

도시정책 성능 검증을 위한 디지털 트윈 시뮬레이션 - YOLO와 Unreal Engine을 중심으로

강민주 · 문영서 · 윤지민

하나고등학교

An Digital twin simulation for verifying the performance of urban policies
-focusing on YOLOv5 and Unreal Engine 5

Min-Ju Kang · Young-Seo Moon · Ji-Min Yoon

Hana Academy Seoul

Abstract It is required to confirm the condition of a certain city and population before the government establishes urban policy. Mapo-gu is one of the states that has a high violence crime rate in Seoul, but the lowest crime detection rate compared to other 2 states(Eunpyeong-gu, Youngdeungpo-gu) in Western Seoul. The research team at Hana Academy purposes to set certain hypothetical condition of Gyeongui-line Forest Park, Mapo-gu in Unreal engine 5 and simulate the process from the incidence of crime to getting arrested criminal using YOLOv5, which is a computer vision model that uses object detection to identify object images. The research team at Hana Academy developed a violence response protocol that utilizes AI-enhanced CCTV powered by YOLOv5 to detect violent incidents in a test-bed environment, deploying a Lifeguard to intervene before police arrival. The team testified whether this process and setting work as an efficient test-bed for the future simulation in Unreal engine 5. The team focused more on the condition in Unreal Engine, which was designed for real-time communication using TCP socket in YOLO v5 based on a custom dataset, which simulated the similar condition to the real Gyeongui-line Forest Park. The team concluded that while Unreal Engine 5 was effective for the previous test-bed, it requires further advancements to support realistic urban simulations and the incorporation of additional variants for broader applications.

Key words: Digital Twin, Simulation, YOLO, AI-enhanced CCTV, Crime policy

1. 서론

도시 정책들 중에는 성공적인 사례도, 실패한 사례도 존재한다. 대표적으로 BRT체계를 도입하여 교통 혼잡을 성공적으로 감소시킨 브라질库里치마 시, 단지 설계 단계에서 타길을 잘못 설

정해 슬럼화된 미국의 프루이트 아이고 단지가 있다. 본 프로젝트는 디지털 트윈 시뮬레이션을 통해 도시 정책 도입 전에 그 효용성을 검증할 수 있도록 하여 정책을 효율적으로 수립할 수 있도록 하고자 한다. 이를 위해 “절도, 폭력 범죄 검거율 증가”를 목표로 설정하였고 이를 달성하기 위한 가상의 범죄 대응 정책을 수립하였다. 또한 Unreal Engine 5를 이용해 디지털 트윈 환경을 구축하여 해당 환경 내에서 가상 정책의 실효성을 시뮬레이션했다. 해당 프로세스가 도시정책의 효용성 검증을 위한 테스트 베드 활용의 선례로서 작용하도록 하는 것을 프로젝트의 최종 목적으로 가진다.

2. 이론적 배경

2.1. 성공한 도시 정책과 실패한 도시 정책의 사례 및 시사점

프루이트 아이고 단지는 2차 세계대전 이후에 도시에 원래 있던 중산층 백인 인구가 교외로 빠져나가 슬럼현상이 발생하자 이 백인 인구를 다시 도시로 끌어모으기 위해서 건설된 단지이다. 상승적인 임대료 체납 문제에 의해 단지는 점점 낡고 노후되어 범죄의 중심지가 되었고, 결국 철거되었다. 도시에서 살아가는 사람들에 대한 고려 없이 개발을 진행한 것이 결국 실패로 이어진 것이다.

쿠리치바는 우수한 도로 교통 체계로 유명하다. 도시데이터를 중심으로 대중교통시스템을 개발했고 버스터미널 가까이에 시청, 전기회사, 수도국 등의 굵직한 기관을 배치하여 사람들이 시간을 낭비하는 일이 없도록 도시계획을 했다. 주로 도심에 모여 있기 일췌인 큰 병원이나 쇼핑센터 등도 버스 선로를 옆두에 두고 넓게 분산시켜 많은 인파들이 도심으로 밀려들 수밖에 없는 현상을 사전에 예방했다. 이런 정도의 노력만으로도 교통의 혼잡이 타 도시에 비해 30~40%까지 절감되었다.

상반된 결과를 낳은 두 정책의 사례를 비교해 보았을 때, 정책 시행 시 정책 수립 과정에서 예상하지 못했던 상황이 발생할 수 있다는 문제점을 찾을 수 있다. 해당 문제점에 대한 해결방안으로 디지털 시뮬레이션 기술을 활용할 수 있다. 1940년 군사 및 항공 훈련을 시작으로 시작된 시뮬레이션은 최근 IoT와 빅데이터를 접목하여 실시간 데이터를 반영할 수 있게 되면서 더욱 정확한 분석이 이루어지고 있다. 디지털 시뮬레이션을 통해 정부는 적은 비용을 들여 정책의 실효성을 미리 판단할 수 있으며, 피드백과 시뮬레이션을 반복하는 과정을 거치며 정책의 완성도를 높일 수 있을 것이다.

2.2. 디지털 트윈

디지털 트윈(Digital Twin)은 21세기 시뮬레이션 기술의 혁신적인 진보를 종합적으로 나타내는 개념으로, 물리적 객체나 시스템의 가상 모델을 통해 그 상태와 성능을 실시간으로 반영하는 기술이다. 이는 사물인터넷(IoT), 빅데이터, 인공지능(AI)과의 통합을 통해 가능하며, 현실 세계의 복잡한 시스템을 효과적으로 시뮬레이션하고 분석하는 데 필수적인 역할을 한다.

특히 스마트 시티 구축에 있어 디지털 트윈은 중요한 도구로 자리 잡고 있다. 이는 도시의

교통 흐름, 에너지 소비, 환경 변화 등을 모델링함으로써 도시 계획 및 관리에 실질적인 기여를 할 수 있다. 대표적 예로 도시의 교통 체증을 분석하고, 이를 기반으로 신호 체계를 최적화함으로써 효율적인 교통 관리를 가능하게 한다. 또한, 환경 오염 데이터를 실시간으로 분석하여 지속 가능한 발전 방안을 모색하는 데 기여함으로써, 도시의 환경적 측면을 개선하는 데에도 중요한 역할을 수행한다.

디지털 트윈의 주요 장점 중 하나는 실시간으로 들어오는 데이터를 활용할 수 있다는 점이다. 이는 시스템의 현재 상태를 정확하게 반영하고, 분석 및 예측을 통해 미래의 상황을 미리 시뮬레이션할 수 있는 기반을 제공한다. 또한, 디지털 트윈은 다양한 이해관계자 간의 효과적인 소통을 촉진한다. 가상 모델을 통해 다양한 시나리오를 시각적으로 제시함으로써, 정책 결정자, 시민, 기업 등이 상호작용하며 협력할 수 있는 환경을 조성한다.

결론적으로, 디지털 트윈은 복잡한 시스템의 관리와 최적화를 위한 시뮬레이션 기술 진보의 집합체이며, 다양한 분야에서 그 가능성을 입증하고 있다.

2.3. YOLO 객체 탐지 모델

본 프로젝트에서는 가상의 범죄 대응 정책으로 ‘YOLO 지능형 CCTV 기반 라이프가드 시스템’을 제안한다. 정책의 세부 내용은 4-1. 가상 정책 수립에서 자세히 설명한다.

YOLO는 객체 탐지 분야에서 혁신적인 접근 방식을 제시하는 딥러닝 기반 모델로, 입력 이미지에서 객체를 동시에 인식하고 위치를 예측하는 특징을 가진다. 기존의 객체 탐지 기술, 예를 들어 R-CNN 계열 모델들은 이미지에서 후보 영역을 추출한 후 각 영역에 대해 별도로 분류를 수행하여 탐지하는 방식으로, 처리 시간이 길고 계산 비용이 높은 단점이 있다. 이러한 전통적인 방식은 특히 실시간 응용에서 한계를 나타낸다.

반면, YOLO는 전체 이미지를 단 한 번 분석하여 여러 객체를 동시에 탐지할 수 있도록 설계되었다. 이로 인해 처리 속도가 크게 향상되어, 초당 수십 프레임을 처리하는 것이 가능하다. 이러한 속도는 자율주행차, 실시간 감시 시스템 등 시간 민감성이 요구되는 응용 분야에서 중요한 장점으로 작용한다.

또한, YOLO는 전역 정보를 활용하여 객체의 위치를 예측하므로, 객체 간의 상호작용을 고려한 정확한 탐지가 가능하다. 이는 복잡한 환경에서도 높은 정확성을 유지하게 하여, 오탐지율을 줄이고 정확도를 향상시키는 결과를 초래한다.

결론적으로, YOLO 모델은 속도, 정확성 및 효율성을 결합하여, 기존 기술에 비해 실시간 객체 탐지의 가능성을 획기적으로 확장시킨다. 이러한 특성은 Unreal Engine으로 구현한 시뮬레이션 배경에서 특정 객체의 행동을 감지하고 분별할 때 적합하다고 판단되었다. 높은 정확성과 낮은 오탐지율이라는 장점을 가지고 본 프로젝트에서는 객체탐지에 YOLO모델을 활용하였다.

2.4. 지능형 CCTV

지능형 CCTV는 인공지능 기술을 활용하여 영상 데이터를 실시간으로 분석하고, 이상 행동을

감지하거나 특정 사건을 자동으로 기록하는 시스템이다. 이는 기존 CCTV 시스템과의 차별점이 있다. 기존 CCTV는 단순히 영상을 기록하고 저장하는 기능에 중점을 두며, 사후에 사건을 확인하는 데 주로 사용된다. 반면, 지능형 CCTV는 인공지능 기술을 활용하여 실시간으로 영상 데이터를 분석하고, 이상 행동을 자동으로 감지하는 기능을 갖추고 있다. 예를 들어, 사람의 움직임, 특정 지역 침입, 또는 무단 주정차 등을 실시간으로 모니터링하여 즉각적인 경고를 생성할 수 있다.

이러한 기술은 범죄 예방 및 대응에 유용하다. 예를 들어, 2020년부터 2022년까지 영국 런던의 특정 지역에서 지능형 CCTV를 도입한 결과, 범죄율이 20% 감소한 사례가 있다(출처: London Metropolitan Police). 이 시스템은 고급 이미지 인식 기능을 통해 의심스러운 행동을 실시간으로 탐지하고, 경찰에 즉시 알림을 보내어 범죄 발생을 예방하는 데 효과적이다. 또한, 미국 샌프란시스코에서는 지능형 CCTV 도입 후 30% 이상의 범죄율 감소가 보고되었다(출처: San Francisco Police Department). 이러한 사례들은 지능형 CCTV가 범죄 감시에 있어 혁신적인 역할을 하고 있음을 보여준다. 이 시스템은 경찰의 대응 시간을 단축시키고, 공공 안전을 향상시키는 데 기여하여, 미래의 도시 환경에서 필수적인 요소로 자리 잡을 것으로 예상된다.

2.5. Unreal engine 5

언리얼 엔진(Unreal Engine)은 에픽 게임즈(Epic Games)에서 개발한 고성능 게임 엔진으로, 1998년 첫 출시 이후 다양한 게임 개발자들 사이에서 높은 인기를 얻고 있다. 특히 언리얼 엔진 4와 최신 버전인 언리얼 엔진 5는 실시간 렌더링, 고해상도 텍스처, 뛰어난 애니메이션 기능 등을 지원하여 게임 산업뿐만 아니라 영화, 건축, 자동차 디자인 등 다양한 산업 분야에서도 널리 활용되고 있다. 이러한 특성은 특히 사실적인 그래픽과 물리 효과를 중시하는 분야에서 두드러진 장점을 제공한다.

언리얼 엔진은 도시 설계 분야에서도 중요한 역할을 한다. 실시간 렌더링과 고해상도 텍스처를 통해 도시의 세부 사항을 생동감 있게 재현할 수 있어, 도시 계획자와 건축가들에게 효과적인 도구로 자리 잡고 있다. 언리얼 엔진을 활용하면 빌딩, 도로, 공원 등의 도시 요소를 사실적으로 표현할 수 있으며, 날씨 및 조명 조건을 시뮬레이션하여 다양한 시간대와 계절의 변화를 구현할 수 있다. 이는 도시 계획자와 건축가들이 설계안을 보다 직관적으로 검토하고, 시민들에게 미래 도시의 모습을 현실감 있게 전달하는 데 큰 도움이 된다.

언리얼 엔진을 이용한 도시 시각화의 장점은 다음과 같다. 첫째, 실시간 렌더링 기능은 도시의 세부 사항을 즉각적으로 반영하므로 설계안 검토 과정이 효율적이다. 둘째, 고해상도 텍스처와 사실적인 조명, 날씨 시뮬레이션을 통해 다양한 시간대와 계절의 변화를 시각적으로 표현할 수 있어 도시 계획자와 건축가의 의사결정 과정에 실질적인 기여를 한다. 셋째, 언리얼 엔진의 블루프린트 비주얼 스크립팅 기능은 프로그래밍 지식이 없는 사용자도 쉽게 시각화를 구현할 수 있도록 하여, 기술적 장벽을 낮춘다. 마지막으로, 이러한 시각화는 시민들에게 미래 도시의 모습을 현실감 있게 전달하여 공감대를 형성하는 데 유리한 환경을 제공한다.

결론적으로, 언리얼 엔진(Unreal Engine)은 1998년 출시 이후 고품질 그래픽과 사실적인 물리 효과를 제공하며, 특히 언리얼 엔진 4와 5는 실시간 렌더링과 고해상도 텍스처를 통해 도시 시각화에 적합한 도구로 자리매김하였다. 이러한 기능들은 도시 계획자와 건축가가 설계안을 직관적으로 검토하고 시민들에게 미래 도시의 모습을 현실감 있게 전달할 수 있게 하며, 블루프린트 비주얼 스크립팅을 통해 기술적 장벽을 낮춘다. 이러한 특성을 기반으로 초심자가 진행하는 도시 구현에 적합하다고 판단하여, 본 연구의 시각적, 시스템적 구현을 위해서 Unreal Engine 5를 사용하였다.

3. 선행 연구

3.1. 선행 연구 분석

현대 도시계획에 있어 스마트 시티 개념은 효율성과 안전성을 증대시키기 위한 중요한 패러다임으로 자리 잡고 있다. 본 팀은 본격적인 환경 구현 전 언리얼 엔진 4를 기반으로 한 스마트 교통 통제 시스템의 설계 및 구현, GIS와의 통합을 통한 3D 도시 장면의 효율적 시각화, 그리고 가상 현실(VR)을 활용한 범죄 현장 재구성 및 예방 디자인의 선행연구를 분석하였다. 이러한 연구들은 도시 계획 및 범죄 예방 분야에서 언리얼 엔진의 활용 가능성을 제시하며, 스마트 시티 구축의 중요한 요소로 작용할 수 있음을 시사한다.

디지털 트윈 기술은 도시 정책과 관리의 효율성을 급격하게 증가시키고 있다. 선례로 Virtual Singapore는 싱가포르의 도시를 3D로 정밀하게 재현한 디지털 트윈 프로젝트이고 정부와 산업 간 협력을 통해 개발되었다, 주로 GIS 데이터와 3D 모델링 기술을 기반으로 구현되었으며 건물, 도로, 공원 등의 도시의 물리적 인프라뿐 아니라 인구 밀도, 교통 흐름, 에너지 사용 패턴 등의 다양한 데이터를 통합하여 도시의 디지털 복제물을 만들었다. 복제된 도시를 활용하여 정책 결정자들과 연구자들은 실제 도시 환경을 시뮬레이션하여 정책의 영향을 사전에 평가할 수 있는 테스트 베드로 도구를 활용한다. 아래는 3D 모델링 방식을 언리얼 엔진으로 특정하여 GIS 데이터를 기반으로 구현한 디지털 트윈을 구현하고 여러 시스템을 시뮬레이션하는 다양한 방식에 대한 연구들이다.

"Design and Implementation of an Unreal Engine 4-Based Smart Traffic Control System for Smart City Applications" 연구에서는 스마트 시티의 교통 문제를 해결하기 위한 시스템을 제안한다. 언리얼 엔진 4를 활용하여 실시간 데이터 수집 및 분석을 통해 교통 신호를 최적화하고, 혼잡도를 줄이는 방법을 모색하였다. 이 시스템은 차량 흐름과 보행자 안전을 동시에 고려하여, 도시 내 교통 효율성을 극대화할 수 있는 잠재력을 보여준다. 따라서, 언리얼 엔진의 강력한 시각화 기능은 교통 관리 시스템의 성능 향상에 기여할 수 있음을 시사한다.

"Efficient visualization of 3D city scenes by integrating the GIS and Unreal Engine" 연구에서는 GIS 데이터와 언리얼 엔진을 통합하여 3D 도시 장면을 효율적으로 시각화하는 방법을 탐구하였다. 이 접근법은 도시 계획자들이 도시 설계를 보다 직관적으로 검토하고, 시민들에게 도시의 변화된 모습을 효과적으로 전달하는 데 유용하다. 특히, 고해상도 텍스처와 실시간 렌더링 기능을

활용함으로써 도시 요소의 세부 사항을 생동감 있게 표현할 수 있으며, 이는 도시 관리 및 정책 결정 과정에서 중요한 역할을 할 수 있다.

마지막으로, "How useful are virtual reality simulations to the field of crime prevention through environmental design?" 연구는 가상 현실을 활용하여 범죄 예방 디자인의 효과를 분석하였다. VR 시뮬레이션은 범죄 현장을 재구성하여, 환경 디자인의 변화가 범죄 발생에 미치는 영향을 시각적으로 검토할 수 있는 강력한 도구로 작용한다. 연구 결과, VR을 통한 시뮬레이션은 범죄 예방 전략을 수립하는 데 실질적인 기여를 하며, 도시 계획자와 경찰이 공동으로 효과적인 예방 조치를 개발하는 데 도움을 줄 수 있음을 보여준다.

이와 같이, 언리얼 엔진을 활용한 연구들은 도시계획 및 범죄 예방 분야에서 효율적인 접근을 제공한다. 스마트 교통 통제 시스템, 3D 도시 시각화, VR 기반 범죄 예방 디자인 모두 효율성과 안전성을 제고하는 데 기여할 수 있으며, 향후 스마트 시티 구축에 있어 중요한 요소로 작용할 것으로 기대된다. 이러한 연구들은 언리얼 엔진의 다양한 응용 가능성의 기반이 될 것이며, 도시 계획 테스트베드의 형태로 자리매김 할 수 있다.

4. 프로젝트 과정

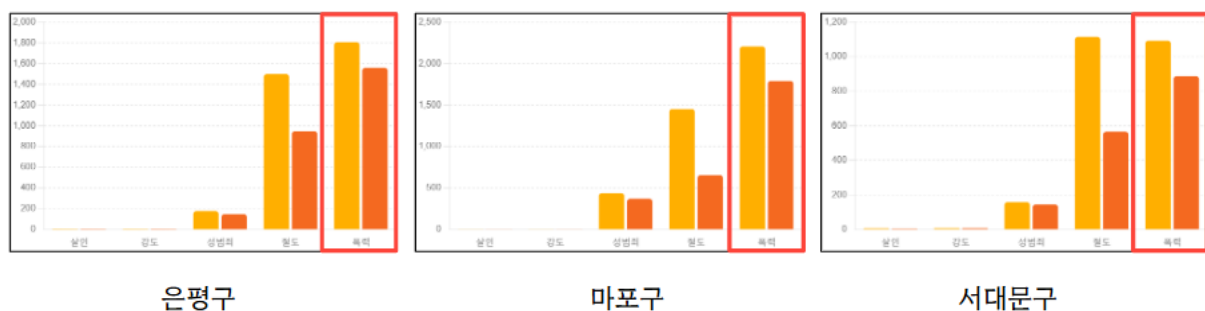
4.1. 가상 정책 수립

본 프로젝트의 가상 정책은 서울시 5대 범죄 발생률 통계 중 상위 1개 항목인 폭력 범죄의 검거율 증가를 목표로 한다.

본 팀이 수립한 가상 정책은 우버 택시 시스템에서 착안하였다. 시에서 시민 라이프가드를 선발하여 교육 후 공인 자격증을 발급한다. 지능형 CCTV에서 범죄 발생을 감지하였을 시 경찰에 신고가 접수됨과 동시에 인근 라이프가드에게 알림이 발송되며, 라이프가드는 경찰이 도착하기 전까지 초기 진압을 맡는다. 폭력 범죄는 그 특성상 시민들의 초기 진압이 검거에 큰 역할을 할 수 있기에 이와 같은 정책을 고안하게 되었다.

4.2. 디지털 트윈 부지 선정

은평구, 마포구, 서대문구 3구의 범죄 발생률 대비 검거율을 비교한 결과, 마포구에서 그 격차가 가장 큰 것으로 나타났다.



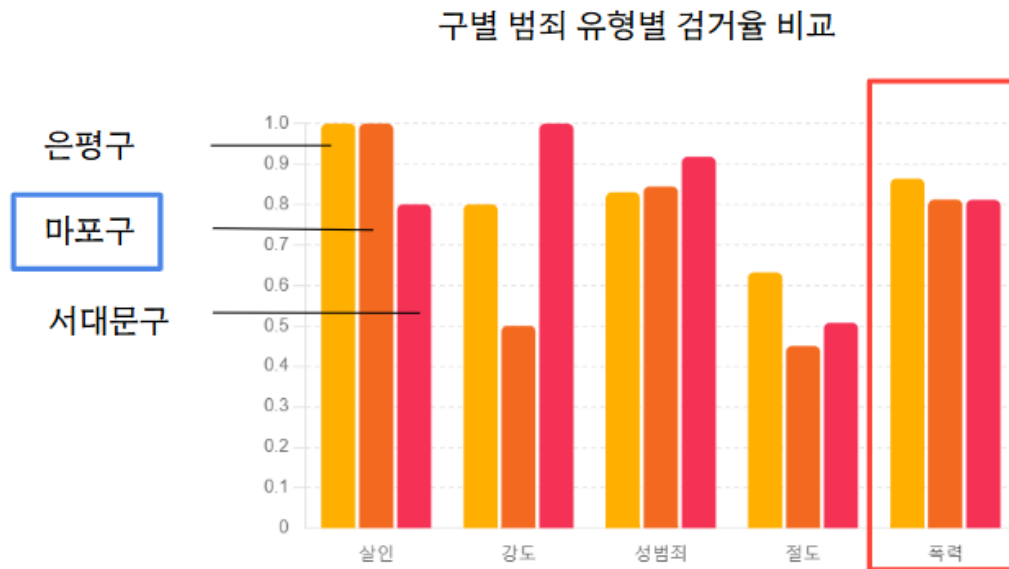


Figure 1. crime rate

마포구 연남동의 경의선숲길공원은 도시근린공원으로 낮 시간에는 유동 인구가 많은 반면 밤과 같은 특정 시간에는 유동 인구가 현저히 적어져 폭행 범죄가 발생할 가능성이 높을 것으로 판단 되었다. 따라서 경의선숲길공원을 디지털 트윈 구현 부지로 선정하였다.

4.3. 디지털 트윈 환경 구현



Figure 2. 마포구 연남동 경의선 숲길 landscape

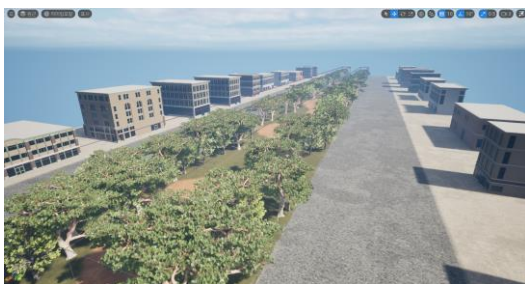


Figure 3, 4. make 마포구 연남동 경의선 숲길 in digital twin

4.4. 시뮬레이션 환경 구현

디지털 트윈 내 시뮬레이션 환경을 다음과 같이 설정하였다. 가상 정책을 우버 택시 시스템에
서 착안하였기에 우버 택시의 평균 대기 시간이 5분, 일반 택시의 평균 대기 시간이 8.4분인 것
을 고려하여 경찰과 라이프가드의 출동 시간을 세팅하였다.

(출처: 서울시 2020년 택시 서비스 시민만족도 조사)

시뮬레이션 1: 가상 정책 도입 전

길거리 행인: 무작위 위치로 이동을 반복

범죄자: 특정 위치로 이동해 폭력 범죄 (hp: 100)

경찰: 범죄 발생 10초 뒤 등장 (행인의 신고 시간 1.6초+출동 시간 8.4초) (-hp 10)

시뮬레이션 2: 가상 정책 도입 후

길거리 행인: 무작위 위치로 이동을 반복

라이프가드: 행인과 함께 무작위 위치로 이동을 반복하다가 CCTV에서 범죄가 감지되면 5초 뒤
범죄자 위치로 이동 -> 초기진압 (-hp 5)

범죄자: 특정 위치로 이동해 폭력 범죄 (hp: 100)

경찰: CCTV에서 범죄를 감지한 후 8.4초 뒤 등장 -> 진압 (-hp 10)

길거리 행인	라이프가드	범죄자	경찰
			
Figure 5. people	Figure 6. lifeguard	Figure 7. criminal	Figure 8. police

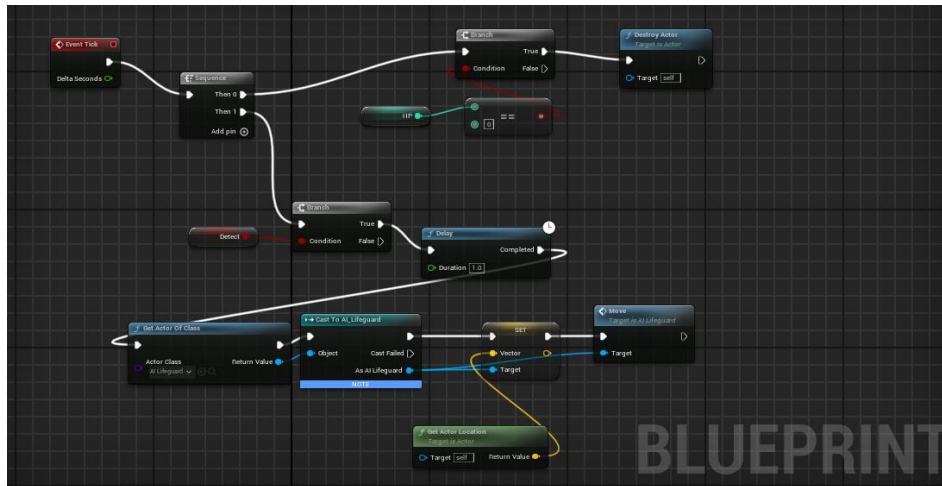


Figure 9. acting logic blueprint

4.5. YOLO 지능형 CCTV 구현

1) 데이터 수집

Mixamo는 코드와 아이디를 통해 무료로 3D 애니메이션을 다운로드 받을 수 있는 사이트이다. 애니메이션 제작자와 내려받는 자 모두 커스텀이 가능하고, 프레임 단위 당 애니메이션 데이터를 제공하는 사이트이기 때문에 개발을 위해 자주 활용된다. 객체 탐지에서 범죄와 처리 모두 특정이 필요한 행동이기 때문에 사이트에서 필요한 모션을 골라 각각 sitting, punching, lying idle, nursing 파일을 각각 fbx 형식, 스크린샷으로 다운로드 후 train-test, val 파일 분류하였다.

탐지해야 하는 객체의 위치를 표시하기 위해 Labeling를 활용하여 Bounding Box을 치고 좌표를 텍스트 파일로 추출하는 방식으로 라벨 이미지를 제작했다.



Figure 10, 11, 12. mixamo images and screenshot of the animations



Figure 13. file path of train_test split, and val

2) YOLO 정확성 검증

IOU(intersection over union)는 객체 탐지 모델이 예측한 bounding box가 정답과 겹치는 부분의 넓이를 계산하여 모델의 성능을 확인하는 방법이다. 이를 사용함으로써 x,y 좌표값이 분류모델의 경우처럼 정확히 일치하는 지를 보는 것이 아닌 예측한 bounding box가 정답과 최대한 가깝도록 학습시키는 것을 목적으로 한다. 관념적으로 0.5이상의 IoU면 유의미한 객체탐지를, 0.85이상의 IoU값을 가지면 정확도가 높은 모델로 인식한다.



Figure 14. concept of intersection over union

객체 탐지 모델에서 Precision와 Recall은 모델의 성능을 평가하는 중요한 지표이다. precision은 모델이 탐지한 객체 중 실제로 올바르게 탐지된 객체의 비율을 나타낸다 즉, 모델이 예측한 객체들 중에서 얼마나 정확하게 객체를 식별했는지를 나타내며, 높은 정밀도는 모델이 잘못된 객체를 적게 탐지했음을 의미한다.

반면에, recall은 실제 존재하는 객체들 중에서 모델이 올바르게 탐지한 객체의 비율을 나타냅니다. 즉, 이미지 내의 실제 객체들 중에서 얼마나 많은 객체를 탐지했는지를 나타내며, 높은 recall은 모델이 대부분의 객체를 놓치지 않고 탐지했음을 의미한다.

		Predicted	
		0	1
Actual	0	TN	FP
	1	FN	TP

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

실제 상황 (ground truth)	예측 결과 (predict result)	
	Positive	Negative
Positive	TP(true Positive) 옳은 검출	FN(false negative) 검출되어야 할 것이 검출되지 않음
Negative	FP(false positive) 틀린 검출	TN(true negative) 검출되지 말아야 할 것이 검출되지 않음

Figure 15. concept images of precision and recall

Average Precision은 객체 탐지 모델의 성능을 평가하기 위해 중요한 지표로, 모델의 Precision을 다양한 임계값에서 측정한 후 이들의 평균값을 계산하는 방법이다. 각 임계값에서 모델의 탐지 결과가 얼마나 정확한지를 나타내는 precision을 계산하고, 이를 다양한 임계값에 대해 통합하여 모델의 전반적인 성능을 평가한다. 이를 통해 모델의 성능을 측정할 수 있다. AP는 모델이 예측한 객체와 실제 객체 간의 일치도를 바탕으로 계산되며, 높은 AP 값은 모델이 정확하게 객체를 탐지했음을 의미한다.

mean Average Precision(mAP)는 여러 클래스 또는 데이터셋에 대해 평균적인 성능을 평가하는 지표이다. mAP는 개별 클래스에서 계산된 Average Precision 값들의 평균값을 취함으로써, 모델이 여러 객체 클래스에 대해 얼마나 고르게 우수한 성능을 발휘하는지를 평가한다. 이는 객체 탐지 모델이 다양한 클래스에서 동일한 수준의 성능을 유지하는지, 혹은 특정 클래스에서만 뛰어난 성능을 보이는지를 확인하는 데 유용하다. mAP는 모델이 전반적으로 얼마나 잘 학습되었는지를 나타내는 중요한 지표로, 객체 탐지 모델의 최종 평가에서 중요한 역할을 한다.

Weights & Biases (wandb)는 모델 학습 추적과 시각화를 위한 도구이다. wandb는 간편한 설정과 실험 추적 기능을 제공하며, 한 번의 로그인으로 YOLO모델을 실시간으로 실험을 추적하고 팀원들과 쉽게 공유할 수 있는 클라우드 기반 환경을 제공한다. 또한, 하이퍼파라미터, 성능 지표, 모델 체크포인트 등의 메타데이터를 자동으로 기록하고, 실험을 비교하거나 복원할 수 있는 고급 버전 관리 기능도 제공한다. 시각화에 주로 사용되는 Tensorboard는 로컬 서버에서 실행되며 텐서플로우와의 연동이 기본이지만, 설정이 상대적으로 복잡하고 협업 기능이 부족한 점이 단점이다. 시각화 측면에서도 wandb는 상호작용 가능한 대시보드와 직관적인 성능 비교와 클라우드 기반 서비스를 제공하여 팀(entity)단위로 활용할 때 효과적이다. wandb는 협업, 자동화된 실험 추적, 클라우드 기반 환경에서 실험을 공유하고 비교하는 데 유리하기 때문에 본 연구에 사용했다.

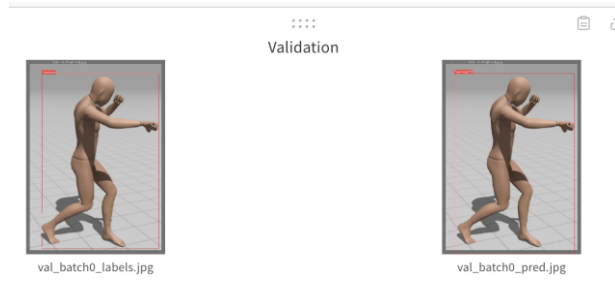


Figure 16. validation of trained YOLO (label and predicted images)

본 연구에서는 4-5) 단계에서 전처리된 데이터를 기반으로 YOLOv5 모델을 학습시켰으며, 그 결과 전체 데이터셋뿐만 아니라 특정 데이터셋을 추출하여 모델을 적용한 경우에도 높은 정확도를 보임을 확인하였다. 특히, 폭력 범죄의 행동 특성에 집중하여 모델을 구성한 결과, 해당 범죄 유형에 대한 탐지 성능이 개선되었음을 알 수 있었다.



Figure 17, 18. composition of label, image detection result by YOLOv5

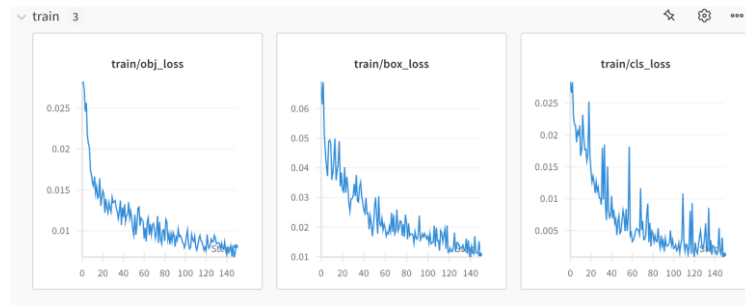


Figure 19. train/obj, train/box, train/cls loss of punching detection through YOLOv5 in wandb

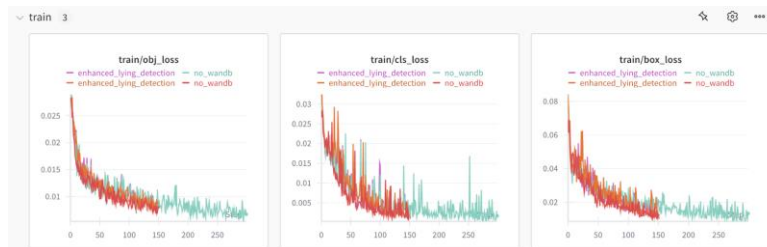


Figure 20. train/obj, train/box, train/cls loss of punching detection varied by epoch and rate of hyperparameter through YOLOv5 in wandb

epoch수치를 변경시키고, 각 하이퍼파라미터를 4번 조정 한 값의 손실도 비교 그래프는 위와 같다.

wandb: Using wandb-core as the SDK backend. Please refer to <https://wandb.me/wandb-core> for more i
wandb: Currently logged in as: inyang1112 (inyang1112-hana-academy-seoul). Use `wandb login --relog
train: weights=yolov5s.pt, cfg=, data=/content/drive/SharedDrives/지역 연계 프로젝트/datasets/data_cust
github: up to date with <https://github.com/ultralytics/yolov5> ✓
YOLOv5 🚀 v7.0-387-g4c904ac7 Python-3.10.12 torch-2.5.1+cu121 CUDA:0 (Tesla T4, 15102MiB)

hyperparameters: lr0=0.01, lrf=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=3.0, warmup

Figure 21. Login to wandb confirmation of the YOLO model

최종적으로, 본 연구에서 학습된 YOLO 모델의 성능을 평가한 결과, mean Average Precision (mAP) 수치는 0.8955에 도달하였다. mAP는 모델이 예측한 객체들의 정확도를 평가하는 지표로, 0과 1 사이의 값으로 표현되며, 값이 클수록 모델이 높은 정확도로 객체를 탐지했다는 의미를 갖는다. 0.8955라는 수치는 모델이 상당히 높은 수준의 성능을 보였음을 시사하며, YOLO 모델이 학습 과정에서 잘 작동하고 있음을 확인할 수 있다. 또한, 모델의 성능을 더 잘 이해하기 위해, mAP 수치 외에도 모델이 예측한 각 객체의 위치를 나타내는 x, y 좌표 및 height 값을 시각화한 그래프를 함께 제공한다. 이 그래프는 모델이 예측한 객체의 위치와 크기, 그리고 실제 객체 간의 일치도를 시각적으로 비교할 수 있게 해주며, 모델의 탐지 능력을 더 직관적으로 이해하는 데 중요한 역할을 한다. 이러한 시각화는 모델의 성능을 평가하는 데 유용하며, 특정 객체에 대한 정확한 탐지와 예측이 이루어졌음을 보여준다. mAP 수치와 함께 제공되는 시각화된 결과는 YOLO 모델의 실용성을 더욱 부각시키며, 향후 실제 환경에서의 적용 가능성을 평가하는 데 중요한 기준을 제공한다.

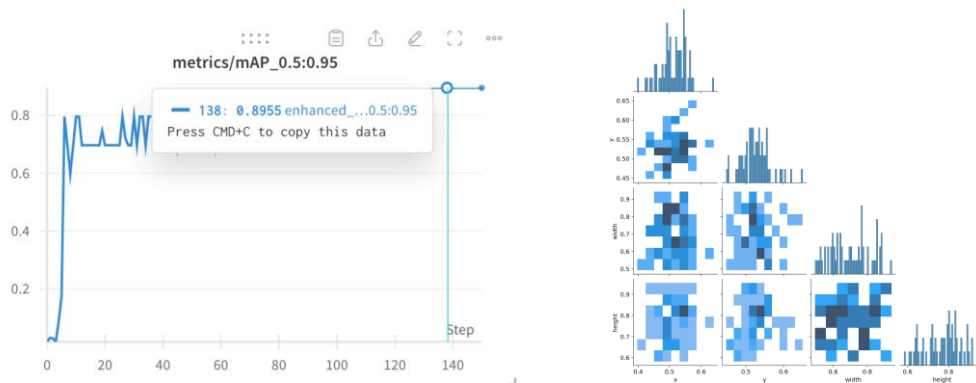


Figure 22. graph of metrics/mAP and visualization of image datas' informations

4.6. 엔리얼 엔진 내 Yolo 모델 적용

앞서 학습시킨 라이프 가드 및 행인의 행동을 인식하는 Yolo 객체 인식 모델을 엔리얼 엔진에 통합시키는 과정을 진행했다. 구체적으로 사용자의 게임 화면에 특정 행동을 하는 객체가 탐지된다면 Box 형태로 객체를 인식하며 분류된 행동과 분류의 정확도 또한 함께 표시하는 프로그램을 만들고자 한다. 파이썬 서버를 만든 뒤 TCP 소켓 기능을 활용하여 엔진과 연결한다. 연결 후 게임 창 화면을 스크린샷 떠 이를 Yolo 모델에 전달하고 그 결과값을 엔진으로

재전송하는 기능을 구현하여 엔진과 모델 사이에서 정보를 주고받을 수 있도록 한다.

1) 엔진과 파이썬 서버 연결

앞서 학습시킨 라이프 가드 및 행인의 행동을 인식하는 Yolo 객체 인식 모델을 언리얼 엔진에 통합시키는 과정을 진행했다. 구체적으로 사용자의 게임 화면에 특정 행동을 하는 객체가 탐지된다면 Box 형태로 객체를 인식하며 분류된 행동과 분류의 정확도 또한 함께 표시하는 프로그램을 만들고자 한다. 파이썬 서버를 만든 뒤 TCP 소켓 기능을 활용하여 엔진과 연결한다. 연결 후 게임 창 화면을 스크린샷 떠 이를 Yolo 모델에 전달하고 그 결과값을 엔진으로 재전송하는 기능을 구현하여 엔진과 모델 사이에서 정보를 주고받을 수 있도록 한다.

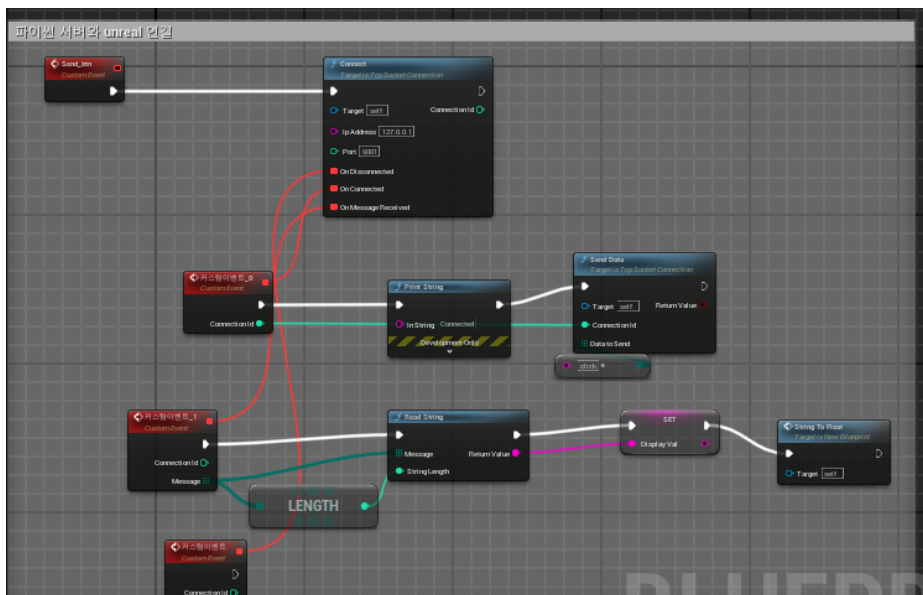


Figure 23. Blueprint connecting python server and unreal engine

서버 측에서는 Visual Studio Code(VSCode)를 사용하여 Python 파일을 작성하고, Python의 socket 모듈을 활용해 서버를 구현했다. 서버는 엔진에서 다양한 입력값을 받을 수 있도록 설계하였으며, 각 입력에 따라 수행할 동작을 코드로 작성했다. 초기 테스트 단계에서는 연결 기능 검증은 목표로 하기에 임의로 간단한 메시지를 출력하는 방식으로 진행했다.

또한, 새로운 위젯 블루프린트 클래스를 정의하여 사용자로부터 입력을 받을 버튼을 추가했고, 버튼 클릭 시 입력값을 변수로 저장하는 블루프린트 코드를 작성했다. 이후, Python 서버와 언리얼 엔진 간 연결을 테스트하였으며, 연결 성공 여부를 확인했다. 이러한 과정을 통해 양방향 통신의 기초적인 기능 구현과 테스트를 완료했다.

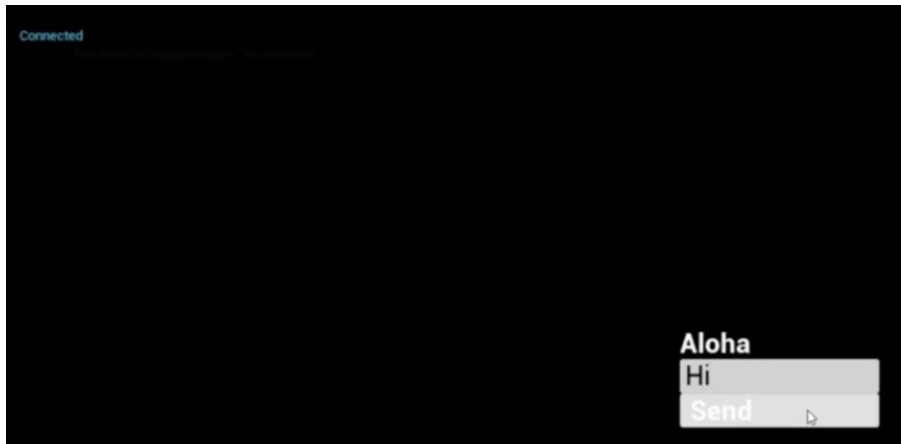


Figure 24. server connection validation using python tcp socket

1) YOLO와 서버, 엔진 연결

앞서 학습시킨 모델 중 가장 좋은 성능의 가중치와 편향을 저장한 파일인 best.pt를 불러와 서버 파이썬 파일 내에 불러왔다. 서버 파일에는 tcp 소켓과 통신하기 위한 socket, Yolo를 불러오기 위한 ultralytics, 스크린 샷 및 카메라 관련 기능을 위한 cv2와 Pyautogui 등을 불러왔고, 엔진 블루 프린트 내에 정의했던 IP, 포트 주소와 일치하는 서버를 하나 생성했다. 그리고 엔진과 연결된 이후부터 지속적으로 프레임을 캡처하고 배열로 내보내 정의한 모델에 인식시키고 인식된 결과를 다시 엔진의 변수에 넣어주는 코드를 작성했다. 이로써 파이썬 서버 쪽의 정보 송수신이 완료되었다.

엔리얼 엔진 내에서 특정 신호를 수신하면 화면에 바운딩 박스를 표시하는 기능을 구현하기 위해 Bounding_box라는 새로운 블루프린트 클래스를 정의했다. 해당 클래스에서는 바운딩 박스의 좌표를 기반으로 화면에 적절히 표시될 수 있도록 좌표 계산을 수행하는 Coord_Cal 함수를 새로 작성했다. 이 함수는 바운딩 박스 좌표가 Python 서버로부터 엔리얼 엔진으로 전달될 때, 해당 좌표를 엔진 내 화면의 해상도와 크기에 맞추어 조정하는 역할을 한다. 이를 위해 화면 크기와 바운딩 박스 크기를 비교한 후, 다양한 연산자를 활용하여 포지션을 변환하고 조정하는 과정을 거쳤다.


```

import sys
import socket
from ultralytics import YOLO
import numpy as np
import cv2
import pyautogui
from PIL import Image
from asyncio.windows_events import NULL

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind(('127.0.0.1', 6001))
model = YOLO('best.pt')

server.listen()

while True:
    client, address = server.accept()
    print('connected')
    val = client.recv(1024).decode('utf-8')
    print(val)
    if val == 'click':
        while True:
            frame = pyautogui.screenshot()
            frame = np.array(frame)

            frame = cv2.cvtColor(frame,
                                cv2.COLOR_BGR2RGB)(model(frame)

            for r in val:
                boxes = r_boxes.cpu().numpy()
                coord = boxes.xywhn
                conf = boxes.conf
                cls = boxes.cls

                str_arr = ""
                for i in coord:
                    for j in i:
                        str_arr += str(j) + " "
                print(str_arr)

                client.send(str_arr.encode('utf-8'))
    if val == 'close':
        client.close()
        sys.exit()
    elif val == 'Hi':
        client.send('Hello'.encode('utf-8'))
    else:
        client.send('Aloha'.encode('utf-8'))
        client.close()

```

Figure 25. python server final code

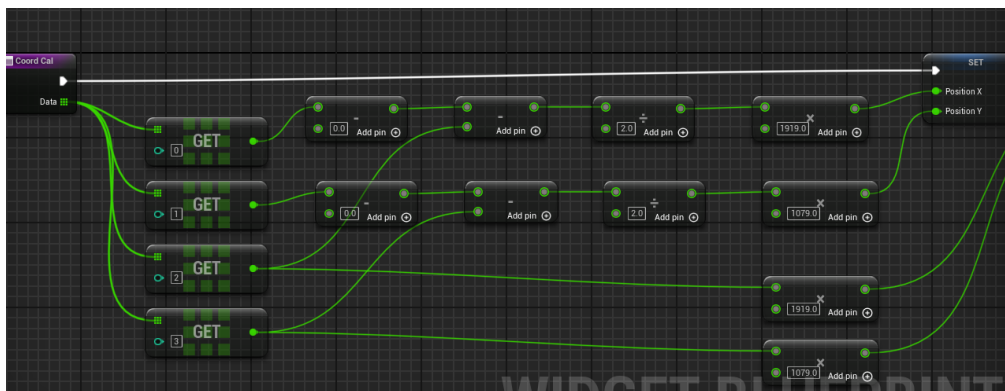


Figure 26. Coord Cal (calculating bounding box area)

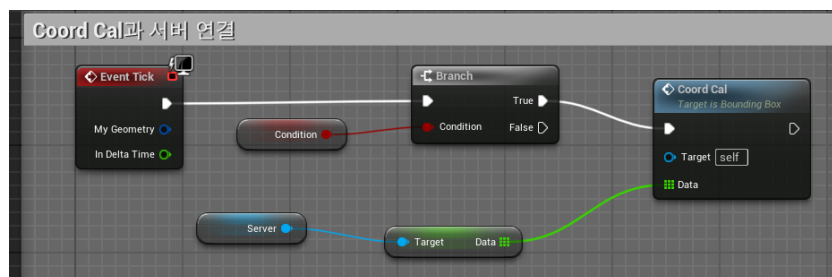
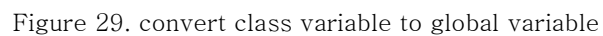
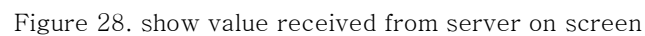


Figure 27. deliver value received from server to coord cal function

또한, 시각적으로 바운딩 박스를 표시하기 위해 SlateBrushAsset을 활용하여 박스 형태를 생성했다. 이를 통해 화면 상에서 바운딩 박스가 명확히 시각화될 수 있도록 설정했다. 생성된 박스 형태는 Bounding_box 블루프린트에 연결되었으며, 데이터 수신 시 바운딩 박스를 동적으로

추가적으로, 바운딩 박스 좌표 외에도 모델에서 예측된 클래스 명칭과 해당 예측의 정확도 값을 함께 표시하는 기능을 구현했다. 이를 통해 사용자에게 바운딩 박스의 의미를 명확히 전달할 수 있도록 했다. 모델에서 전달받은 클래스 명과 정확도 값은 바운딩 박스와 연계되어 화면에 실시간으로 표시되며, 사용자는 이를 통해 탐지된 객체의 정보를 직관적으로 확인할 수 있다.



5.1. 연구 결과

- ### 1) 시뮬레이션 1: 정책 도입 전



Figure 30. Step 1: people walking around park



Figure 31. Step 2: violation crime occurs



Figure 32. Step 3: police come 10 seconds after crime occurs and suppress (-hp 10)



Figure 33. Step 4: suppression finish

총 진압 시간: 12.37초

2) 시뮬레이션 2: 정책 도입 후



Figure 34. Step 1: people, lifeguard walking around park

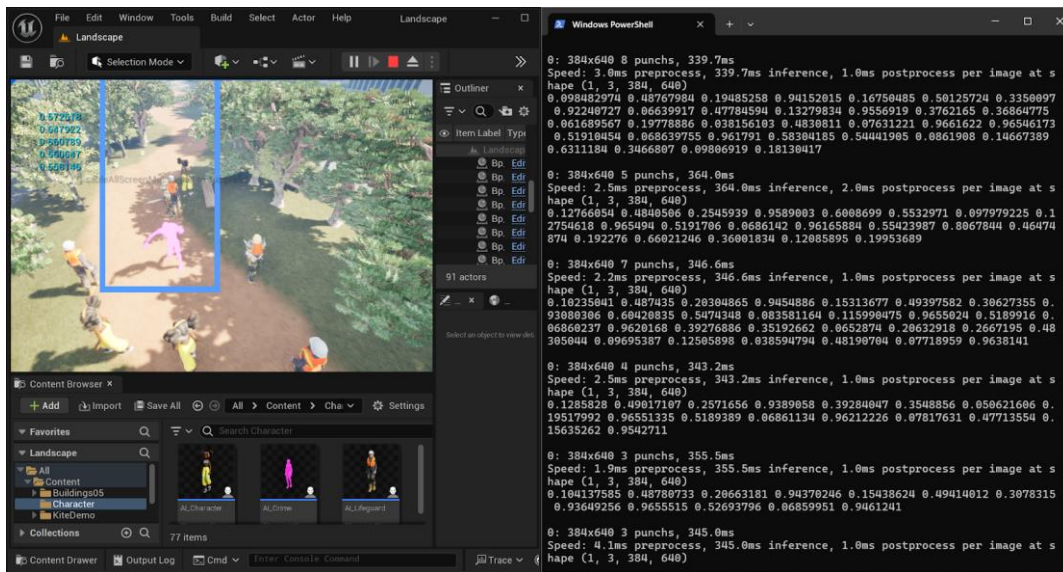


Figure 35. Step 2: crime occurs → enhanced AI CCTV detect it



Figure 36. Step 3: lifeguard approaches to criminal 5 seconds after detection → suppress (-hp 5)



Figure 37. Step 4: police come to criminal 8.4 seconds after detection → suppress (-hp 10)

총 진압 시간: 10.34초

5.2. 결론 및 의의

본 프로젝트는 언리얼 엔진 내에서 Yolo모델을 활용한 범죄 탐지 시뮬레이션을 시도하며 특정 도시를 디지털 트윈하여 가상의 실험적인 정책의 테스트 베드로 활용했다. 본 프로젝트의 차별성은 현재까지 폭넓은 연구가 이루어지지 않은 실시간 범죄 데이터 기반의 객체 탐지 시스템과 함께, 디지털 트윈을 통해 새로운 정책을 실제 환경에 적용하여 효용성을 검증할 수 있는 새로운 구조의 시뮬레이션을 구현했다는 점이다.

선례 연구들은 주로 디지털 트윈 환경을 실제와 최대한 유사하게 만들어 정확하게 현재 존재하는 정책의 개선책 시뮬레이션을 돌리는데 치중했다면 본 연구는 실험적이고 혁신적인 정책 및 시스템들에 필요한 다양한 신기술을 손쉽게 추가하고 개발할 수 있으며 새로운 패러다임 및 시스템을 직접 구성해서 테스트해 볼 수 있는 툴을 만들었다는 점에서 큰 의의를 가진다. 실제 연구에서 제안되어 테스트 해봤던 라이프가드 시스템 또한 지능형 CCTV와 우버 시스템을 접목하여 자격을 갖춘 일반인의 범죄 예방 시스템이라는 다소 파격적인 정책이다. 따라서 이런 실험적인 정책을 Unreal Engine으로 개발된 테스트 베드에서 실험함으로써 정교한 시각화 능력과 다양한 액터 간의 상호 작용을 가능하게 하는 게임 엔진으로써의 속성, 개발한 신기술을 손쉽게 추가 가능한 특성을 활용할 수 있게 한다. 따라서 만든 도시의 외부 모습 및 내부의 다양한 요소들의 상호 작용까지 시뮬레이션할 수 있게 하는 환경을 제공한다.

이러한 접근이 심화된다면 사회 전반을 개혁할 수 있게 만드는 혁신적 정책들의 실현 가능성에 대한 확실한 근거가 될 것이며 정책 관리자 및 시민들의 모험심과 도전 정신을 자극하여 전보다 대담한 정책의 실현을 촉구할 것이다.

참고문헌

- Subree, M. I. H., Hasan, M. R., & Sayma, M. H. (2022). Design and Implementation of an Unreal Engine 4-Based Smart Traffic Control System for Smart City Applications. *International Journal of Advanced Computer Science and Applications*, 13(12), 100-107.
- Xu, H., He, B., Li, Z., Lin, H., & Tang, A. (2023). Efficient visualization of 3D city scenes by integrating the GIS and Unreal Engine. *Proceedings of the Fourth International Conference on Geoscience and Remote Sensing Mapping (GRSM 2022)*, 12551, 1255101.
- Zhang, Y., & Wang, L. (2020). Crime Scene Reconstruction Based on Virtual Reality. *Journal of Forensic Sciences*, 65(3), 870-878.
- Smith, J. A., & Brown, T. L. (2019). How useful are virtual reality simulations to the field of crime prevention through environmental design? *Security Journal*, 32(4), 456-

472.

- 서울시 열린데이터광장. (2024). 서울시 5대 범죄 발생현황 통계.
<https://data.seoul.go.kr/dataList/OA-1234/S/1/datasetView.do>
- London Metropolitan Police. (2024). Crime Data Dashboard.
<https://www.met.police.uk/sd/stats-and-data/>
- San Francisco Police Department. (2024). Crime Dashboard.
<https://www.sanfranciscopolice.org/stay-safe/crime-data/crime-dashboard>