深層学習を用いたロボットアームのピックアップ作業に関する研究

坂田　修一[*1]　、清水　智[*1]、福田　聖斗[*1]、辛　徳[*2]

Study on Pick-up Works using Deep Learning

Shuichi Sakata[*1], Satoru Shimizu[*1], Masato Fukuda[*1], Duk Shin[*2]

Abstract　In recent years, industrial robots has been improved in automatic works by using deep learning. For example, a system has been developed in order to pick up parts piled in a bulk. This system recognizes objects and generates gripping motions and accurately pick up parts using various gripping methods. The purpose of this research is to create a system that performs general object recognition on an object projected by the camera using deep learning. The proposed system could measure the distance to the object with the depth of camera and perform pick-up action.

1 序論

1.1 研究背景

工場や倉庫では生産性、効率性向上のために多くの作業を機械によって自動化する取り組みがされている[1]。しかし、ピックアップ作業では様々なモノを扱うため、人の手で作業を行うことが多く、単純な作業ではあるものの重労働であるため、時間がかかっていた。近年、深層学習（Deep Learning）を活用することでロボットによるピックアップ作業の自動化が進んできており作業の効率化が行われている。例えば、バラ積みにされている部品をピックアップするために、深層学習を用いて対象物の認識、把持動作の生成を行い、多様なつかみ方で部品を正確にピックアップするシステムが開発されている[2-4]。

本研究では、深層学習を用いてカメラで映し出された物体が何であるか一般物体認識を行い、物体との距離を深度カメラで測定し、ピックアップを行うシステムを開発することを目的とする。

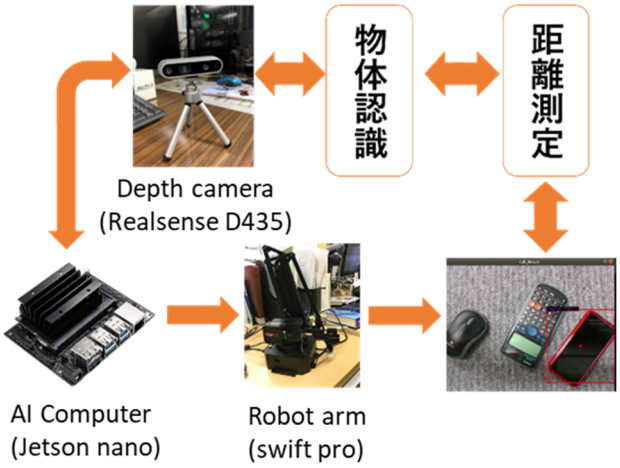

図 1 提案システムの構成図

2 深層学習を用いたピックアップシステム

2.1 提案システムの概要

図 1 のように提案するピックアップシステムを示す。深層学習を行うため提案システムは AI コンピュータ、深度カメラ、ピックアップアームの三つの要素で構成される。

2.1.1 Jetson nano

動作環境は AI コンピュータである Nvidia 社の Jetson nano を使用する。Jetson nano は、小型でありながら演算処理能力が高く、AI アルゴリズムを高速で実行することが可能であるため、物体認識などのディープラーニングを行うのに適したデバイスであり、エッジデバイスへの AI 導入も可能である。

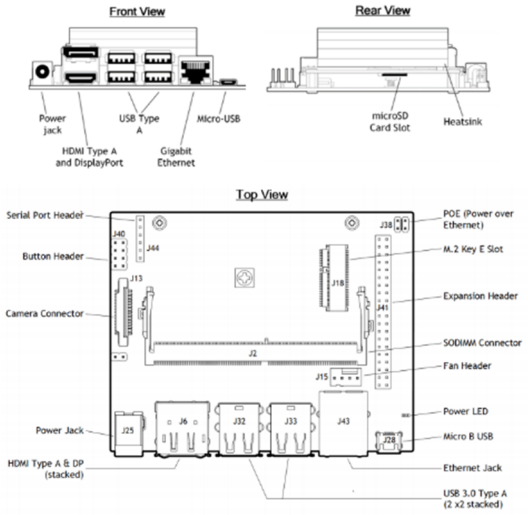

図 2 Jetson nano 開発者キット

[*1] 東京工芸大学工学部電子機械学科学部生　[*2] 東京工芸大学工学部電子機械学科准教授

2020 年 3 月 25 日 受理

エッジデバイスとは、機械などに搭載されているセンサなどから収集したデータをクラウドなどのデータセンターへ送信し、処理をさせる従来のクラウドコンピューティングでは大容量の通信回線が必要になってしまい、通信に遅延が発生してしまう恐れがあるので、デバイス側に処理機能を持たせたデバイスのことを言う。

Jetson nano は、SD カードに OS をダウンロードして使用する。OS は ubuntu18.04LTS で、公式サイトに Jetson nano で使用するための OS があるため、そちらを使用した Jetson nano を図 2 に示す。

2.1.2 深度カメラ

本研究では、図 3 に示したように物体との距離を測定するために、深度カメラ（RealSense D435、Intel）を使用した。オープンソースの intel RealSense SDK2.0 は、Windows や Mac、LINUX などでも使用可能で、Python、ROS、Unity、OpenCV、LabVIEW のプラットフォームや言語もサポートされており、開発研究のしやすいデバイスである。

本来であれば、一般物体認識を行うためのカメラと深度を測定するためのカメラが二台必要になるが、D435 は RGB センサと深度センサが搭載されており同時に使用することが可能である。図 3 の左側は RGB カメラの写真であり、右側は深度写真である。色により距離（青いほど近い、赤のほど遠い）が表れていることが分かる。

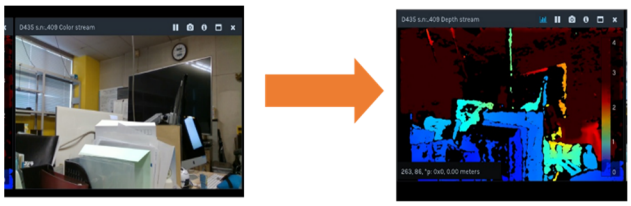

図 3 深度カメラでの距離測定

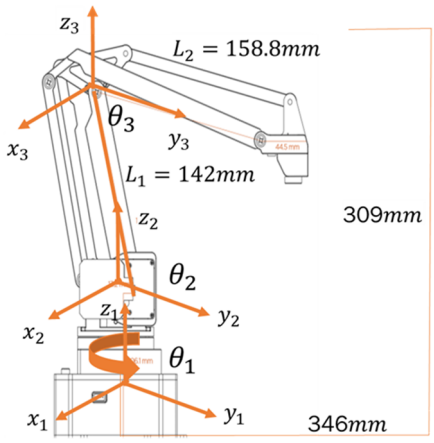

図 4 ピックアップアームのリンクの座標系

2.1.3 ピックアップアーム

本研究で使用するピックアップアームは、UFACTORY 社の uArm Swift Pro である。これは、3 自由度のオープンソース卓上ロボットアームで、 プログラム制御することができる。 また、ソフトウェアである uArmStudio を使えば、パソコンでの操作やプログラムによる制御が可能である。先端はオプションパーツを用いることで３D プリントや、レーザー刻印が可能である。今回は先端パーツを吸盤で吸引してピックアップするパーツでピックアップするシステムを Python 言語でプログラミングを行う。
図 4 にピックアップアームのリンクの座標系を示す。

2.2 物体認識プログラム

2.2.1 深層学習の概要

深層学習とは、人や動物などの脳の神経回路をモデルにしたニューロンを多層構造にした図 5 のようなニューラルネットワークを構成した学習アルゴリズムのことであり、最新の人工知能技術の一つである[5]。

ニューラルネットワークは、大きく分けて 3 つの層に分かれている（図６）。1 層目は入力層、データが一番初めに入っていく層になる。2 層目は隠れ層、この層が多段階になっているニューラルネットワークのことを深層学習と呼び、層が多ければより複雑で正確な判断ができる。3 層目は出力層、2 層目で判断した結果を出力する層である。特に、隠れ層の中間層が多いことが特徴である。深層学習はアルゴリズムを用いて学習データに含まれる特徴を抽出し、指示をしなくても自動で学習を行う手法である。また、機械自らが実行するため、精度を高めるには大量の学習データが必要になり、学習データによって学習の方向性も変わるので膨大な量のデータを慎重に選ぶ必要がある。

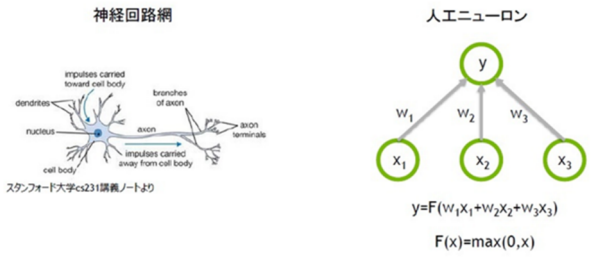

図 5 ニュロンモデル

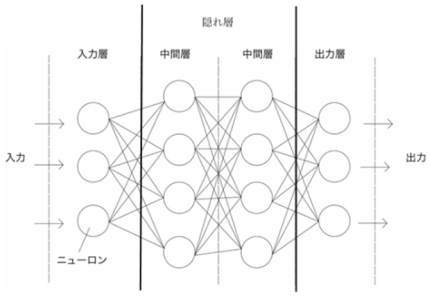

図 6 ニューラルネットワーク

2.2.2 YOLO の概要

物体認識を行うため使用した YOLO (You Only Look Once) はリアルタイムオブジェクト検出システムである[6]。YOLO は end-to-end と呼ばれる深層学習手法の一つで、画像をニューラルネットワークに入れるだけで簡単に検出することが可能である。end-to-end 手法を使った検出システムはほかにもあるが、YOLO は検出と識別を一つのニューラルネットワークを使い、同時に行うことによって処理速度を上げている。

アルゴリズムは入力する画像を 7×7 のグリッドで区切り、各グリッドに対して B 個のバウンディングボックス(bounding box)と信頼度スコア(confidence score)を推測する。信頼スコアは、そのバウンディングボックスの領域が背景であるのか物体であるのかの確率で

$$P_r(Object) \times IOU_{pred}^{truth} \quad (1)$$

と定義する。数値は物体である場合 1 になり、背景であるなら 0 になる。それと並行して、各グリッドが各クラスに該当している確率を

$$P_r(Class_i|Object) \quad (2)$$

で表す。各グリッドにおいては一つのクラスだけしか推測はしない。上記の二つを掛け合わせることで、バウンディングボックスの領域がどのクラスであるかを推論する。

$$P_r(Class_i|Object) \times P_r(Object) \times IOU_{pred}^{truth} = P_r(Class_i) \times IOU_{pred}^{truth} \quad (3)$$

2.3 距離測定プログラム

深度カメラで映し出された物体が何であるかを、検出システム YOLO で認識を行い、物体との距離を深度カメラで測定するために、以下のように Python でプログラミングを行った。制作したプログラムは、バウンディングボックスの領域が cell phone(スマートフォン)だった場合、そのバウンディングボックスの中心位置の距離を測定し、cell phone 以外の物体の距離は測定しないようなプログラムになっている。閾値として、認識率が 30%を越えなければ測定を行わない。また、cell phone が複数カメラに映った場合、最も認識率が高いものだけを測定する。
プログラムの実行結果を図 7 に示す。

2.4 ピックアッププログラム

YOLO によって認識した物体の場所と、距離測定で測定した数値によってアームを制御し、ピックアップを行うプログラムを Python でプログラミングを行った。アームは 2.3 のプログラムで距離を測定する位置である、バウンディングボックスの中心位置にアームを動かすようにした。また、アームの先端部品である吸盤にはスイッチが付いており、今回は安全装置としてスイッチが押されたらアームの下降を中断し、待機状態に戻るようにプログラミングを行った。

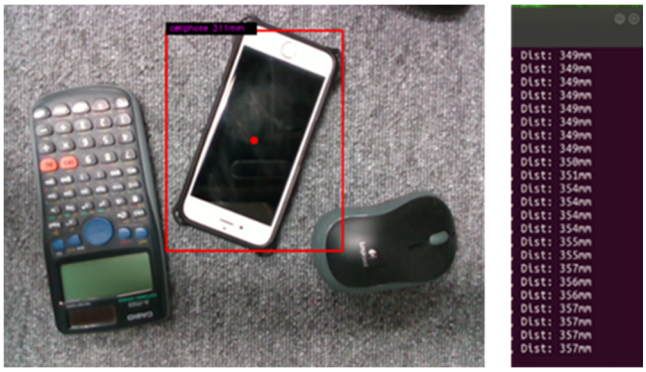

図 7 プログラム実行結果

3 システムの性能評価実験

3.1 YOLO の選定実験

3.1.1 実験方法

物体認識を行うリアルタイム検出システムである YOLO は、多くの種類があり種類によってニューラルネットワークのモデルが異なり、処理速度や認識精度に違いがある。そこで、Jetson nano で動作させるのに、適したモデルを選定するための選定実験を行った。今回は、YOLOv3、YOLOv2、YOLOv2-Tiny、YOLOv3-Tiny の 4 種類のモデルの中から選定を行う。実験方法は、実際に Jetson nano で各モデルを動作させることで選定を行った。今回選定する際に認識した物体は、2.3 で制作したプログラムに合わせて cell phone を認識させた(図 7 参照)

3.1.2 実験結果

結果は表 1 に示す。YOLOv2[7]や YOLOv3[8]は、認識率が高くはあるのだが処理が遅く、カメラ画像に遅延が生じており、これではリアルタイムオブジェクト検出とは言い難かったため、Jetson nano で YOLOv2、YOLOv3 を用い、物体認識を行うには厳しいという結果となった。YOLOv2-Tiny では、処理が早くカメラ画像に遅延は生じていなかったが、測定物である cell phone を信号機と誤認してしまい、認識精度に問題があったため、YOLOv2-Tiny は本研究での物体認識には適していないとした。YOLOv3-Tiny は、YOLOv3、YOLOv2 と比較すると認識率は低かったが、しっかりと cell phone と認識できており、2.3 で制作したプログラムの閾値である認識率 30%を満たし、かつ処理も早くカメラ画像に遅延が生じていなかった。このことから、Jetson nano で YOLO を用いる際に最

も適したモデルは、YOLOv3-Tiny であるという結果になった。この結果から。本研究では YOLOv3-Tiny を用いて物体の認識を行った。

表 1 YOLO のベンチマーク

種類	FPS	認識率（スコア）[%]
YOLOv2-Tiny	8.5～9	0 (信号機と誤認80%)
YOLOv3-Tiny	7.0～8.0	30
YOLOv2	2.8	90
YOLOv3	1.2～1.3	70～80

3.2 深度カメラでの距離の測定実験

3.2.1 実験方法

3.1 で選定した YOLOv3-Tiny を用いて、認識した物体との距離を測定する。測定は、2.3 でのプログラムで行う。実験方法は、3.1 と基本的には同じだが、測定物は深度カメラから見て中心に置くこととする。これは実際の距離を測定する際に読み間違いを少なくするためである。測定は高さを 35.9cm、25.5cm、21.6cm の三段階に分け、それぞれの高さで 3 回ずつ図 8 のように測定を行う。

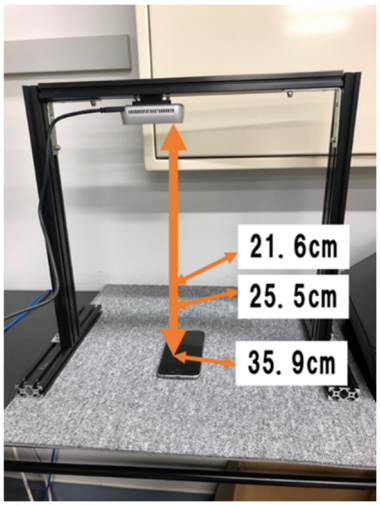

図 8 距離測定実験

3.2.2 結果と考察

実験結果を表 2 に示す。深度カメラで測定した数値は、距離を変えたり測定物を動かしたりしない限り、何度繰り返しても一定の数値を測定することができた。実距離との誤差は、約 0.2cm 生じた。この誤差は、 実距離を測定する際に誤差が生じてしまった可能性もあるが、深度カメラのサンプルプログラムにおいても、0.2cm 程度の誤差が生じていることから、実距離の測定誤差や自作したプログラムが原因で誤差が生じているわけではなく、深度カメラの性能によって生じたと考えられる。今回の誤差が 0.2cm と小さかったため、このプログラムを使いピックアップの実験を行った。

表 2 深度カメラによる距離測定結果

実距離	一回目	二回目	三回目	平均	誤差
35.9cm	35.6cm	35.6cm	35.6cm	35.6cm	**-0.3cm**
25.5cm	25.6cm	25.6cm	25.6cm	25.6cm	**+0.1cm**
21.6cm	21.7cm	21.7cm	21.7cm	21.7cm	**+0.1cm**

3.3 ピックアップ実験

3.3.1 実験方法

3.1 の選定実験で選定した YOLOv3-Tiny を用いた 2.3 のプログラムと、2.4 のプログラムを組み合わせ、深度カメラによって取得したデータからアームを制御し、実際にピックアップできるか実験を行う。実験は図 9 のような環境で行い、測定物をカメラの中心に置き、高さを 3.2 の実験と同じ、35.9cm、25.5cm、21.6cm と三段階に分けて、それぞれの高さで 3 回ずつピックアップを行う。

図 9 ピックアップ実験の様子

3.3.2 結果と考察

実験結果は表 3 に示す。測定物との距離を 35.9cm、25.5cm、21.6cm とし、どの高さにおいても三回と実験を繰り返し、ピックアップをすることにすべて成功した。しかし、測定物の中心にアームを下降するようにプログラムを制作したが、誤差が生じてしまった。また測定物がカメラの中心から離れれば離れるほど座標に誤差が生じ、下降す

る距離にも誤差が生じた。原因は 2.4 で制作したプログラムで、カメラ画像の座標とアームの座標に誤差があるためだと考えられ、どの位置においてもピックアップを可能にするには、座標と下降する距離のプログラムを修正する必要があると考えられる。

表 3 ピックアップ結果

測定物との距離	一回目	二回目	三回目
35.3cm	○	○	○
29.3cm	○	○	○
30.5cm	○	○	○

4 結論

4.1 まとめ

本研究では深層学習を用いたピックアップ作業に関する研究を行った。深層学習を用いた一般物体認識システムである、YOLO の性能評価実験では、認識率や処理の性能評価を行った結果、Jetson nano では YOLOv3-Tiny が最適であることした。深度カメラでの距離測定の性能評価実験では、誤差が平均 0.2cm と小さくピックアップ作業に支障はないことを確認した。実際に制作したプログラムでピックアップ作業を行った結果、測定物（スマートフォン）との距離によってアームを下降させるピックアップに成功した。

4.2 今後の課題

課題として、測定物がカメラの中心ではなく、どの位置においてもピックアップすることが可能なシステムにするには、カメラの座標とピッキングアームの座標のプログラムを修正する必要がある。

参考文献

1. L. Pinto and A. Gupta, "Supersizing self-supervision: Learning to grasp from 50K tries and 700 robot hours," 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, pp. 3406-3413, 2016.
2. E. Matsumoto, et. al.: End-to-end learning of object grasp poses in the Amazon Robotics Challenge, 2017 IEEE International Conference on Robotics and Automation (ICRA), 2017.
3. E. Matsumoto: 深層学習でバラ積みロボットの 0 から学習, Preferred Networks, https://tech.preferred.jp/ja/blog/robot_binpick_deep_learning/
4. 三次元物体認識技術を応用したバラ積みピッキングシステムの開発、林　俊寛　他、https://www.ihi.co.jp/var/ezwebin_site/storage/original/application/4a2a6b334c4ee44309400fcde6dae7c3.pdf
5. “ニューラルネットワークの基礎解説：仕組みや機械学習・ディープラーニングとの関係は”. ビジネス IT https://www.sbbit.jp/article/cont1/33345#&gid=null&pid=1
6. J. Redmon et. al.: You Only Look Once: Unified, Real-Time Object Detection, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016.
7. J. Redmon, & A. Farhadi: YOLO9000: Better, Faster, Stronger, IEEE conference on computer vision and pattern recognition (CVPR), 2017.
8. J. Redmon, & A. Farhadi: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.