

딥 러닝을 기반으로 한 자율주행 기술

송종원, 강태준, 김태홍, 홍유리, 김기천*

건국대학교 컴퓨터공학과

{송종원 foemzos, 강태준 bulletdog, 김태홍 rlaxoghd94, 홍유리 yuri0127, kckim*}@konkuk.ac.kr

Deep-Learning Based Automatic Drive Technology

Jongwon Song, Taejun Kang, Taehong Kim, Yuri Hong, Keecheon Kim*
Department of Computer Science and Engineering, Konkuk University

요 약

본 논문에서는 Raspberry Pi 와 Arduino 로 제작된 모형자동차를 가지고 Tensorflow 기반으로 학습한 데이터를 외부 서버에서 연산하여 자율적으로 운행하는 방법을 보인다.

1. 서 론

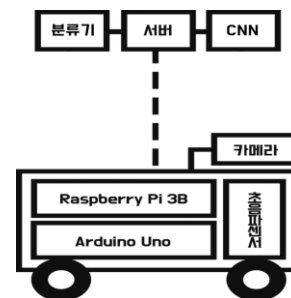
현재 국내외로 자율주행 차량은 많은 기업들이 관심을 가지고 투자하고 연구하는 분야이다. 자율주행 자동차는 총 6 단계로 나누어 지는데, 현재 상용화된 자율주행 자동차는 1~2 단계 정도이고, 여러 기업들이 3~4 단계의 자율주행 자동차를 연구하고있다. 완전한 자율주행을 하기 위해선 자동차 스스로 생각하고 판단을 내려야 하는데 그러기 위해서는 인공지능이 필수적이다.

본 논문에서는 카메라를 설치한 모형자동차를 사용하여, 사진을 수집하고, 수집한 사진들을 여러가지 딥 러닝(신경망)을 사용해서 학습을 시킨 후에, 학습한 내용대로 모형자동차가 제작한 트랙에서 별다른 조작없이 정해진 트랙을 주행하는지 확인한다.

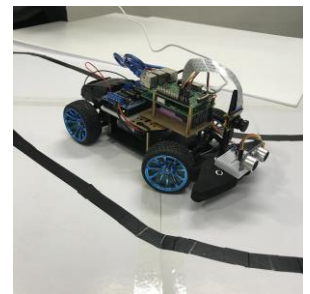
2. 시스템 구성

Raspberry Pi 와 Arduino 를 사용해서 모형자동차를 만들고, 모형자동차는 자신의 카메라로 촬영 중인 정보를 스트리밍 처리 후 Server 로 전송한다. Server 는 수신한 데이터를 바탕으로 Deep-Learning 을 통해서 학습한 데이터를 기반으로 모형자동차의 주행에 필요한

데이터를 생성하고 모형자동차에 전송한다. 모형자동차는 수신된 주행 데이터를 바탕으로 주행을 수행한다.



[그림 1. 시스템 구조]



[그림 2. 실제 모형자동차]

3. 딥 러닝

3.1. 신경망 구축

NVIDIA 의 ‘End-to- End Learning for Self-Driving Cars’ 논문을 바탕으로 Convolutional Neural Network 을 모델링 하였다. 기본 모델을 생성한 후 여러 시행 착오를 거쳐 아래와 같은 신경망 구조를 생성하였다. [1]

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 12, 12, 24)	624
p_re_lu_1 (PReLU)	(None, 12, 12, 24)	3456
batch_normalization_1 (Batch Normalization)	(None, 12, 12, 24)	96
conv2d_2 (Conv2D)	(None, 4, 4, 36)	21636
p_re_lu_2 (PReLU)	(None, 4, 4, 36)	576
batch_normalization_2 (Batch Normalization)	(None, 4, 4, 36)	144
conv2d_3 (Conv2D)	(None, 4, 4, 48)	43248
p_re_lu_3 (PReLU)	(None, 4, 4, 48)	768
batch_normalization_3 (Batch Normalization)	(None, 4, 4, 48)	192
conv2d_4 (Conv2D)	(None, 4, 4, 64)	27712
p_re_lu_4 (PReLU)	(None, 4, 4, 64)	1024
batch_normalization_4 (Batch Normalization)	(None, 4, 4, 64)	256
max_pooling2d_1 (MaxPooling2D)	(None, 3, 3, 64)	0
conv2d_5 (Conv2D)	(None, 3, 3, 64)	36928
p_re_lu_5 (PReLU)	(None, 3, 3, 64)	576
batch_normalization_5 (Batch Normalization)	(None, 3, 3, 64)	256
max_pooling2d_2 (MaxPooling2D)	(None, 2, 2, 64)	0
flatten_1 (Flatten)	(None, 256)	0

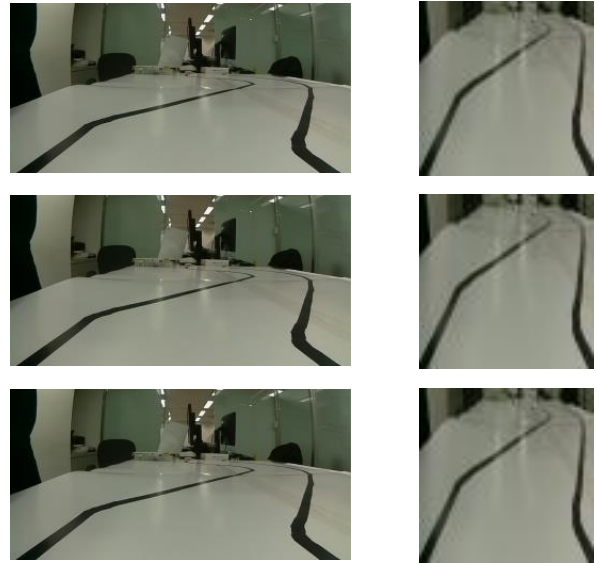
[그림 3. 신경망 구조]

tensorflow 기반 python 라이브러리 ‘keras’를 사용하여, 구축하였다. CNN의 Input data로는 RC Camera로부터 수신된 Image, Output data로는 RC가 가져야 할 조향각을 갖는다. Output data의 범위를 제한하기 위해, -1~1 사이의 숫자 값으로 조향각을 표시하며, output의 범위가 -1~1 까지이기에 마지막 layer의 활성화 함수를 tanh 함수로 사용하였다. NVIDIA에서 제안한 CNN의 구조에 성능향상을 위하여, 기존의 ReLU 활성화 함수를 PReLU 함수로 대체하고, 비선형성을 고려하여, ‘he_normal’ 함수를 초기화 함수로 사용한다. 또한, Inner Covariate Shift 현상을 해결하기 위해, Batch Normalization을 사용하여 Layer의 입력을 정규화 하고, Lambda Layer를 추가하여, CNN의 Input 데이터를 정규화 한다.

3.2 학습 데이터 생성

학습데이터를 생성하기 위해, PC에서 방향키를 눌러서 RC를 조작한다. 이 과정에서 PC에서 누른 방향키를 Raspberry Pi로 송신하게 되고, 이 값을 통해서 Raspberry Pi는 RC Car의 좌, 우 Motor 값을 조정한다. RC Car가 동작하는 와중에 10Fps로 RC Camera에서 수집된 Image와 RC Car의 좌, 우 Motor 전력의 차이를 쌍으로 PC로 전송한다. 수신된 Image를 배열로 변환한

뒤 이 이미지 배열과 쌍으로 받은 전력 차이를 npz 파일에 저장한다. 학습 데이터가 모두 준비되면 CNN을 학습시키고, 학습이 완료되면 모델은 JSON 파일로 저장하고, Weight는 H5 파일로 저장한다.



[그림 4. 학습한 데이터] [그림 5. 가공한 데이터]

4. 결론

본 논문에서는 자율주행을 위해서 신경망을 구축하고 그에 따라서 학습데이터를 만들어내서 모형자동차의 주행을 가능하게 했다. 실제 자동차의 경우에는 고려해야 할 사항이 많기 때문에 많은 연구가 필요하다.

ACKNOWLEDGMENT

“본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW 중심대학지원사업의 연구결과로 수행되었음”(2018-0-00213)

참 고 문 헌

- [1] NVIDIA: End-to-End Learning for Self-Driving Cars
Fabian schilling: The Effect of Batch Normalization on Deep Convolutional Neural Networks
- [2] Microsoft Research: Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification