。

4) Q 値(Q-Value)の定義

Q 値とは行動価値関数のことである。強化学習の基礎であるマルコフ決定過程(Markov Decision Process)に基づいて定義され、ベルマン方程式により求められる価値関数である。強化学習における目的関数である。実際に Q 学習で使用する Q 値の式は、ベルマン最適方程式から求められる式で、TD 法(Temporal Difference)の 1 種である。式は以下の通り。

$$Q'(S_t, A_t) = Q(S_t, A_t) + \alpha \{R_t + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)\} \quad (1)$$

式(1)に基づき Q 関数を繰り返し更新することで、最適方策における Q 関数へ収束していく。

5) エージェント(Agent)の学習

エージェントは、Q 値を更新することで学習するが、学習中には「探索」と「活用」のバランスが重要である。探索では、ランダムな行動を選択することで新しい情報を収集する。活用では、現在の最適な行動を選択することで、学習済みの知識を活用する。これらのバランスは、 $\epsilon(0\sim1)$ の確率で探索を行い、 $1-\epsilon$ の確率で活用を行う ϵ -greedy 法により実装する。

以上を元に、強化学習アルゴリズムを PyBullet 上のロボット動作に適用する。

5. 実験

実際に、展開車輪の階段昇降動作を行い、以下の実験を行った。

5.1. 実験1：シミュレータと現実のギャップ実験

強化学習を行う前に、シミュレータと現実のギャップを測るための実験を行う。具体的には去年度の卒業研究にて著者らが製作した楕円型展開車輪で実際に行った実験を PyBullet 上で再現をし、比較を行うことである。その実験とは、乗り越え可能な最大段差測定実験である。なお2で示した楕円型には、図3のような従来の楕円型に加え、図4のような、著者が令和4年度卒業研究にて製作した楕円型の2種類がある。後者は、従来の楕円型の展開するための回転軸をずらし、非展開時から展開時における車輪径の変化量増加に成功した。

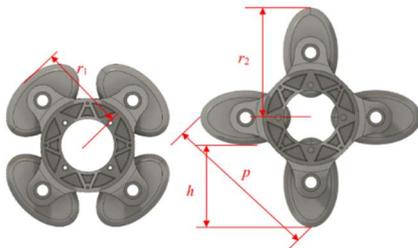


図3. 従来の楕円型概要

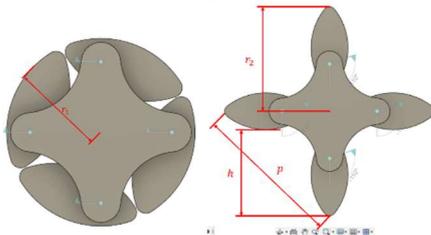


図4. 去年度の楕円型概要

実際に去年度の研究で行った実験は、スチレンボードで作成した段差の乗り越えである。スチレンボードは10mm厚であったため、10mm刻みで測定を行った。なお、卒業研究で製作した展開車輪の寸法は以下の通りであり、実寸大の1/2スケールで製作した。

表1. 卒業研究で製作した楕円型の寸法

非展開時の半径 r_1	96[mm]
展開時の半径 r_2	160[mm]
最大乗り越え段差 h	132[mm]
展開車輪ピッチ p	226[mm]

この展開車輪は、最大で150mmの段差乗り越えに成功した。シミュレーションでも同様の実験を行ったが、150mmを乗り越え、160mmを乗り越えられなかった。これは実機での実験結果と一致し、シミュレーションの精度はある程度担保されていることが証明できた。今後は、本実験以外で様々な検証を行い、より現実に近いシミュレーションを行えるようにする。

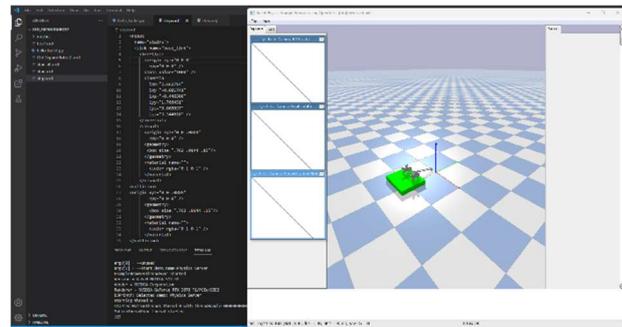


図5. 段差150mm乗り越え成功

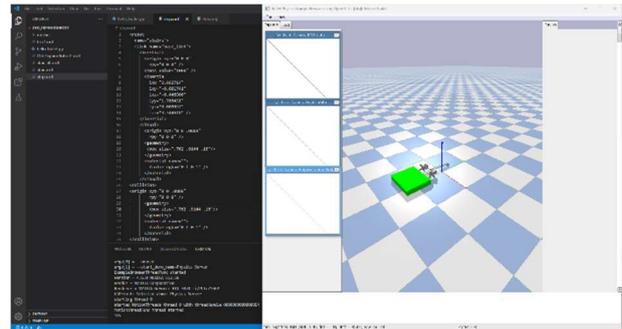


図5. 段差160mm乗り越え失敗

段差乗り越えの挙動が、現実に近いことが分かったため、今度は150mmから160mmの間を、1mm刻みで調べた。結果は、157mmまで乗り越え可能であり、158mmで乗り越え失敗であった。

5.2. 実験2：Q学習による階段昇降動作の最適化

4.2で示した内容を、Pythonコードに書き起こした。しかし、うまく学習ができなかった。階段昇降のシミュレーション自体はできているが、それを強化学習の枠組みに当てはめることができなかった。

6. 展望

今後の展望としては、まず今回うまく実装できなかった強化学習アルゴリズムを実装することである。原因としては、PyBulletへの理解が足りなかったことである。PyBulletでの動作は完了し、強化学習のアルゴリズムの理解もできていたが、PyBulletで状態をうまく取得できなかったことが原因であった。

これを解決し、Q学習を実装できた後は、DQNを実装する。その後はDQNが適しているかどうかを、他のアルゴリズムとの結果と比較をすることで検証していく。

また、実機とシミュレーションのギャップを埋めるためにも、もう一度実機で同様の実験を行い、まず157mmを乗り越えることができるのか、そして158mmで乗り越え不可能となるのかを実証する。

参考文献

- [1] 斎藤康毅:ゼロから作るDeepLearning4 強化学習編
- [2] PyBullet @MISC{coumans2020, author = {Erwin Coumans and Yunfei Bai}, howpublished = {¥url{http://pybullet.org}}, year = {2016--2020} }
- [3] 牧野浩二/西崎博光:TensorFlowによる深層強化学習入門-OpenAI Gym+PyBulletによるシミュレーション

データセット作成における自己教師あり学習の有効性の検討

Investigating the effectiveness of self-supervised learning in dataset creation

塩崎雄晴¹⁾ 入江寿弘²⁾ 小林伸彰³⁾ 新宮清志⁴⁾
Yusei Shiozaki, Toshihiro Irie, Nobuaki Kobayashi, and Kiyoshi Shingu

¹⁾ 日本大学大学院 理工学研究科 精密機械工学専攻 大学院生

(〒 274-8501 千葉県船橋市習志野台 7-24-1, csys21001@g.nihon-u.ac.jp)

²⁾ 博士 (工学) 日本大学 理工学部 精密機械工学科 教授 (同上, tirie@eme.cst.nihon-u.ac.jp)

³⁾ 博士 (工学) 日本大学 理工学部 精密機械工学科 准教授 (同上, kobayashi@eme.cst.nihon-u.ac.jp)

⁴⁾ 工学博士 日本大学 名誉教授 (〒 102-0072 東京都千代田飯田橋 3-10-1-1901, kshingu@ocean.cst.nihon-u.ac.jp)

Self-supervised learning (SSL) is a hopeful way to obtain features of target data with minimal human supervision in the future. There are well-known extensive datasets like COCO or ImageNet in the world that we can use to train the models. However specific manual annotation tasks have been required to make unique datasets. Therefore, we developed an application to make a dataset and present the efficiency of an SSL-based annotation that can learn features without ground truth in advance.

Key Words : ML/AI, CV, SSL, App, Annotation, Dataset

1. はじめに

データセット作成作業は膨大な単純作業と時間の消費を伴うため、教師あり学習における最も避けたい作業の一つである。このために以下にして効率化するか、一人当たりの作業量を減らすか等、近年に至るため様々な手法が提案されてきた。半教師あり学習や自己教師あり学習もその一つであり、教師データ作成作業を伴うことなく、入力データの特徴量を効率的に取得できる。現在では、COCO [1] や ImageNet [2] 等の大規模データセットは既に存在しており、これらを用いた学習済みモデルを使用することで下流タスクではより高いスコアを得ることができる。しかし、食品の傷みなどを判別するような大規模データセットにない特定の条件ではいまだに自身で教師データを追加してデータセット作成する必要がある。そこで、我々は自己教師あり学習を応用して事前に任意の入力データを分類することで人間のタスクを最小限に抑える手法を提案する (図-1)

2. 背景

教師あり学習は学習手法の一つであり、多くは手作業で作成されたデータセットを参照して、正例を正しく認識できるモデルを作るために行われる。そこで、我々はこの参照データを作成することをアノテーションと呼び、モデルに識別して欲しい情報をデータに含めることを指している。特に画像領域ではアノテーションには画像に含まれる対象の大きさ、位置の指定が必要とされ、これらを一枚一枚手作業で行うことは膨大な時間と体力を要するとされている。人海戦術的に解決することも可能だが、人的資源や予算に限りがある場合は現実的でなく、この点で研究開発に遅れがでる可能性があり早急な改善が求められている。我々が提案する

アプリケーション及び手法では事前にクラスターとして分類された画像群を人間が一括で名付けを行う。これによってアノテーションに要する作業量は入力データ量が多ければより効果を示すことが期待できる。

3. 関連研究

(1) 作業工程の改善

クラウドソーシングを利用し複数人でデータセット作成することで一人当たりの負荷を軽減する手法が提案されているが、作業人間のバウンディングボックスのバラツキなどが問題とされている。この問題に対処するためにアノテーション作業を三つの工程に分割することでアノテーションの精度を改善している [3]。

(2) ツールの改善

事前学習済みモデルを使用し、事前にバウンディングボックスを作成することでアノテーション作業を効率化する研究は数多くあり、この事前に作成したバウンディングボックスの正負をユーザーが指定することでさらなる作業の効率化に成功している [4]。加えて、オブジェクトの中心点をクリックすることで物体の位置形状を推定しバウンディングボックスを生成する手法も提案されている [?]

LOST [5] はアノテーション作業負荷を軽減するために開発されたツールであり、Multi Image Annotation (MIA) に対応している。MIA とは、事前に画像群をまとめて分類しておくタスクのことであり、これによる作業者の作業時間を図-2に示す。しかし、このツールでは前もって各画像にアノテーションを行うことが必要であり、この作業は前述の通り最も削減した工程である。この工程のことを Single Image Annotation (SIA) と呼ぶ。

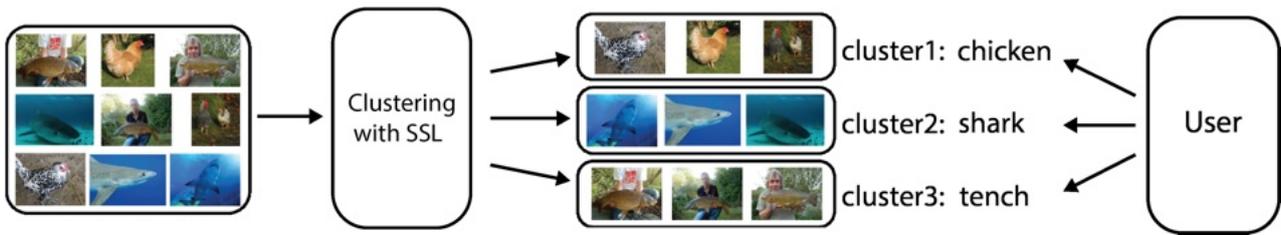


図-1 概要

(3) 手法やモデルの改善

自己教師あり学習ではアノテーションをすることなく良い特徴量の学習ができる。これを事前学習とすることで様々な下流タスクでより良い結果を期待できる。この特徴を利用することで我々がアプリケーションで達成したい機能を得られると期待できる。そこで、我々はこれら自己教師あり学習モデルである SwAV [6] と iBOT [7] について、比較検討を行った。

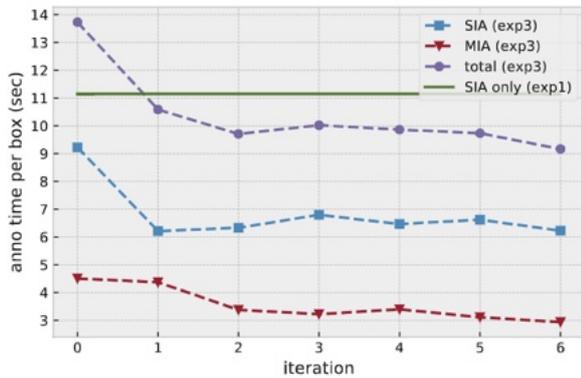


図-2 Time per box

4. 実験

SwAV と iBOT を使用した際のスコアを比較する。スコアには教師なしでのクラスタリングでは NMI, ARI, FMI を使用し、教師ありでの分類では学習させたデータの割合毎に精度を求め、比較を行った。その実験結果を表-1 から表-6 に示す。各クラスターに割り当てられたデータラベルを図-3, 図-4 に示す。

(1) クラスタリング

クラスタリング評価の結果を表-1 に示す。SwAV, iBOT はそれぞれ NMI において 57.9%, 71.0% を示した。これらのモデルには訓練済みのパラメータを使用した。加えて、ImageNet でのクラスタリング結果を図-3, 4 に示す。これらは各クラスターに含まれる種類が少ない方が良い結果である。これらの図を見ると、NMI の高い iBOT の方がより良くクラスタリングができていることが視覚的にも理解できる。

(2) 画像認識 (MNIST)

次に、MNIST を使用してデータ情報が少ない場合の SwAV と iBOT の半教師あり学習の精度を比較する。そ

表-1 クラスタリングスコア

Method	Arch	NMI [%]	ARI [%]	FMI [%]
SwAV	RN50	57.88	52.45	42.16
iBOT	ViT16/S	71.00	70.80	57.74

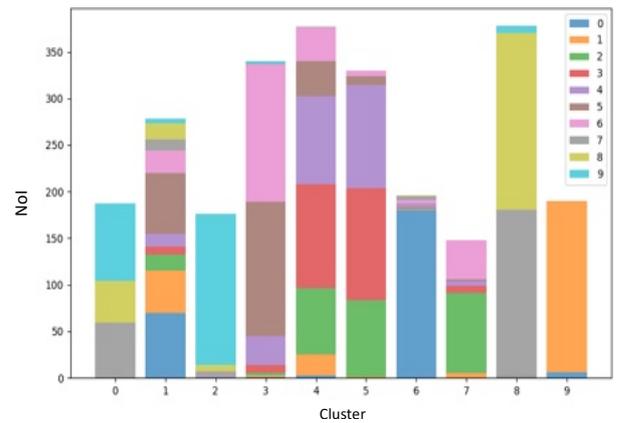


図-3 各クラスターに含まれるラベル (SwAV)

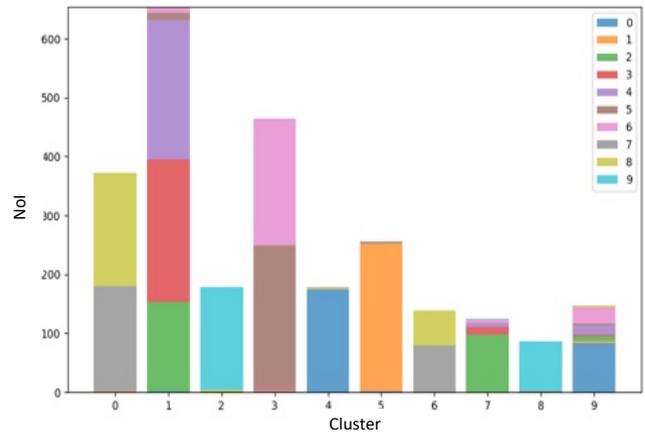


図-4 各クラスターに含まれるラベル (iBOT)

の実験結果を表-3 と表-4 に示す。実際の環境では事前に入力データのカテゴリ分布を揃えて入力することを想定していない。その状況を再現するためにランダムに取得したデータを入力として使用した場合とそうでない場合について実験を行った。本実験で使用したシード値は 251 とした。この場合、不均衡データの学習について考慮しなければならない点に注意する。



図-5 使用したラベルと画像

表-2 使用したラベル ID の対応表

Label	Category
n01440764	tench, Tinca tinca
n01443537	goldfish, Carassius auratus
n01484850	great white shark, white shark, man-eater, man-eating shark, carcharodon carcharias
n01491361	tiger shark, Gakeocerdo cuvieri
n01494475	hammerhead, hammerhead shark
n01496331	electric ray, crampfish, numbfish, torpedo
n01498041	stingray
n01514668	cock
n01514859	hen
n01518878	ostrich, Struthio camelus

表-3 MNIST での SwAV の各抽出割合における精度

Extraction	Epoch	1% Label	5% Label	10% Label	50% Label
Not Random	20	96.67	99.79	99.51	99.95
Not Random	200	100	100	100	100
Random Seed 251	20	57.81	93.72	97.12	99.10
Random Seed 251	200	78.93	94.42	97.61	99.31

表-5 ImageNet での SwAV の各抽出割合における精度

Extraction	Epoch	1% Label	5% Label	10% Label	50% Label
Not Random	20	52.58	64.00	77.77	92.31
Not Random	200	51.27	68.73	79.19	92.50
Random Seed 251	20	44.46	61.69	75.62	92.39
Random Seed 251	200	44.46	65.16	78.42	93.12

表-4 MNIST での iBOT の各抽出割合における精度

Extraction	Epoch	1% Label	5% Label	10% Label	50% Label
Not Random	20	10.87	92.69	44.71	91.86
Not Random	200	75.77	99.24	97.71	99.27
Random Seed 251	20	9.79	20.41	92.90	97.50
Random Seed 251	200	41.42	96.76	98.71	99.54

表-6 ImageNet での iBOT の各抽出割合における精度

Extraction	Epoch	1% Label	5% Label	10% Label	50% Label
Not Random	20	38.96	65.42	79.65	87.35
Not Random	200	58.16	72.23	81.69	93.58
Random Seed 251	20	21.08	70.73	77.35	85.08
Random Seed 251	200	34.69	74.85	79.15	92.96

(3) 画像認識 (ImageNet)

続いて、ImageNetに含まれる10クラスを抽出して半教師あり学習の評価を行った。その実験結果を表-5, 6に示す。

(4) アノテーション作業時間

ImageNetから取り出した10クラス2600枚のデータセットを下に、一枚あたりのアノテーション作業時間を計測した。この時のデータセットに使用したラベル群と画像例をそれぞれ表-2, 図-5に示す。しかし、取り出したラベルにおいて、識別が困難なものがあったため、n01484850とn01491361を”shark”, n01496331とn01498041を”ray”, n01514668とn01514859を”chicken”として扱う。よってラベルとしては全7種として実験を行った。この実験結果を図-6に示す。この実験ではiBOTのアーキテクチャをベースに最終層にk-meansを使用したものを使用した。実験では画像260枚の作業時間は129.9秒であり、一枚あたりの作業時間は0.5秒となる。また、画像1300枚での作業時間は294.7秒、一枚あたりでは0.23秒、画像2600枚では一枚あたり0.15秒という結果となった。

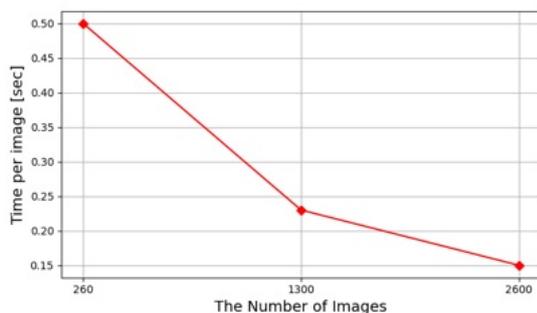


図-6 Time per image

5. 終わりに

本研究では、SwAVとiBOTを使用してデータセット作成における自己教師あり学習の有効性を検討した。iBOTでのクラスタリングではNMIが71%と高い値を示し、SwAVとiBOTは半教師あり学習において両モデルともMNIST, ImageNetで90%を超える高い精度を示した。さらに、SwAVは特にMNISTでは20エポック時点ではiBOTよりも高い精度を示している。このことから、SwAVは情報の少ない画像の学習において、半教師あり学習で使用する場合には一般に精度が高いことで知られているViTモデルであるiBOTよりも早く良い結果を得られると言える。また、ImageNetにおいても50%ラベルにおいては20エポック時点ではSwAVがiBOTよりも高い精度を示した。

また、ImageNetでは90%を超えるような高い精度を得るには50%以上のデータが必要であることが、5, 6よりわかる。しかし、iBOTの学習にはより多くの時間が必要になる。そのため、クラスタリングにおいてはiBOTのようなViTモデルを使用し、作成したデータを用いた半教師あり学習ではSwAVのようなCNNモデルを

使用することとした。実際にクラスタリングを行っ多者に対し作業者がラベル付けを行った場合にかかった時間は画像2600枚に対して一枚あたり0.15秒という結果となった(図-6)。

本研究の目標はアノテーションにおけるSIAプロセスを撤廃することであるため、実際には物体検出のために領域候補の予測が必要になる。特に教師なしでの領域候補の予測については2022年に発表があった[8]。また、生成モデルの学習において、物体の位置空間を含んだ情報を学習することで物体の位置予測が可能になることが示唆されているため、これらの手法について検討し研究を進めていく[9]。特に、大規模モデルの使用については限られた資源における実際の運用について詳しく検証を重ねる必要がある。

参考文献

- [1] COCO - Common Objects in Context.
- [2] ImageNet.
- [3] Hao Su, Jia Deng, and Li Fei-Fei. Crowdsourcing Annotations for Visual Object Detection. Technical report.
- [4] Dim P. Papadopoulos, Jasper R. R. Uijlings, Frank Keller, and Vittorio Ferrari. We don't need no bounding-boxes: Training object class detectors using only human verification. feb 2016.
- [5] Jonas Jäger, Gereon Reus, Joachim Denzler, Viviane Wolff, and Klaus Fricke-Neuderth. LOST: A flexible framework for semi-automatic image annotation. oct 2019.
- [6] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised Learning of Visual Features by Contrasting Cluster Assignments. jun 2020.
- [7] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. iBOT: Image BERT Pre-Training with Online Tokenizer. nov 2021.
- [8] Amir Bar, Xin Wang, Vadim Kantorov, Colorado J Reed, Roi Herzog, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. DETReg: Unsupervised Pretraining with Region Priors for Object Detection. Technical report.
- [9] Zhixuan Lin, Yi-Fu Wu, Skand Vishwanath Peri, Weihao Sun, Gautam Singh, Fei Deng, Jindong Jiang, and Sungjin Ahn. SPACE: Unsupervised Object-Oriented Scene Representation via Spatial Attention and Decomposition. jan 2020.