아주대학교
세상의 A+가 되겠습니다

2023-2
AJOU SOFTCON

인공지능에게 눈을 주다

팀 명 AI 사냥꾼 **팀 원** 유승민, 노진환, 김범기 **지도교수** 유종빈

개발 동기 및 목적

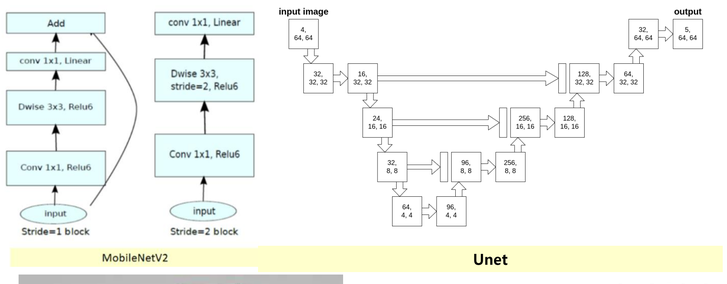
- Unity와 같은 가상 시뮬레이션 제작 툴은 인공지능 연구에 유용함
- Unity도 인공지능 연구에 도움이 되는 플러그인을 다량 제공함
- 그 중에는 ML-Agent라는 강화학습 플러그인이 있음
- ML-Agent에선 또한 강화학습에 관한 다양한 예제를 제공함
- 하지만 Computer Vision과 관련된 예제는 딱 한 가지며 매우 단순함
- PyTorch를 ML-Agent에 적용하면 정교한 Task를 수행할 수 있다고 생각
- 관련된 Computer Vision Task를 원했으나 자료가 없음
- 따라서 우리는 직접 ML-Agent와 PyTorch를 연동하는 Task를 진행함
  1. '이미지'를 학습 데이터로 사용하는 어려운 강화학습 환경 구현
  2. 강화학습 환경을 더 어렵게 만든 후, PyTorch로 성능 개선

➢ 궁극적인 목적은 프로젝트를 정리해 오픈소스로 공개
➢ 후배들의 Computer Vision 공부에 보탬이 되는 것

주요기술

Unity ⟷ Flask ⟷ PyTorch

- Flask API로 HTTP 통신망을 구축해 이미지를 송수신



- Mobile net V2 으로 원본 이미지를 인코딩
- Unet으로 Semantic Segmentation 수행

- ML-Agent로 강화학습을 진행
- 원본 이미지로 학습하는 Fig.1, Fig. 2
- Segmentation 이미지로 학습하는 Fig.4
- 학습 없이 Fig.2에서 Fig.1모델을 사용하는 Fig.5

개발 내용

- 강화학습은 Unity ML-Agent에서 제공하는 예제 'Pyramids' 로 진행
- 'Pyramids'의 기존 학습 데이터는 단순한 '센서'임
- 우리 목적에 맞게 'Pyramids'의 학습 데이터를 '이미지'로 변경

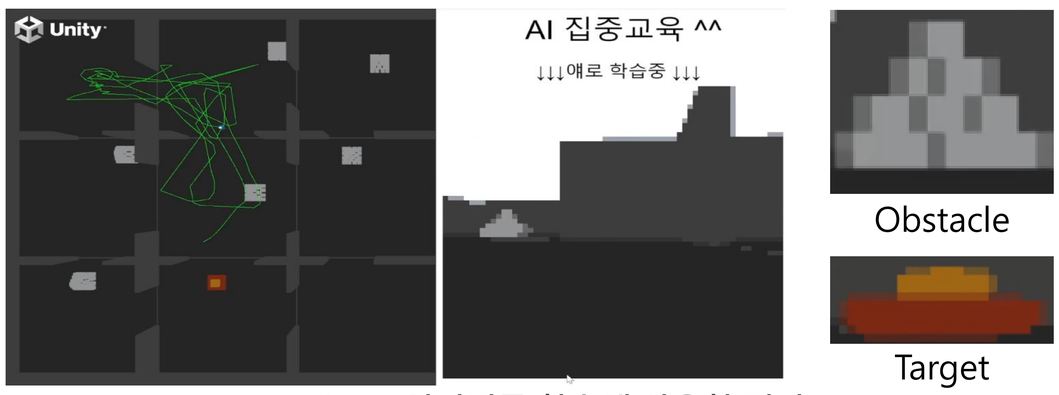


Fig.1. 이미지를 학습에 사용한 결과

- Agent는 미로 속에 숨겨진 Target을 눈으로 찾음
- 미로에 남겨진 초록색 선이 Agent가 약 3분 동안 이동한 경로
- 벽이나 Obstacle 뒤로 이동해 주변을 둘러보며 Target을 물색하길 반복
- Fig.1과 같이 간단한 환경에선 인공지능에게 눈을 주는데 성공

Fig.2. 더 복잡한 환경에서 이미지를 학습에 사용한 결과

- Fig.2에서는 Fig.1의 미로에 텍스처를 적용, 더 이상 눈으로 Target을 찾지 못 함
- 경로를 보면, 불 꺼진 방에서 스위치를 찾듯 최대한 많은 구역을 더듬으며 움직임
- 이 결과는 학습 데이터로 아무것도 주지 않아도 똑같이 도출됨
- 즉, 학습에 이미지를 사용한 것이 아님. 이래선 이미지를 사용하는 것이 의미 없음
- Fig.2와 같은 복잡한 환경에선 인공지능에게 눈을 주는데 실패

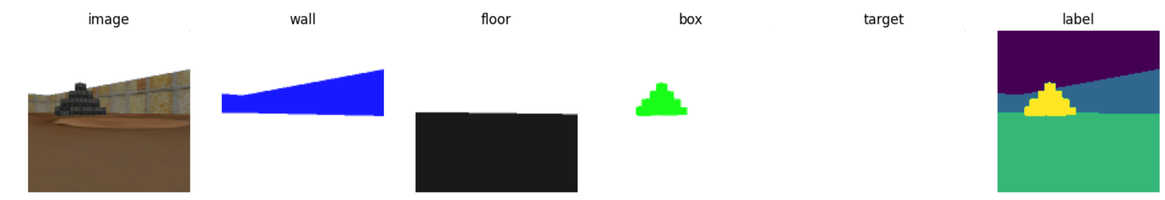


Fig.3. Segmentation Model 학습에 사용할 Dataset 구축

- 복잡한 이미지는 강화학습에 사용되는 의미가 없음
- Image Segmentation을 이용해 복잡한 이미지를 단순하게 바꿔야 함
- 12,868 장의 원본 이미지 생성, 12,868 장의 라벨 이미지 생성
- 라벨 이미지 생성에 12,868*4 장의 이미지 생성과 Shading 과정을 거침
- Model은 인코더로 Mobile net V2, 디코더로 Unet을 사용. 학습 진행

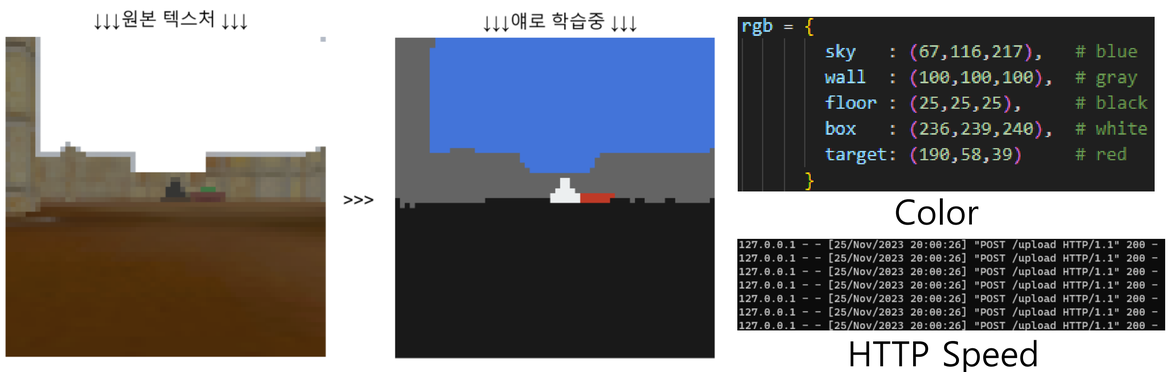


Fig.4. HTTP 통신과 Image Segmentation

- Fig.4에선 이미지를 Segmentation 처리 후 학습에 사용
- Legion 5 pro 노트북에서 초당 15번의 Image Segmentation이 가능
- 강화학습에 소요되는 시간이 약 30배 증가. Fig.1과 유사한 학습 결과
- 엄청난 컴퓨터 리소스를 요구함. 개인 차원에서 감당하기 힘듦
- 긴 학습 후, Fig.4에선 복잡한 환경임에도 인공지능에게 눈을 주는데 성공함

Fig.5. 최종

- 두 모델 모두 Pre-Trained된 모델을 사용. 따라서 학습이 필요 없음
- Fig.5에선 매우 적은 컴퓨터 리소스만 사용, 인공지능에게 눈을 주는데 성공
- 그런데 시력이 안 좋음. Target은 인식하나 벽 뒤를 수색하는 능력 없음
- 가까운 물체에 대한 Segmentation이 불안정
- Fig.4의 Segmentation 모델을 적용, Fig.1의 강화학습 모델을 사용

결과 및 분석

	Fig.1	Fig.2	Fig.4	Fig.5
학습시간(h)	1	2	60	X
수색능력	O	X	X	X
Target인지	O	X	O	O

Fig.6. 결과

- Fig.1 ML-Agent의 기존 예제에서 인공지능에게 눈을 주는 것은 성공
- Fig.2 텍스처를 적용하자 학습 시간이 늘어났으며 눈을 주는 것에 실패
- Fig.4 Computer Vision과 융합해 눈을 주는 것에 성공했으나 시력이 나쁨. 요구하는 리소스가 비약적으로 상승했으며 학습시간이 몇 십 배 증가
- Fig.5 Pre-Trained된 Computer Vision을 융합해 학습을 안 해도 됨. 요구하는 컴퓨터 리소스도 적음. 눈은 줬으나 여전히 시력이 나쁨

+ Unity를 Computer Vision 연구에 사용한 것은 다음의 시사점을 남김

1. Unity를 통해 Dataset을 자유롭게 생성. Vision연구에 최적화됨
2. Unity와 PyTorch를 연동하는데 이슈가 없어 자유롭게 사용 가능
3. Domain Adaption 등의 기술을 적용하여 더 유효해질 여지가 있음

결과적으로 프로젝트 '인공지능에게 눈을 주다 ' 는 Unity로 복잡한 Computer Vision Task를 완료한 선례자료가 됐으며 자율주행차량, 인공지능로봇 등의 더 어려운 Computer Vision Task로 발전시킬 수 있다는 가능성을 남긴다. 오픈소스로써 아주대 후배들이 훌륭한 Task를 수행하는데 도움이 되길 진심으로 바란다

오픈소스 URL

GitHub:
https://github.com/h0dol9805/Artificial_Hunter_Vision

GitHub QR