

# End to End 딥러닝 기반의 자율주행을 위한 실세계 환경을 반영한 가상 주행 데이터 수집 및 활용

김준태, 배창석  
대전대학교 전자정보통신공학과  
e-mail: [kjt7889@naver.com](mailto:kjt7889@naver.com), [csbae@dju.kr](mailto:csbae@dju.kr)

## Collecting and utilizing virtual driving data reflecting real-world environment for autonomous driving based on End to End deep learning

Jun-Tae Kim, Changseok Bae  
Dept. of Electronics, Information, and Communication Eng., Daejeon University

### 요 약

최근 인공지능 연구가 활발하게 진행이 되면서 여러 기업에서 자율 주행연구도 활발하게 진행되고 있다. 하지만 실제 상황에서 자동차 주행 데이터를 얻기에는 여러 위험사항들과 경제적인 낭비가 있다. 그렇기 때문에 게임 상에서 데이터를 수집하고 딥러닝을 이용해 학습을 하기로 했다. 본 논문에서는 실제 세계와 유사한 환경을 가지고 있는 자동차 게임을 이용하여 자율 주행을 시도 했다. 자율 주행 시 많이 쓰이는 End to End 방법으로 데이터를 수집하면 두 가지 데이터가 저장된다. 하나는 이미지 데이터고 두 번째는 방향키 데이터다. 이러한 데이터들을 numpy 타입으로 40분간 데이터를 수집한 후 딥러닝에 많이 쓰이는 tensorflow를 사용하여 구현한 CNN을 이용하여 학습이 되는 것을 확인을 하고 91.9%의 정확도를 얻었다. 이를 기반으로 실세계에서의 사용 가능성을 확인했다.

### 1. 서론

자율 주행이란 용어는 1960년대에 개념이 생기고 1970년대에서부터 연구가 시작되었다 [1]. 하지만 이 당시 자율 주행이라 하면 차선을 검출해 중앙선을 넘지 않고 주행하는 수준이었다 하지만 1990년대에 들어서는 컴퓨터의 기술이 급진전되어 장애물이 개입되는 자율 주행 분야가 연구되기 시작하였다. 최근에는 인공지능 분야가 활발하게 연구되어 자율 주행 자동차에도 적용되고 있다. 딥러닝이 자율 주행 자동차에 적용이 되어 카메라로 전방에 있는 물체를 인식하기도 하고 스스로 판단을 해 어떠한 상황이 왔을 때 스스로 판단하여 주행을 하기도 한다. 하지만 이런 판단을 하기 위해서는 학습을 시킬 수많은 데이터가 필요하다. 하지만 이러한 실제 주행 데이터를 얻기에는 많은 시간과 비용이 들어가기 마련이다. 그렇기 때문에 이 논문에서는 현실세계와 가장 비슷한 환경을 가지고 있는 자동차 게임을 통해 주행 데이터를 얻기로 했다. 이 게임을 이용해 비, 눈, 바람, 맑음, 안개등의 날씨를 사용자 마음대로 조절할 수 있고 새벽, 낮, 밤 등의 시간도 조절을 해 원하는 환경을 조절을 만들어 학습시킬 데이터를 만들 수 있다.

본 논문에서는 자동차 게임을 이용해 Nvidia에서 발표한 자율 주행 기법인 End-to-End [2] 방식으로 학습 데이터를 수집한 다음 패턴인식 분야에서 많이 사용되는 CNN [3] 모델 중 하나인 Alexnet [4]을 이용해 데이터를 학습시킨 후 실제 게임 주행에 적용을 해 본다.

본 논문 2장에서는 학습용 실세계 주행데이터를 수집하는 방법을 설명하고 3장에서는 CNN을 이용하여 학습하고 마지막 4장에서는 학습된 모델을 가지고 테스트하는 방법을 설명한다.

### 2. 실세계 환경 학습용 주행 데이터수집

주행 데이터를 End-to-End 방식으로 수집하기 위해서는 우선 게임 화면을 프레임 단위로 캡처를 해주어야 한다. 캡처를 해줄 때에는 파이썬에서 이미지 처리 할 때 많이 사용하는 PIL (Python Imaging Library) 패키지를 사용해서 캡처를 해줘도 되지만 PIL을 사용한다면 딜레이 지연 현상이 발생하므로 python에서 지원하는 Window API인 pywin32 [5]을 사용해주다. 캡처를 해주는 동시에 방향키 값 [6]도 같이 넣어 줘야 하므로 키보드 입력 값을 받는 Window API를 사용해서 입력을 받아 넣어준다.

표1과 같이 Label은 게임 상에서 방향키인 W:직진 A:좌회전 D:우회전키를 입력한다. 입력할 때 python에서 행렬을 쉽게 다룰 수 있게 해주는 패키지인 numpy [7] 형식으로 대입을 시켜주면 [A, W, D] 형태로 저장이 된다. 표

※ 이 논문은 2016년도 정부 (교육부) 재원으로 한국연구재단의 지원을 받아 수행된 이공학개인지초연구지원사업 (NRF-2016R1D1A3B03933964) 연구 결과입니다.

2와 같이 0번째 열에는 이미지 값 이 들어가고 1번째 열에는 방향키 값을 넣어 준다.

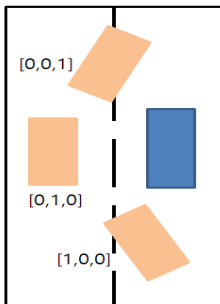
<표1> 입력되는 라벨

키값	실제 입력 라벨
A	[1, 0, 0]
W	[0, 1, 0]
S	[0, 0, 1]

<표2> 저장되는 데이터

이미지 데이터	라벨
[200, 200, 200, 199, 199, 199, 198, 198,...]	[0, 1, 0]
[198, 200, 195, 199, 199, 199, 199, 198, 198,...]	[0, 0, 1]
[192, 191, 191, 191, 191, 191, 191, 191, 190,...]	[1, 0, 0]
[197, 197, 197, 196, 196, 196, 196, 196, 196,...]	[0, 0, 1]

예를 들어 이미지에서 전방에 차가 없고 직직만 한다면 이미지 값과 직진 방향인 w 값인 [0, 1, 0] 값이 들어간다. 하지만 전방 2차선에 차가 있어 추월해서 가야한다면 그림 1 에서와 같이 왼쪽으로 가는 이미지와 좌회전 방향키인 A 추월하기 위한 직진 w 다시 2차선으로 들어오기 위한 D값이 [1, 0, 0] [0, 1, 0] [0, 0, 1] 라벨로 들어가는 것이다. 또한 데이터를 최대한 줄여주기 위해 이미지를 gray값으로 바꾸어 주고 이미지 사이즈를 160x120으로 지정하였다. 본 게임의 장점 중에 하나가 그림2 에서와 같이 비, 바람, 눈, 안개 등의 여러 날씨를 설정하여 데이터를 수집할 수 있다. 날씨가 화창한 날에는 도로의 상태가 문제없을 것이고 비가 오는 날에는 좀 더 미끄러움에 조심해서 달려야 할 것이고 눈이 와 도로에 눈이 쌓이거나 도로가 빙판이 된 날에는 매우 조심해서 달려야 할 것이다. 또한 낮, 밤 등의 시간을 지정해 데이터를 수집할 수 있다. 본 게임은 이러한 날씨, 시간, 각종 차량 등의 변수를 게임 상에 적용시켜 환경을 만들어 놓았기 때문에 자율주행 자동차의 데이터를 만들기에 적합하다.



(그림 1) 추월 시 입력되는 라벨 값



(그림 2) 게임 상에서의 자연환경 조절

### 3. 가상 환경 주행데이터 학습

첫 번째로 저장된 데이터들을 numpy 내장 함수인 load()를 사용해 데이터들을 불러온다. 데이터를 학습시키기 전에 우선 데이터들을 균일하게 해주기 위해 numpy 에 속해있는 배열에 들어가 있는 값을 섞어주는 shuffle [6]함수를 사용해서 데이터들을 섞어준다. 두 번째로는 섞어준 데이터들을 학습 데이터와 테스트 데이터를 나눠 줘야 한다. 최종적으로 주행 데이터를 실제 상황에서 쓰기를 원하므로 가상세계의 주행 데이터와 실제세계의 데이터의 차이가 가능한 없어야 한다. 그러므로 RGB 값이 들어있는 데이터보다는 흑백처리를 해 최대한 가상세계와 현실세계의 차이를 최대한 최소화 한 후 학습을 시켜준다. 이 논문에서는 제한된 컴퓨팅 파워로 인해 표 3과 같이 이미지의 크기를 길이 160 높이 120의 이미지로 변환시킨 후 그림 3에서와 같은 Alexnet [4]을 사용하였다. 학습시간은 18 시간 정도가 걸렸다. 또한 학습된 모델은 테스트를 위해



## 5. 결론

앞으로 자율주행이란 분야는 더 연구 되어야하는 부분이 있고 자동차의 블랙박스 주행영상데이터와 자동차 핸들 각도 값만으로 자율주행을 학습시키기에는 많은 시간과 많은 비용을 투자해야 한다. 이 논문에서 사용한 자동차 게임에서도 데이터를 수집할 때 실제 운전이라고 생각하고 운전을 해야 어느 상황에서 우회전을 하는지 좌회전을 하는지 판별이 명확하게 해야 좋은 데이터를 얻을 수 있다. 또한 가상환경과 실제 환경과의 차이를 고려해 많은 데이터를 학습시켜야 실제 환경에서 높은 정확도를 얻을 수 있다. 또한 차량의 종류가 한 가지만 있는 것이 아니라 버스, 트럭, 승용차 등등 많은 차종의 데이터를 수집해 학습하기에는 어려움이 많다. 앞으로 이런 오락용 게임이 아닌 실제 자동차환경과 실제 도로환경을 시뮬레이션 환경을 구축 한다면 많은 차들의 주행데이터를 얻을 수 있고 실제 환경에서 얻는 것 보단 좀 더 안전하고 많은 시간과 비용을 절약 하면서 수집하기에 쉬울 것이다.

앞으로 이미지 크기를 확장하고 그레이 영상의 학습 데이터를 RGB로 바꾸고 학습 데이터를 좀 더 수집한 뒤 지도학습을 한 뒤 강화학습을 적용을 해 좀 더 실제 환경과 비슷하고 네비게이션 경로를 입력하면 안전하게 목적지까지 갈 수 있는 모델을 만드는 것이 앞으로의 연구 방향이다.

## 참고문헌

- [1] 자율주행자동차 개요  
url:<https://namu.wiki/w/%EC%9E%90%EC%9C%A8%EC%A3%BC%ED%96%89%20%EC%9E%90%EB%8F%99%EC%B0%A8>
- [2] Karol Zieba, Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D.Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao,  
*End to End Learning for Self-Driving Cars*
- [3] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural Computation*, 1(4):541 - 551, Winter 1989.
- [4] Alex Krizhevsky, Ilya Sutskever, Geoffrey E.Hinton  
*ImageNet Classification with Deep Convolutional Neural Networks*
- [5] pywin32 documetation  
url:[timgolden.me.uk/pywin32-docs/contents.html](http://timgolden.me.uk/pywin32-docs/contents.html)
- [6] *Simulate Python keypresses for controlling a game*  
url:<http://stackoverflow.com/questions/14489013/simulate-python-keypresses-for-controlling-a-game>
- [7] Wikipedia Numpy  
url:<https://en.wikipedia.org/wiki/NumPy>