

## 레이싱 게임(카트라이더) 자율주행 인공지능 개발

201524604 최준혁

201524568 정상원

201624416 김기서

### 과제 배경

현실에서 사용가능한 자율주행 프로그램을 개발하기 위해선 많은 양의 데이터와 테스트가 필요합니다. 하지만 시공간의 제약과 안전상의 이유로 실제로 데이터를 수집하거나 테스트를 진행하기는 어렵기에, 여러 기업들은 시뮬레이터를 이용해 데이터 수집과 테스트를 수행하고 있습니다. 레이싱 게임은 이러한 시뮬레이터 환경과 유사한 특징들을 가지고 있기 때문에, 졸업 과제를 통해 레이싱 게임 내에서 잘 동작하는 자율주행 인공지능을 만들어 보는 것은 많은 도움이 될 것으로 판단되어 주제를 선정하였습니다.

### 진행방안

우리 팀은 이 주제로 졸업 과제를 수행하기 위해 인터넷을 통해 자료조사를 진행하며 대략적인 방법과 데이터들을 준비해 나가고 있습니다. 아래는 현재까지 구상하고 있는 방안과 관련 자료들입니다.

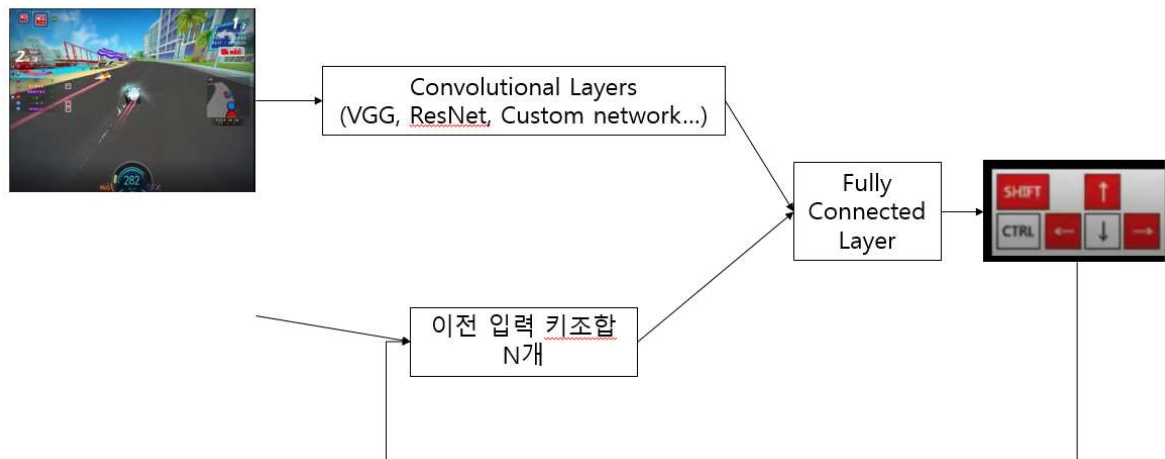
<https://github.com/MattYoon/Autonomous-Kartrider>

위 링크는 건국대 학생들이 몇달 전 카트라이더 게임 내에서 자율 주행 인공지능을 개발한 내역입니다. 이 프로젝트에서는 강화학습의 DQN, HER, DDPG, PPO2 베이스 모델들과 OpenAI gym 환경을 이용해 개발을 진행하였으며, 입력값으로 게임 화면에서의 미니맵을 OpenCV로 일부 보정한 결과에서 특징이 될 만한 점 5개(차와 중앙 사이의 거리, 속도, 길의 휘어짐 정도, 차 축의 각도, 길 축의 각도)를 직접 찾아내어 사용했고 그 결과로 3개의 방향(전진, 우회전, 좌회전)을 받아 반영하는 것을 확인할 수 있습니다.

그러나 이러한 접근방식의 한계로는 카트라이더의 특징적인 시스템인 드리프트와 부스터를 활용하지 못하며, 전체 화면이 아닌 미니맵만 보고 주행을 하고 있어 결과물(깃허브 내 유튜브 링크 참조)이 실제 플레이어에 미치지 못한다는 점이 있습니다. 이러한 점은 강화학습의 특성상 대량의 학습이 필요한데 Atari 게임이나 바둑과 달리 카트라이더는 서비스중인 게임이라 학습에 현실적인 어려운 점들이 있어 선택의 폭을 제한한 것으로 추측됩니다.

우리 팀은 이러한 점에서 차별점을 두어 강화학습이 아닌 지도학습을 기반으로 한 End to End 모델로 해당 과제를 풀어보려 합니다. 카트라이더를 하는 플레이어를 모델링 해보았을 때, 플레이어는 게임 화면의 정보를 받아 총 6개 키의 입력(방향키 상, 하, 좌, 우, Shift, Ctrl)을 선택하게 됩니다. 이러한 키의 조합은 총  $2^6=64$ 개이므로, 우리 팀은 카트라이더 게임에서의 플레이어를 화면의 한 프레임을 입력 받아 매 프레임마다 64개 중 하나를 선택 또는 분류(Classification)하는 모델로 볼 수 있다고 판단했습니다. 따라서 우리 팀은 이미지를 입력받아 64개의 레이블 중 하나를 선택하는 CNN Classifier를 일부 변형하여 과제를 수행할 예정입니다.

현재 구상중인 모델의 대략적인 구성은 다음과 같습니다.



이미지를 Convolutional Layer를 통과시켜 특징 벡터로 바꾸고, 이 벡터만으로 분류를 진행하면 키 입력을 평균적인 선택으로만 진행할 것으로 예상되기 때문에 이전의 키 입력 조합들을 받아 함께 Fully Connected Layer의 입력으로 넣어 분류를 진행하는 모델입니다. Convolutional Layer는 기존의 모델들을 사용할 수도 있고, 모델 크기로 인한 성능 이슈가 발생할 경우 몇개의 층으로 쌓은 Custom Convolutional Layers를 사용할 것을 생각하고 있습니다. 또한 Fully Connected Layer의 입력으로 부스터 잔량, 부스터 게이지 수치, 속도, 등수 등을 OpenCV를 통해 추출하여 사용할 수도 있으며, 성능에 따라 CNN+RNN 형태의 모델을 사용할 수도 있을 것으로 예상됩니다.

데이터 수집은 인터넷 방송을 통해 수집할 것을 생각하고 있습니다.

<https://www.twitch.tv/mhj1682/videos>

위 링크는 문호준 카트라이더 전 프로 선수의 방송 링크입니다. 방송 중 화면을 보시면 화면 하단에 방향키와 Shift, Ctrl 키의 누름 상태를 볼 수 있는 키뷰어가 활성화되어 있기 때문에, 이를 이용해 모델을 학습시킬 수 있을 것으로 예상하고 있습니다. 실제로 트위치를 통해 데이터를 일부 모았을 때 초당 60프레임의 영상으로 얻을 수 있었고 이를 0.05~0.1초 단위로 끊어서 이미지 형태로 만들었을 때, 평균적인 한 경기가 2분이므로 약 1200(~2400)장 정도의 이미지 데이터를 얻을 수 있었습니다. 이를 OpenCV나 다른 이미지 처리 프로그램을 통해 데이터로 가공하면 모델을 학습시키기에 적합한 형태의 데이터로 가공할 수 있을 것으로 보고 있습니다.

## 예상되는 문제점

현재 예상되는 문제점으로는 첫째로, 모델이 매 0.05초~0.1초마다 추론을 수행해야 하는데, 카트라이더 클라이언트 프로그램을 실행하면서 동시에 이 정도의 속도를 낼 수 있는지에 관한 것입니다. 강화학습 프로젝트에서는 mss 라이브러리를 통해 화면을 추출하고 pywin32, pyautoit 라이브러리로 입력을 처리하고 매 step마다 0.025초의 딜레이를 주고 있는 것을 확인할 수 있었는데, 강화학습 모델을 CNN 기반 모델로 바꿨을 때 이 속도가 그대로 나올지는 프로젝트를 진행하며 모델의 크기나 컴퓨터 자원 등을 고려해 결정하게 될 것 같습니다.

둘째로, 해당 프로그램이 다른 유저들과 함께하는 멀티플레이에 사용될 경우 다른 유저들에 대한 일종의 치팅(cheating)이 될 수 있다는 점입니다. 이 점은 모델의 학습 및 테스트를 진행할 때 반드시 사람이 아닌 프로그램끼리의 경쟁으로 진행하며, 혹시 유저와 겨루게 될 경우 반드시 사전에 공지하는 방식으로 프로젝트를 진행하여 어느정도 해결할 수 있을 것으로 예상하고 있습니다.

## 추후 진행방안

카트라이더 방송인들의 주행을 보고 학습을 진행시켜 어느정도 성과가 나온 뒤에도 프로젝트를 여러가지 방법으로 발전시킬 수 있을 것으로 예상됩니다. 가장 간단히 생각해 볼 수 있는 방법으로는, 같은 모델로 서로 경쟁하여 이긴 쪽의 결과로 모델을 튜닝하는 기초적인 강화학습을 생각해 볼 수 있습니다. 또한 학습의 목표가 단순히 classification accuracy rate를 높이는 것이 아니라 게임을 잘 하는 모델을 만드는 것이므로, 학습 데이터의 등수, 승패여부, 경기 시간 등에 따라 learning rate를 조절해 가중학습 시키는 것도 고려해 볼 수 있을 것입니다. 이외에도 프로젝트를 진행하며 시간/자원적 여건에 따라 여러 아이디어를 추가적으로 적용해 볼 수 있을 것입니다.