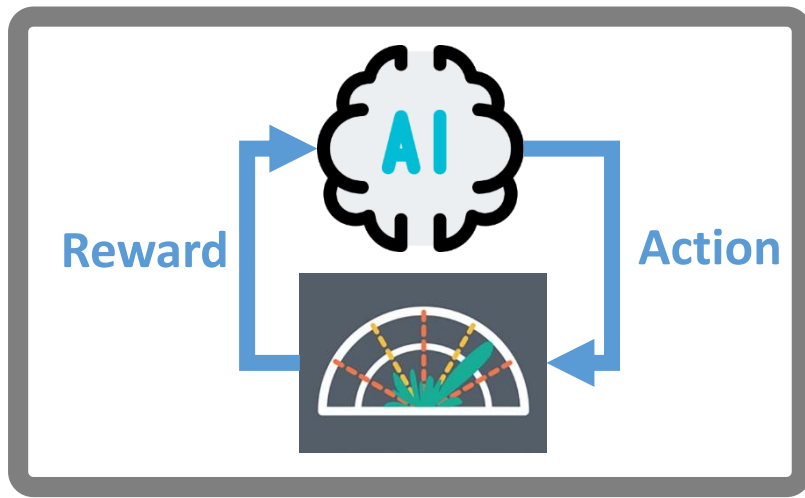


Beam Tracking using Deep Reinforcement Learning

최민우(20141477), 박지원(20141429), 현슬아(20151514)

서강대학교 전자공학과



배경

- 기존 빔포밍 시스템에 AI 딥러닝 기술을 적용해 효과적으로 user를 tracking하는 시스템 설계.
- 수신자로부터 주기적으로 Feedback을 받고 이를 통해 송신 빔을 조정하는 방식이 RL과 유사.
- RL의 단점을 극복하고자 CNN 모델과 DDQN 알고리즘을 사용.

게임 룰

Step 1: 게임 환경

32 frame마다 새 시나리오가 시작.
30°~150° 내에서 움직임.
실제 통신환경과 유사.

Step 2: 유저의 움직임

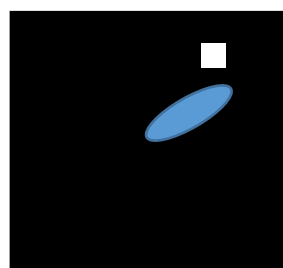
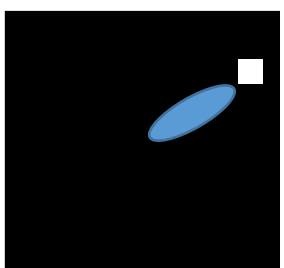
유저는 랜덤위치시작 랜덤이동.
4 frame마다 left, pause, right
중 하나의 action을 선택.

Step 3: 빔의 움직임

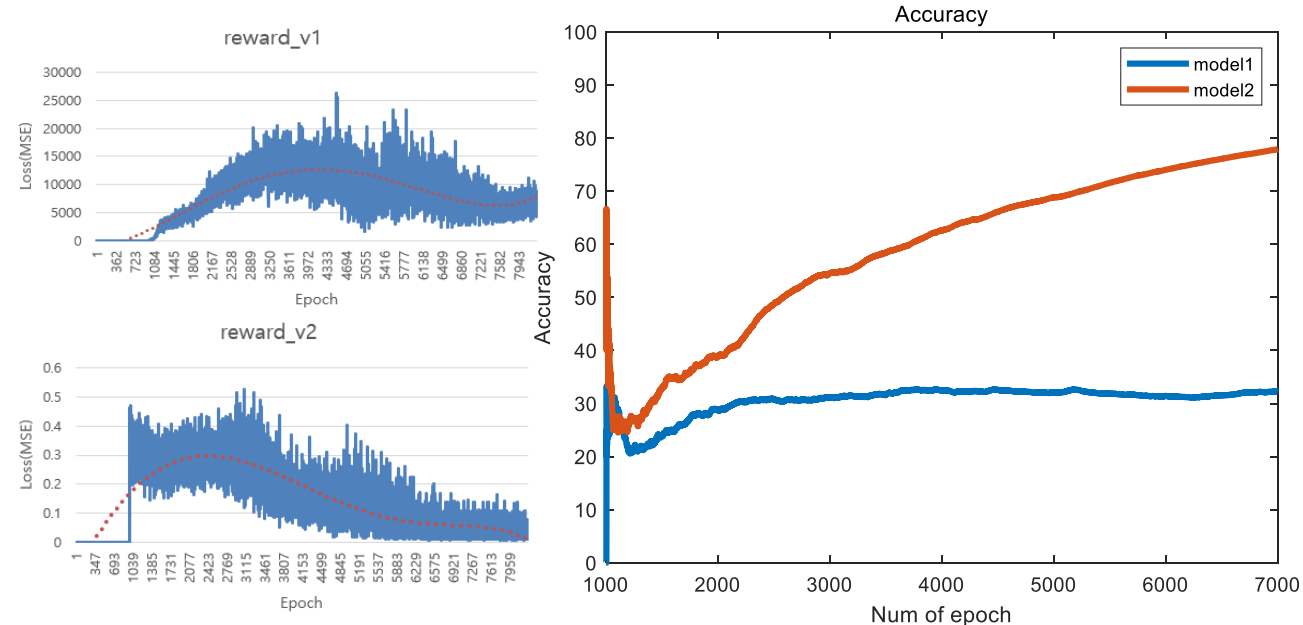
빔은 90°에서 시작.
Model의 prediction에 따라
1 frame마다left, pause, right
중 하나의 action을 선택.

Step 4: 리워드

마지막 32frame에 reward 지급.



Positive feedback : +1점 Negative feedback : -1점



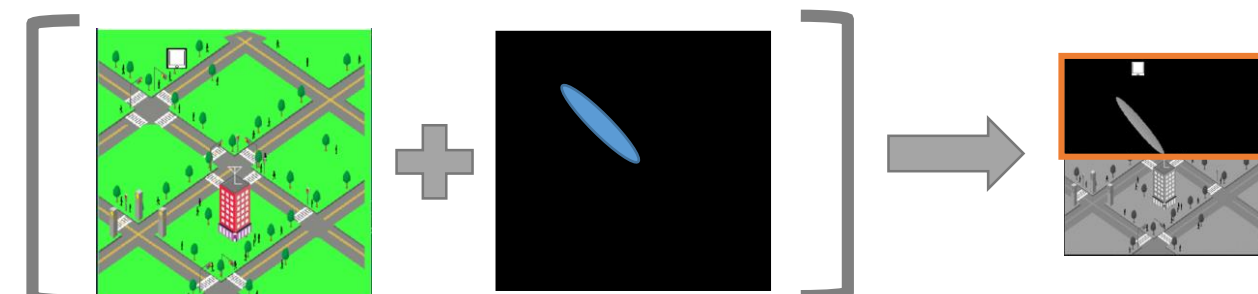
Reward_v1 : 매 frame마다 reward 지급
Reward_v2 : 마지막 frame에만 reward 지급
*Accuracy : 현재까지 진행한 게임 횟수 대비 총 이긴 게임 횟수

모델 구조

CNN(Convolutional Neural Network)

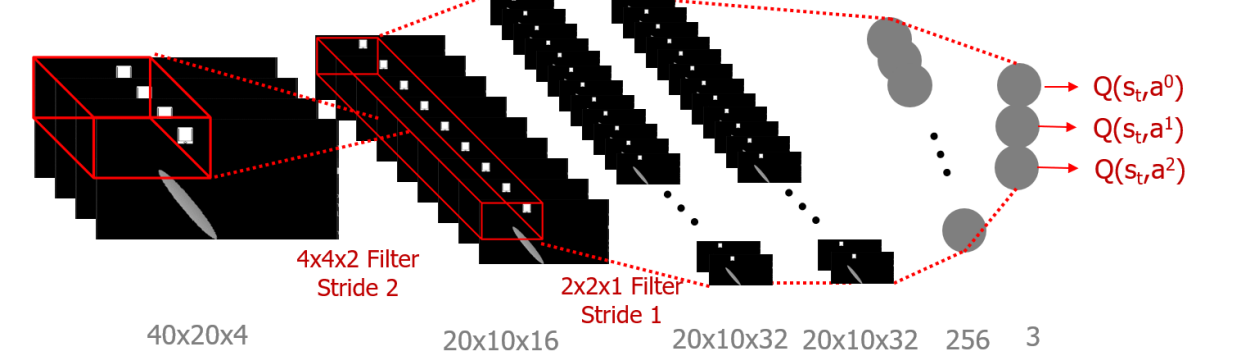
시각적 이미지를 layer로 쌓아서 분석하는 네트워크.

화면 구성



- 1) User 위치 정보가 담긴 영상데이터에 Beam의 정보를 합성.
- 2) 전처리과정(Resize & Extraction)을 거침.

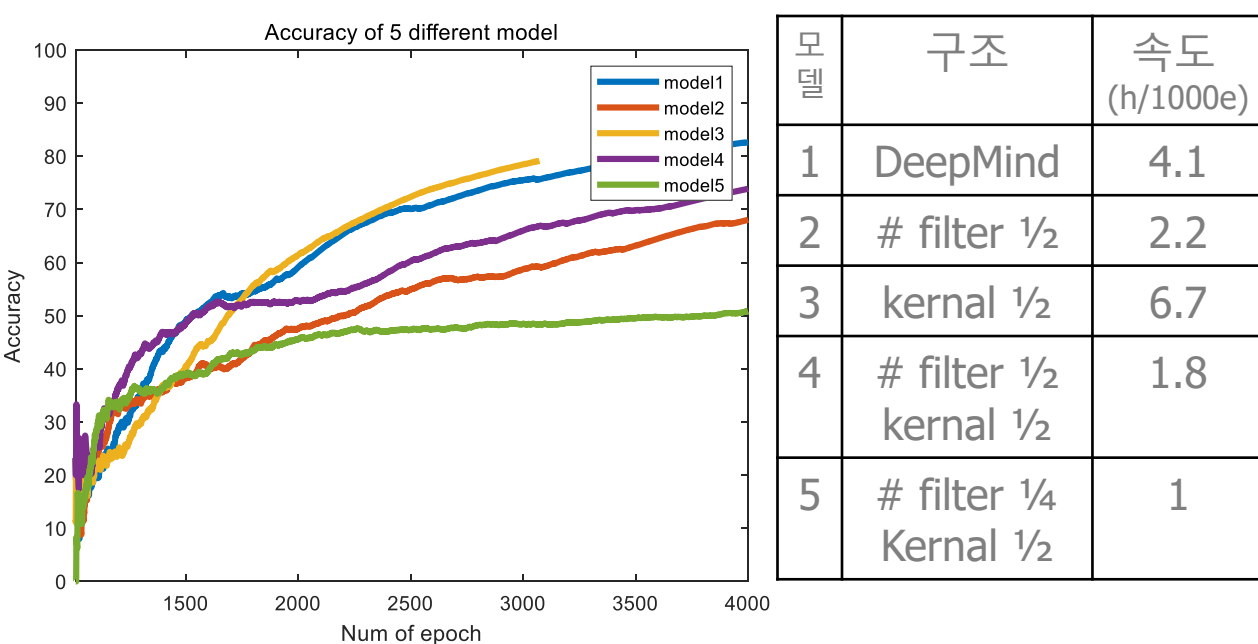
구조



- Input : 전처리한 이미지 프레임 x 4개
- Layer : Conv x 3, Flatten x 1, FC x 2

Layer	Input	Num filters	Filter size	Stride	Activation	Output	Num Parameter
Conv1	40x20x4	16	4x4	2	ReLU	20x10x16	1,040
Conv2	20x10x16	32	2x2	1	ReLU	20x10x32	2,080
Conv3	20x10x32	32	2x2	1	ReLU	20x10x32	4,128
Flatten	20x10x32					6,400	
Fc4	12800				ReLU	256	1,638,656
Fc5	256				Linear	3	771

- Output : Q-value(=Action-value)



알고리즘

DQN(Double Deep Q-learning Network)

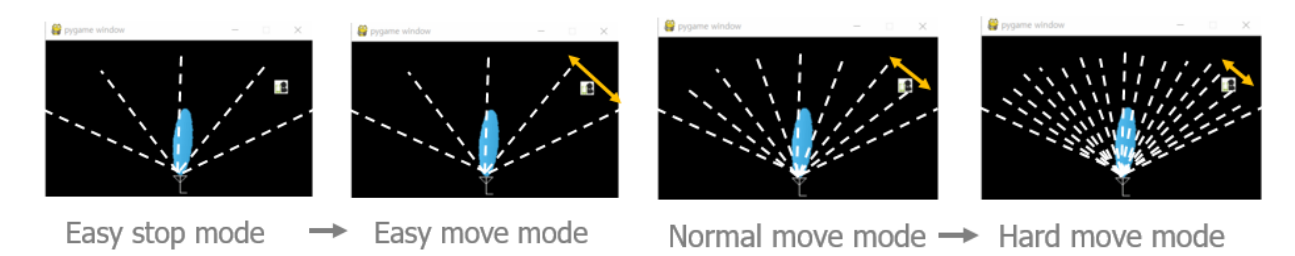
- 1) Experience Replay (Replay Memory)
- 2) Preprocessing
- 3) Exploration & Exploitation
- 4) Fixed Q-targets

DDQN(Double Deep Q-learning Network)

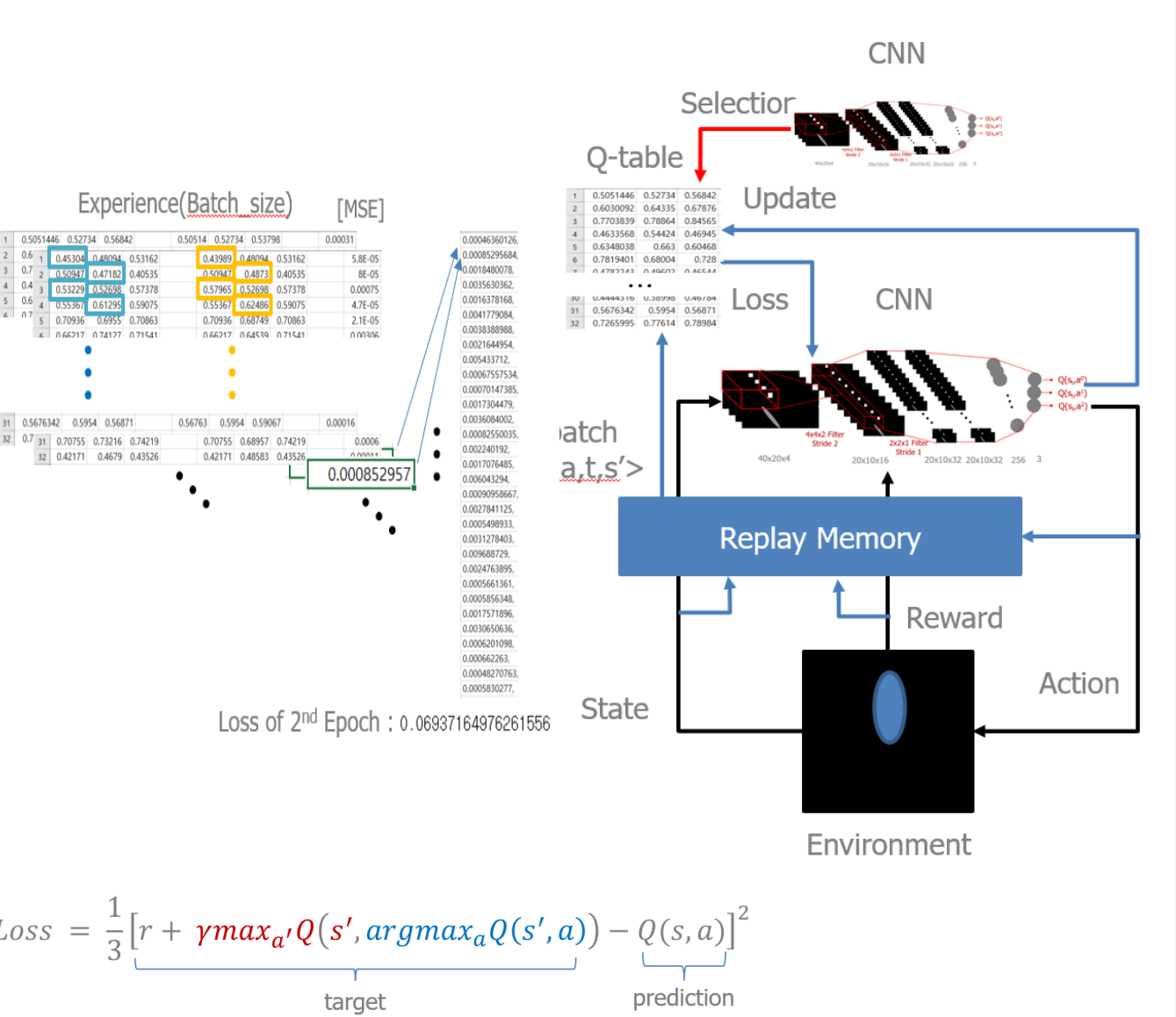
DQN에 double estimator Algorithm을 추가.

Gradual Learning

쉬운 난이도의 게임부터 학습하고
학습된 model weights를 점점 어려운 게임에
load하여 가면서 학습.



실제 게임 알고리즘(개념도)



연구 결과

결과

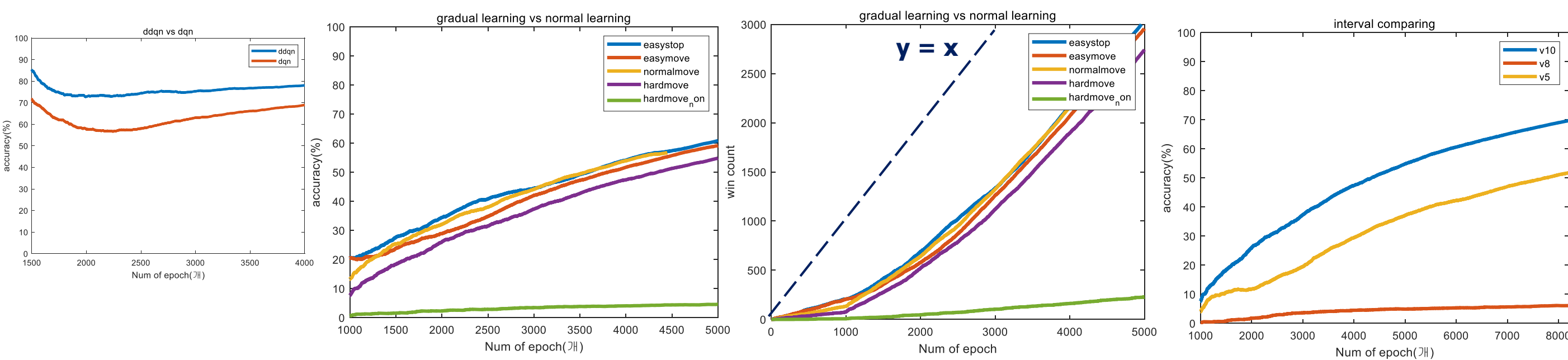


사진 1

사진 2

사진 3

사진 4

- 사진 1 : DQN 대비 Double DQN의 성능 비교
- 사진 2 : 점진적으로 학습(Gradual Learning)했을 때와 일반적으로 학습했을 때의 성능 비교(Accuracy)
- 사진 3 : 점진적으로 학습(Gradual Learning)했을 때와 일반적으로 학습했을 때의 성능 비교(Win count)
- 사진 4 : Gradual Learning할 때 load한 model과 overlap되는 state가 없다면 성능 개선 효과가 거의 없음.
- * Win count : 총 이긴 게임 횟수 , Accuracy : 현재까지 진행한 게임 횟수 대비 총 이긴 게임 횟수

성과

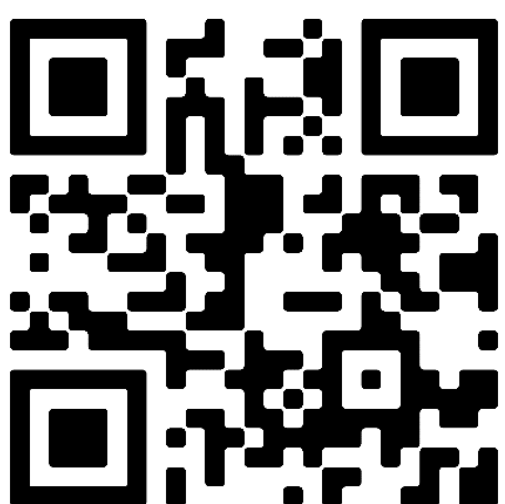


<SU-MIMO>

<MU-MIMO>

결론 및 의의

- 1) 기존의 빔포밍 시스템에 딥러닝 기술을 사용하여 유저를 트래킹하는 성능을 향상시킬 수 있었다.
- 2) 향후 MU-MIMO 시스템으로의 활용도 가능성을 확인하였다.
- 2) Hyper-parameter를 바꿔 가면서 학습의 효율을 높일 수 있는 게임의 알고리즘을 결정했다.
- 3) 5G 통신 환경에서 강조되는 저지연 통신을 구현할 방법을 제안하였다.



Code

2019 전자공학과 디자인 프로젝트