

딥러닝 기반의 로봇팔 시스템 연구

신준호*, 심규석**
*한국산업기술대학교 기계공학과
**충북대학교 기계공학부
jhshin9345@gmail.com, rbtjrl23@gmail.com

A Study on Deep Learning Based RobotArm System

Jun-Ho Shin*, Gyu-Seok Shim**
*Dept. of Mechanical Engineering, Korea Polytechnic University
**Dept. of Mechanical Engineering, ChungBuk National University

요 약

본 시스템은 세 단계의 모델을 복합적으로 구성하여 이루어진다. 첫 단계로 사람의 음성언어를 텍스트로 전환한 후 사용자의 발화 의도를 분류해내는 BoW방식을 이용해 인간의 명령을 이해할 수 있는 자연어 처리 알고리즘을 구성한다. 이후 YOLOv3-tiny를 이용한 실시간 영상처리모델과 OctoMapping모델을 활용하여 주변환경에 대한 3차원 지도생성 후 지도데이터를 기반으로하여 동작하는 기구제어 알고리즘 등을 ROS actionlib을 이용한 관리자시스템을 구성하여 ROS와 딥러닝을 활용한 편리한 인간-로봇 상호작용 시스템을 제안한다.

1. 서론

본 논문에서는 Google Cloud의 Speech API를 이용한 안드로이드 어플리케이션에서 사람의 음성을 텍스트로 변환한 이후 단계부터 시작하여 로봇이 문자를 송신받아 사람의 의도를 파악하는 과정을 자연어 처리 과정을 통해 순서를 고려하지 않는 빈도수 기반의 단어 표현 방식인 Bag of Words(BoW)를 응용한 BoWClassifier 방식을 이용해 사용자 발화의 도인 Intent를 구별하는 IntentClassifier와 로봇이 가능한 기능을 연결 시킬 수 있도록 하였다.

자연어처리 사전작업으로 형태소 분석기를 통해 미리 분류해두었던 명사부분을 인식할 객체로 고려하여 이미지 처리부로 전달하고, 이미지 처리부에서는 실시간 객체인식이 가능하면서 경량화된 YOLO-tiny 모델을 이용하여 객체를 잡아내고 로봇암이 객체까지의 거리를 판단하여 명령 작업을 이행할 수 있도록 객체가 존재하는 공간에 대하여 지도화 작업인 3D Mapping을 수행한다.

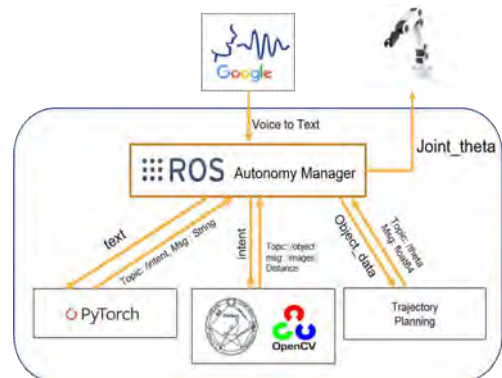
지도화 작업 이후에는 로봇암 기구제어가 시작되고, 동시에 작업영역을 Hough tranform 방식을 이용하여 검출해낸다. 전체적인 프레임워크 구축을 위해 자연어 처리 모델과 실시간 이미지처리 모델, Hough Transform 알고리즘을 ROS기반으로 설계

하여 Gazebo 시뮬레이터에서 자체 제작한 로봇모델을 사용하여 진행하였다.

2. 본론

2.1. 시스템 구조

Actionlib을 이용하여 시스템 구조 관리자인 Autonomy Manager를 만들었고, 각각의 모델들은 node가 되어 topic을 주고 받는 publisher와 subscriber의 역할을 동시에 수행하게 된다. topic publishing과 subscribe 과정을 수행할 때 마다 Autonomy Manager를 거치게 된다.[1]



[figure 1] System Architecture overview

2.2. 자연어처리 모델

자연어처리 모델은 Pytorch를 기반으로 설계되었으며 빈도 수 기반의 BoW방식과 데이터 전처리단계와 Tokenize, 어휘구축 텐서화, 모델링, 훈련, 평가 단계로 이루어진다.

2.2.1. 전처리와 Tokenize, 어휘구축 과정

자연어 처리를 위해서는 우선적으로 형태소 분석을 시행하여야 한다. 한국어 형태소 분석은 KoNLPy라는 형태소 분석기의 Okt 분석엔진을 이용했고, 분석된 문장을 데이터로 수행하고 명사로 분류된 단어들을 이미지 처리단으로 전송하여 주변 객체를 찾도록 한다. 전처리 단계와 토큰화 된 데이터를 대상으로 어휘구축단계를 실행하는데, 사전학습시에 클래스를 나누어 GRAB(잡기), STIR(휘젓기)등의 기능을 index_class를 통해 사전에 인덱싱작업을 마치고 실시간으로 입력되는 데이터들은 index_word를 통해 어휘별로 ID 인덱싱 과정을 거친다.

2.2.2 IntentClassifier

IntentClassifier는 Bag of Words(Bow)방식으로 사전에 인덱싱된 Intent 데이터를 이용하여 사용자 발화의도를 분류하는 과정이다.

index_word와 그 시퀀싱 데이터를 통해 BoW를 정의 하게되는데 index_word의 length데이터를 이용하여 텐서화를 거친다. 사전에 인덱싱된 데이터와 일치하지 않거나 빈공간을 고려하여 BoW를 정의하고, tensor를 return하여 데이터를 재 학습하게되면 학습 데이터의 size를 구해낼 수 있다.

도출된 train_data의 size와 구축된 어휘의 size를 이용하여 선형변환 과정을 거쳐 IntentClassifier를 모델링한다.

IntentClassifier 모델과 CrossEntropyLoss로 정의된 손실함수와 IntentClassifier 모델의 parameter를 호출한 SGD를 통해 optimizer를 정의하여 구성되는 최종적인 자연어처리모델을 이용하여 사용자 명령이 input될때 마다 실시간으로 분석을 통해 intent를 파악 할 수 있다.

2.2.3. 자연어 모델 평가

정의된 모델을 통해 다수의 Test data를 대상으로 예측하고 평가해본 결과는 [figure 2]와 같다.

```

Input : 이거 잡아
Prediction : GRAB

target : GRAB

Input : 이거 좀 저어줘
Prediction : STIR

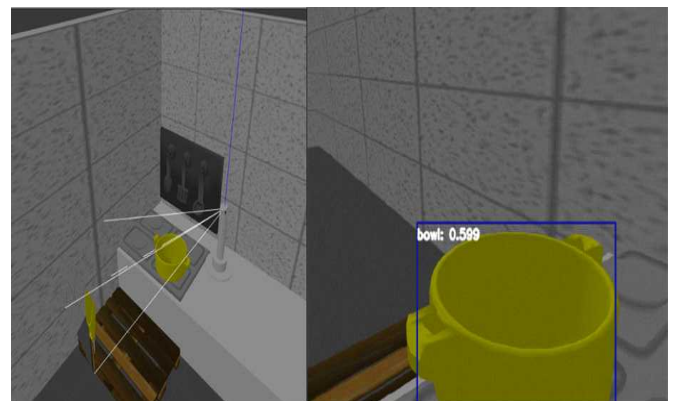
target : STIR
    
```

[figure 2] 모델 예측 및 평가 결과

2.3. 영상처리 모델(Real-Time Object Detect & Mapping)

2.3.1 객체검출

로봇암이 실시간으로 물체를 인식하기 위해 YOLO(You only look once)모델을 사용하였다. image 데이터를 이용하여 하나의 CNN으로 객체를 검출하고 부류를 결정한다.[2] 로봇암 주변의 있는 물체를 파악하여 image상의 Local Poision(x,y)을 파악한다.



[figure 3] YOLO-tiny 테스트 화면

[figure 3]는 YOLO-tiny모델[2]을 테스트하기 위해 알고리즘을 검증한 결과이다. 물체위치를 파악한 후 로봇암이 요리를 돕기 위해 Hough Circle 알고리즘[3]을 이용하여 냄비, 접시, 컵 등의 원 중심을 검출한다. 원의 중심점과 원의 외곽선 검출을 통해 볶기, 젓기 등의 동작이 수행되는 범위를 bowl의 내부로 제한하는 역할을 할수 있다.

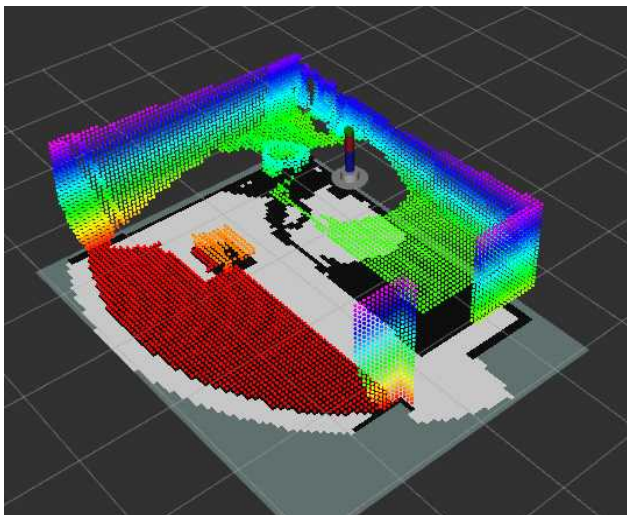


[figure 4] Hough Circle 시뮬레이션 검증

[figure 4]와 같이 물체를 인식하고 물체의 원을 파악하는 알고리즘을 시뮬레이션을 통해 검증하였다.

2.3.2 Octomap을 이용한 3D Mapping

로봇암의 위치와 주변의 지도정보를 위해 SLAM 알고리즘인 Octomapping을 사용하였다.[5] RealSense의 Depth image를 이용하여 로봇암이 구동시 로봇암 주변의 지도정보를 저장한다.

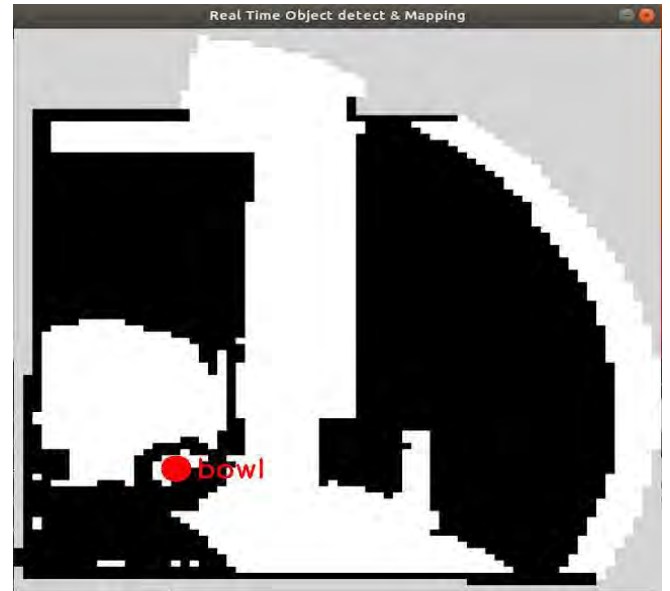


[figure 5] Octomapping 시뮬레이션 검증

[figure 5]와 같이 Depth image의 Point Cloud를 활용하여 로봇암의 주변 지도 정보를 얻어 장애물 및 주변 상황을 실시간으로 파악할 수 있다. 또한 물체와의 거리값을 파악하여 로봇암이 움직일 수 있는 거리를 파악하였다.

2.3.3 알고리즘 병합

병합과정은 2.3.1 - 2.3.2의 과정을 통해 수신받은 데이터를 기반으로 물체인식과 Mapping을 통해 지도 정보위에 물체의 위치를 추가하여 지도를 생성한다. 이 지도는 로봇암이 물체가 어느 위치에 있는지를 파악한다.



[figure 6] 물체위치를 포함한 지도정보

[figure 6]는 시뮬레이션의 주방지도와 냄비의 위치정보를 로봇암이 구동시 물체인식과 Mapping을 하여 저장한 지도정보이다. RealSense-D435i를 이용하여 주변의 지도정보와 물체위치를 갖는 정보 서비스 알고리즘을 검증하였다.

2.4 기구제어(4DOF 로봇암 위치 제어)

Local Position을 tf 변환을 통해 Global Position으로 변환한후[6], 로봇암의 위치를 제어하기 위해 IK(Inverse Kinematics)를 사용하였다. IK는 목적지를 이용하여 각 링크의 각도를 계산하는 알고리즘이다. Z축을 포함한 3DOF에서는 목적지만 있으면 각 링크의 각도를 파악할 수있지만, Z축을 포함한 4DOF에서는 목적지 뿐만 아니라 각 링크의 전체 각도의 합을 필요로한다. 그러므로 IK 알고리즘을 응용하여 최종 목적지로 부터 End Effect Link의 길이를 반지름으로 하는 원을 생성하여 End Effect Link의 각도를 지정하는 알고리즘을 제작하였다.

- $L1 = 25, L2 = 20, L3 = 5$
- *Desire position* : $dpos_x, dpos_y, dpos_z$
- *Current position* : $cpos_x, cpos_y, cpos_z$
- *Circle position* : Cx, Cz

$$d = \sqrt{dpos_x^2 + dpos_z^2} = \sqrt{L1^2 + L2^2 - 2 * L1 * L2 * \cos\theta}$$

→ θ_1, θ_2 도출

$$Px, Pz = \min(\arg(\sqrt{(cpos_x - Cx)^2 + (cpos_z - Cz)^2}))$$

$$\theta_\alpha = \text{atan}\left(\frac{Pz - L1 * \sin(\theta_1)}{Px - L1 * \cos(\theta_1)}\right)$$

$$\theta_\beta = \text{atan}\left(\frac{dpos_z - Pz}{dpos_x - Px}\right)$$

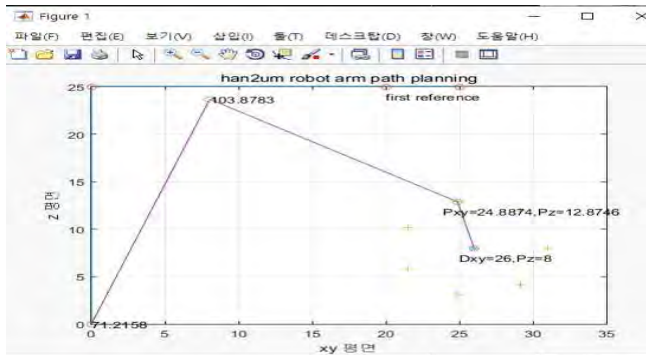
$$\therefore \theta_0 = \text{atan}\left(\frac{y}{x}\right)$$

$$\theta_1 = \text{acos}\left(\frac{d^2 + L1^2 - L2^2}{2 * L1 * L2}\right)$$

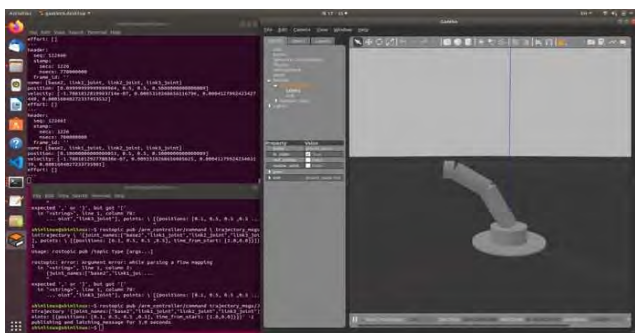
$$\theta_2 = -\text{acos}\left(\frac{d^2 - L1^2 - L2^2}{2 * L1 * L2}\right)$$

$$\theta_3 = (\theta_\beta - \theta_\alpha)$$

위 4DOF 기구학식의 $\theta_0 - \theta_3$ 까지의 각도를 구하여 로봇암 위치를 제어한다. 이 방식은 움직인 현재 위치로 부터 가장 가까운 경로를 설정하여 작동한다.



[figure 7] 로봇암 제어 알고리즘 검증



[figure 8] 로봇암제어 시뮬레이션 검증

[figure 7] 를 통해 위치 값(x,y,z)을 주었을때 4DOF 로봇암의 위치를 제어할 수 있음을 검증하였다.

3. 결론

본 논문은 사용자의 음성명령을 텍스트화 한 데이터를 시작으로 딥러닝을 기반으로 하여 자연어처리와 실시간 이미지 처리과정을 거쳐 로봇암의 제어과정을 ROS를 이용해 종합적으로 관리할 수 있는 시스템에 대해 제안한다. 딥러닝 과정을 거친 결과물들은 높은 수준의 인식률을 보여주고 있으나, 시야각 등의 문제로 카메라 센서를 통해 인지 불가능한 환경영역에서의 변수에 대한 제어방식에 대한 후속 연구가 필요하다. 또한 본 논문에서 제시하는 딥러닝 모델들은 한정적인 분류의 데이터를 대상으로 학습을 진행하였으므로 다변화된 분류의 데이터의 학습을 진행하여 다양한 음성명령인식에도 대비할 수 있는 추가적인 학습모델 준비가 필요 할 것이다.

[본 논문은 과학기술정보통신부 정보통신창의인재양성사업의 지원을 통해 수행한 ICT멘토링 프로젝트 결과물입니다]

참고문헌

- [1] Morgan Quigley 외 2명, Programming Robots with ROS, JPub, 2019년 2쇄
- [2] Naman Mittal 외 3명, “Object Detection and Classification Using Yolo”, International Journal of Scientific Research & Engineering Trends, Volume 5, Issue2, 2019, 4
- [3] 이건명, ”인공지능: 튜링 테스트에서 딥러닝까지”, 경기도 파주시, 생능출판, 2019년.
- [4] 천인국, ”디지털 영상처리”, 경기도 고양시, 인티니티북스, 2019년
- [5] Armin Hornung 외 4명, “OctoMap: an efficient probabilistic 3D mapping framework based on octrees”, Springer Science+Business Media New York, 34, 189-206, 2013.
- [6] 정슬, “로봇 공학5판”, 경기도 파주시, 청문각, 2019년.