

증강현실 게임에서 딥러닝을 활용한 배경객체 분석에 관한 연구

김한호^{1*}, 이동열²

¹공주대학교 게임디자인학과 학생, ²공주대학교 게임디자인학과 교수

A Study on the Analysis of Background Object Using Deep Learning in Augmented Reality Game

Han-Ho Kim^{1*}, Dong-Lyeor Lee²

¹Student, Dept. of Game Design, Kongju University

²Professor, Dept. of Game Design, Kongju University

요약 증강현실기술을 사용하는 증강현실 게임이 늘어남에 따라 사용자들의 요구도 많아지고 있다. 증강현실 게임에서 사용되는 게임 기술에는 MARKER, MARKERLESS, GPS등을 활용한 게임이 주를 이루고 있다. 이러한 기술을 활용한 게임은 배경과 다른 오브젝트를 증강할 수가 있다. 이 문제를 해결하기 위해 증강현실의 중요한 요소인 배경에서 객체를 분석하여 증강현실 게임을 개발하는데 도움을 주고자 한다. 증강현실 게임에서 배경을 분석하기 위해 UNITY엔진에서 TensorFlow Lite를 활용하여 딥러닝 모델을 적용하여 배경 객체를 분석하였다. 이 결과를 활용하여 배경에서 분석된 객체의 종류에 맞춰 게임에 증강되는 오브젝트를 배치 할 수 있다는 결과를 얻었다. 이 연구를 활용하여 배경에 맞는 오브젝트를 증강하여 향상된 증강현실 게임을 개발 할 수 있을 것이다.

주제어 : 증강현실, 딥러닝, Tensorflow Lite, 증강현실배경, 증강현실게임, 객체탐지

Abstract As the number of augmented reality games using augmented reality technology increases, the demands of users are also increasing. Game technologies used in augmented reality games are mainly games using MARKER, MARKERLESS, GPS, etc. Games using this technology can augment the background and other objects. To solve this problem, we want to help develop augmented reality games by analyzing objects in the background, which is an important element of augmented reality. To analyze the background in the augmented reality game, the background object was analyzed by applying a deep learning model using TensorFlow Lite in the UNITY engine. Using this result, we obtained the result that augmented objects can be placed in the game according to the types of objects analyzed in the background. By utilizing this research, it will be possible to develop advanced augmented reality games by augmenting objects that fit the background.

Key Words : AR, DeepLearning, Tensorflow Lite, AR background, AR game, Object detection

1. 서론

1.1 연구배경

스마트폰의 다양한 콘텐츠기술이 발달함에 따라 여러 콘텐츠 기술중 증강현실 기술을 활용하는 콘텐츠가

늘어나고 있다. 2016년 7월 8일에 출시된 포켓몬고의 성공으로 증강현실 기술을 기반으로 하는 콘텐츠들이 늘어나고 있다. 예술, 패션, 광고 등 다양한 분야에서 이를 활용한 콘텐츠들이 증가하고 있으며 게임분야도 증강현실 기술을 적용하는 사례들이 많이 나오고 있다.

*This article is extended and excerpted from the conference paper presented at ICSMB2021

*Corresponding Author : Han-Ho Kim(hanhokim@gmail.com)

Received October 7, 2021

Accepted November 20, 2021

Revised October 25, 2021

Published November 28, 2021

증강현실 콘텐츠를 사용하는 유저들이 늘어남에 따라 증강현실 콘텐츠의 문제점들도 많아지고 증강현실 게임에 대한 유저들의 요구도 늘어나고 있다.

1.2 문제제기

포켓몬고의 성공 이후 증강현실 기술을 사용한 게임 콘텐츠들이 등장 하고 있으며 Fig.1을 살펴보면 증강현실을 활용하는 게임 콘텐츠의 증가가 예상되어진다. 증강현실을 활용한 게임 콘텐츠 게임에서 이용되는 증강현실 기술은 MARKER, MARKERLESS, GPS(Global Positioning System) 등을 사용하는 기술이 사용되어지고 있다. 이러한 증강현실 기술들은 증강현실 게임을 개발함에 있어서 사용자에게 일관된 오브젝트 정보를 전달하는데 한계가 있다. 포켓몬고는 GPS나 지도중심의 위치기반기술을 이용한 게임이다 보니 위치데이터를 통하여 플레이되는 장소가 NPC(Non-Player Character) 캐릭터의 특성과 플레이 장소에서의 환경간에 이질감이 생길 수 있다. 예를 들이 산이 보이는 배경에서 게임이 플레이 되는데 위치 데이터는 강이 될 경우 사용자는 증강현실 게임의 배경과 NPC사이에 이질감을 느낄 수 있다. 포켓몬고의 게임에서 Fake GPS 등의 기술을 사용하여 게임을 진행하는 문제점등도 해결할 필요성이 있다. 증강현실 콘텐츠에서는 원격현장과 현장의 장소 인지도가 차이가 생길 수 있고 이러한 증강정보가 사용자에게 일관된 경험을 유지하기 위한 기술 개발의 필요성이 있다[1].

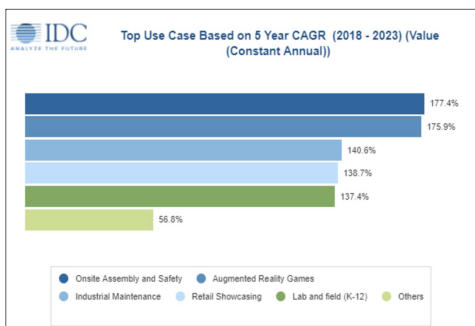


Fig. 1. IDC Top Use Case Nased on 5 Year CAGR(2018-2023)

2. 관련연구

2.1 증강현실게임

증강현실 관련 기술연구가 본격적으로 시작된 것은

1990년대 초 보잉사가 'Augmented Reality'라는 신조어를 등장 시키면서 이다. 전기공의 케이블 조립작업을 위해 실제 환경에 가상의 그래픽을 혼합 할 수 있는 시스템을 구축하였고 이는 증강현실의 개념이 되었다[2]. 1995년 로널드아즈마(RonaldAzuma)가 정의한 바에 의하면, 실세계와 가상세계를 실시간으로 동시에 사용자에게 보여줌으로써, 사용자에게 보다 향상된 몰입감과 현실감을 제공하는 기술이라고 이야기 한다. 로널드아즈마는 증강현실은 Fig.2와 같이 실제 환경 속에서 현실의 환경과 가상의 물체가 혼합될 수 있고, 이러한 혼합은 실시간으로 상호작용할 수 있으며, 3차원으로 정합될 수 있는 특징을 갖는다고 한다[3]. 증강현실을 이용한 게임으로는 포켓몬고가 대표적이다. 포켓몬고는 위치 기반의 실시간 증강현실을 모바일로 구현한 게임이다. 게임이 인기가 높아짐에 따라 GPS의 위치를 속이는 다양한 방법들로 게임플레이를 악용하는 사례들이 나오기도 하였다[4]. 포켓몬고의 등장 후 '우리 동네 히어로'와 같은 증강현실 게임도 등장하였으며, 2020년에 '마인트래프'게임을 증강현실 게임으로 제작한 '마인트래프 EARTH'가 개발 되었다. 이러한 게임들의 사례를 보더라도 게임에서 증강현실의 배경은 단지 증강된 오브젝트를 보여주기 위한 배경의 역할만 있고 배경을 활용한 콘텐츠는 없다. 포켓몬고에서도 숲에서 나와야할 NPC가 강에서 나오는 등 실제 배경 환경과 다른 NPC의 등장에 사용자의 몰입감이 저하될 수 있다. 최근 딥러닝이나 머신러닝을 이용한 게임이 많아지고 있어 이를 활용한 배경 객체 분석에 대한 연구도 필요성이 있다.

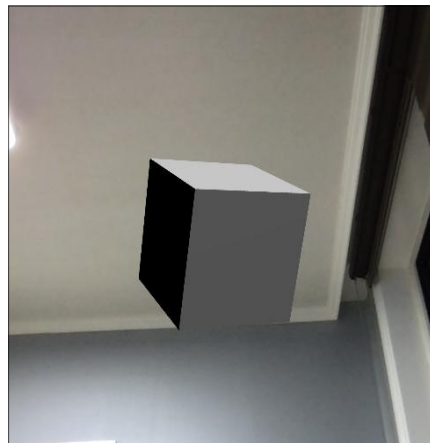


Fig. 2. Augmented reality example

2.1 SSD(Single Shot Detector)

SSD는 객체 검출 속도 및 정확도 사이에 균형을 가지고 있는 알고리즘이다[5]. Fig. 3.을 보면 Object Detect 알고리즘 중 Yolo의 경우는 속도에 장점을 가지고 있고 Faster RCNN은 정확도에 장점을 가지고 있다[6,7].

Yolo는 7x7 grid 하나만을 사용하지만 Fig. 4.을 보면 SSD는 이미지를 38x38, 19x19, 10x10, 5x5, 3x3, 1x1의 그리드로 나누어서 결과를 나타낸다. SSD는 한번만 입력된 이미지에 대해 CNN을 실행하고 형상맵을 계산한다. 경계 상자 및 객체 분류 확률을 예측하기 위해 이 형상맵을 3 x 3 크기로 CNN을 수행한다. SSD는 CNN처리 후 경계 상자를 예측하고 이 방법으로 다양한 스케일의 물체를 검출 할 수 있다. 하지만 SSD는 Yolo에 비하여 사용이 어렵다는 단점을 가지고 있다[8]. 이러한 연구 결과를 바탕으로 SSD알고리즘을 사용하기로 하였다.

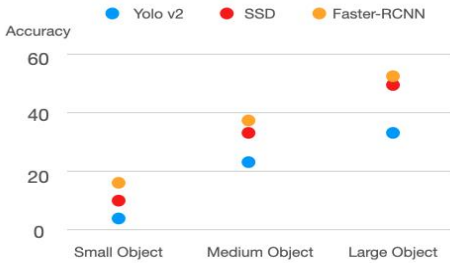


Fig. 3. CV-Tricks.com, Zero to Here: Guide to Object Detection using Deep Learning: Faster R-CNN, YOLO, SSD

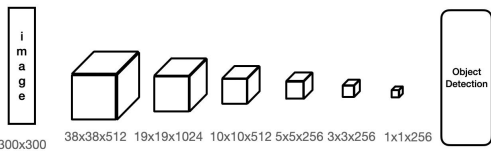


Fig. 4. SSD MultiBox Dector for real-time processing

2.2 Tensorflow Lite

TensorFlow Lite는 Android, Ios, 리눅스 기반의 IoT를 포함하는 플랫폼에서 TensorFlow 모델을 사용하기 위한 공식 프레임워크이다.

모바일 및 임베디드 장치에서는 ML을 통합하는 경향이 있기 때문에 모바일 애플리케이션에서 사용자가

자신의 카메라 및 음성 등을 사용 하기 위해서는 TensorFlow를 최적화시켜야 한다. TensorFlow Lite는 모델선택, 모델변환, 기기배포, 모델 최적화의 단계로 개발이 된다[9].

TensorFlow Lite Converter는 학습된 모델을 TensorFlow LiteFlatBuffer(tflite)의 형태로 변형하여, 디바이스 내부에서 경량화된 모델을 사용할 수 있도록 한다. 모델 최적화 도구를 사용하여 모델의 크기를 줄이고 모델의 성능을 향상시키면서 정확도도 유지할 수가 있도록 한다. 두 번째로 TensorFlow Lite interpreter는 디바이스에서 최적화된 모델을 구동할 수 있도록 도와주며, 다양한 언어(대표적으로 C++ 또는 JAVA)와 호환할 수 있게 한다[10]. TensorFlow Lite는 Unity 엔진용 라이브러리를 제공하여 본 연구에서 사용하기로 하였다.

3. 요구분석 및 설계

3.1 요구분석

선행연구에서 증강현실 게임에서 몰입감을 주는 요인으로 배경에 어울리는 가상의 이미지가 실시간으로 혼합되어 작용한다는 것을 알 수 있다. 증강현실 게임인 포켓몬고의 사례를 살펴보면 GPS 데이터를 기반으로 게임을 하였을 때 생기는 문제점들도 확인하였다. 그렇기 때문에 증강현실 게임에서 딥러닝 모델을 활용한 배경이미지에서 객체를 분석하여 몰입도를 높이고 상황에 맞는 오브젝트의 제공의 필요하다. 이에 증강현실게임에 배경이 사용되어지는데 필요한 요구사항은 다음과 같이 정리될 수 있다.

1) 실재감

게임의 사용자는 카메라 배경에 증강된 물체를 통하여 몰입감을 느끼게 된다. 배경의 이미지와 증강되는 오브젝트의 일체감을 통하여 실제로 오브젝트 존재함을 느끼게 된다.

2) 최적화

증강현실 게임에서 카메라로 들어온 배경 이미지를 실시간으로 분석한다. 이때 게임의 프레임이 저하되는 문제점 등이 생긴다면 사용자들이 게임을 플레이하는데 방해요소로 작용한다. 게임플레이에 최적화된 객체

분석 기법을 활용하여야 한다. ObjectDetect시 정확도 및 속도 사이에서 게임에 사용하기 적합한 알고리즘이 필요로 한다.

3.2 설계

본 논문에서는 선행연구를 통하여 SSD 모델을 사용하여 콘텐츠에 이미지 배경에서 객체를 분석하기 위하여 Unity 엔진에서 TensorFlow Lite를 사용한다. Unity 엔진에서 배경객체를 분석하기 위해 Fig. 5와 같은 순서로 설계를 하였다.

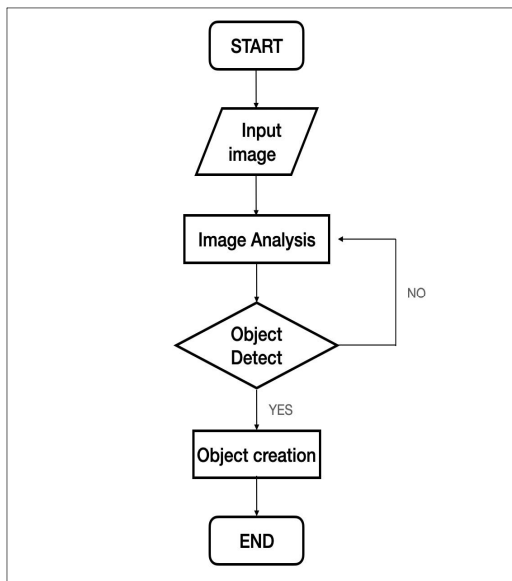


Fig. 5. System Flowchart

- 1) Start : 썬의 시작
- 2) 오브젝트 생성 명령 : 배경에 적합한 오브젝트를 생성하는 명령
- 3) 카메라 이미지 입력 및 분석 : 카메라 이미지를 입력받아 TensorFlow Lite를 통해 분석
- 4) ObjectDetect : 배경에서 객체 판별 및 결과 획득
- 5) 오브젝트 생성 : ObjectDetect가 성공하였을 시 선택된 객체에 맞는 NPC 오브젝트 생성

본 논문에서는 배경에서 객체를 분석하기 위해 SSD로 학습된 모델을 필요로 한다. 연구에서 검증을 위해 TensorFlow에서 제공하는 tflite 모델인 COCO_SSD_MobileNet_v1을 사용한다. COCO_SSD_MobileNet_v1은 COCO이미지 데이터를 학습시킨

모델로 사람, 자동차, 배, 피자 등 약 91종의 Class를 학습시킨 모델이다. 위 모델을 활용하여 배경에서 오브젝트를 3개의 범주로 분류한다. 1번째 범주는 사람, 2번째 범주는 자동차, 모터사이클, 버스, 트럭, 3번째 범주는 바나나, 사과, 오렌지, 당근등으로 분류하여 각 범주별 오브젝트가 배경에 있을 경우 범주에 해당하는 오브젝트가 등장 되었는지 확인한다.

4. 구현 및 검증

4.1 Tensorflow Lite Manager

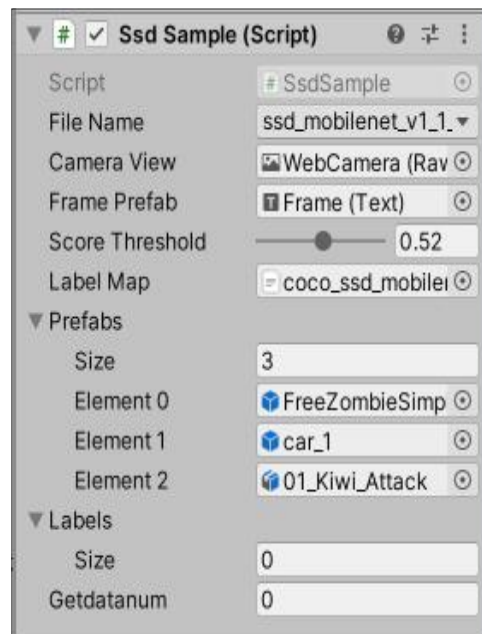


Fig. 6. Unity TensorflowLite Setting

TensorFlowLite 프로젝트를 Unity엔진에서 사용할 수 있게 porting한 tflite-unitysample을 다운받고 import한다. SsdSample.cs 파일에 선언된 변수를 Fig. 6과 같은 형태로 FileName, Camera, Label Map등의 값을 입력한다. SSD학습데이터는 COCO_SSD_MobileNet_v1을 모델로 사용한다. Label Map 변수에는 SSD모델의 라벨을 txt파일로 입력한다. 위 모델은 91종의 Class를 제공한다. ScoreThreshold는 Class 감지 시 점수를 의미하는 것으로 신뢰도 점수를 의미한다. 객체 검출의 정확도를 위해 점수가 0.5 이상의 객체만 검출하도록 한다. 실제 게임에서 사용하기 위해서 Canvas에 Webcam을 렌더링하기위한 Webcam Object

를 설치하고 RawImage를 스크린으로 활용한다. Canvas에 3D NPC 오브젝트가 증강 되어야 함으로 Canvas의 속성 중 RenderMode를 Screen Space-Camera로 변경하고 Camea를 Maincamera로 연결한다.

4.2 객체검출 및 NPC 오브젝트 생성

TensorFlow Lite에서 검출된 classid를 변수로 입력받는다. SSD모델에서 게임 내에서 검증을 하기 위해 다음과 같이 클래스 아이디를 Table 1처럼 분류하여 객체를 검출하는 조건을 생성한다. 객체를 검출할 수 있는 버튼을 제작 후 게임을 플레이시켜 버튼을 누르게 되면 Fig. 7의 setFrame함수를 통하여 검출된 오브젝트가 있을시 객체에 맞는 NPC 오브젝트를 생성되게 하였다. 최종적으로 조건에 맞는 오브젝트를 검출하였을 경우 Table 1의 Ar Object와 같은 결과를 얻게 된다. Table 2와 같은 테스트를 통해 각 분류 모델별 테스트를 실시하였고 카메라로 획득한 배경 오브젝트를 분석하여 다음과 같이 증강현실 오브젝트가 나오는 것을 확인 하였다. Fig. 8은 class id : 0 person을 검출하였을 경우 실제 화면에 좀비 NPC가 나오는 결과 화면이고 Fig9는 class id : 2 car를 검출하였을 때 화면, Fig 10은 class id : 51 banana를 검출한 결과 화면이다.

```
void setFrame(Text frame, SSD.Result result, Vector2 size)
{
    if (result.score < scoreThreshold)
    {
        frame.gameObject.SetActive(false);
        return;
    }
    else
    {
        frame.gameObject.SetActive(true);
    }

    getdatanum[result.classID] = 1;

    frame.text = $"({GetLabelName(result.classID)} : {(int)(result.score * 100)}%)";
    var rt = frame.transform as RectTransform;
    rt.anchoredPosition = result.rect.position * size - size * 0.5f;
    rt.sizeDelta = result.rect.size * size;
}
```

Fig. 7. setFrame Method

Table 1. Augmented object table by class ID

class id	name	Ar Objet
0	person	Zombie
2,3,5,7	car,motorcycle,bus,truck	Carzombie
51,52,54,56	banana,apple,orange,carrot	Foodzombie

Table 2. Test result table

class id	test count	success count
0	20	18
2,3,5,7	20	17
51,52,54,56	20	18

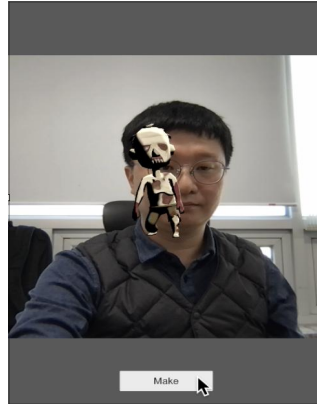


Fig. 8. Human detection result screen

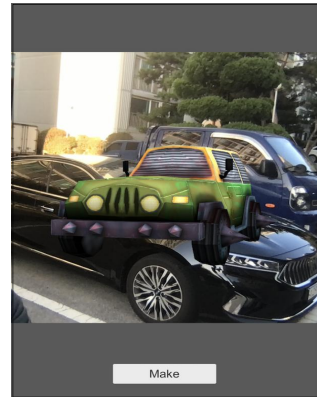


Fig. 9. Car detection result screen

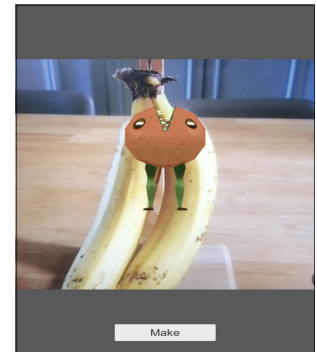


Fig. 10. Banana detection result screen

5. 결론

증강현실에서 배경을 활용한 게임개발을 위해 본 연구에서 딥러닝기술을 활용한 ObjectDetect를 추출하였다. 이를 활용하여 카메라를 통해 입력된 배경 이미지에서 증강현실 게임에서 증강되는 오브젝트와 연관이 있는 NPC를 분류하였고, 필요한 객체를 분석하여 분석된 객체에 맞는 NPC를 증강하였다. 제작 사례를 통하여 배경을 분석하여 얻은 오브젝트 결과를 통해 가상의 NPC 오브젝트를 증강하였다. 연구결과 몇가지 문제도 발견하였다. 모바일에서의 프레임 저하가 있었고, 배경오브젝트가 다른 오브젝트와 겹쳐있을 경우 인식이 저하되는 것도 발견되어 향후 연구를 통한 해결이 필요하겠다. 연구결과 기존 포켓몬고등의 게임에서 나타날 수 있는 배경과의 이질감이 생기는 NPC증강이나 Fake GPS등을 활용하여 게임을 실행하는 문제점등을 해결할 것으로 기대한다. 이를 활용하면 다양한 형태의 증강현실 게임 및 콘텐츠들이 개발 될 수 있다. 향후 지속적인 연구를 통하여 게임 배경에서 객체 분석에 사용할 최적화된 딥러닝 모델에 관하여 연구를 하고 배경에 맞는 캐릭터를 자동으로 분류할 수 있는 시스템제작을 위한 초석이 되길 바라고 이를 활용하여 다양한 증강현실 게임개발에 도움이 되길 바란다.

REFERENCES

- [1] J. E. Hwang, S. H. Lee, H. E. Kim, J. Seo & J. Min. (2012). *A Preliminary Study on an User Created Content Framework for Mobile Augmented Reality*. *Proceeding of 2012 Conference on the HCI Society of Korea*, 921-923.
- [2] K. S. Won. (2010). *Application Method of Image Restoration based on Augmented Reality to Museum Education*. *Journal Of The Korea Contents Association*, 10(6), 205-212.
- [3] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier & B. MacIntyre. (2001). Recent advances in augmented reality, *IEEE Computer Graphics and Applications*, 21(6), 33-47
- [4] J. Han & S. Lee. (2017). *Fake GPS Detection for the Online Game Service on Server-Side*. *Journal of the Korea Institute of Information Security & Cryptology*, 27(5), 1069-1076.
- [5] W. Liu, B. Leibe, J. Matas, N. Sebe & M. Welling. (2016). SSD: Single Shot MultiBox Detector. *In European conference on computer vision*. (pp. 21-37). Springer, Cham.
- [6] S. Park, U. N. Yoon & G. S. Jo. (2020). The Cut Transition Detection Model Using the SSD Method. *Journal of KIISE*, 47(7), 655-664.
- [7] J. Redmon, S. Divvala, R. Girshick & A. Farhadi. (2016). You only look once: Unified, real-time object detection, *IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, (pp. 779-788)
- [8] S. Ren, K. He, R. Girshick & J. Sun. (2016). Faster R-CNN: towards real-time object detection with region proposal networks. *IEEE transactions on pattern analysis and machine intelligence*, 39(6), 1137-1149.
- [9] Google Inc. (n. d.). *Tensorflow Lite*. (Online) <https://www.tensorflow.org/lite>
- [10] H. Y. Jo. (2020). *A Development of Efficient RoadCrack Detection Model under Mobile Environment*, Graduate School of Management of TechnologyPukyong National University. Busan.

김 한 호(Han-Ho Kim)

[정회원]



- 2018년 9월 : 광운대학교 스마트융합대학원 게임학과 (제임학 석사)
- 2020년 2월 ~ 현재 : 공주대학교 게임디자인학과(박사과정)
- 관심분야 : 증강현실, 가상현실, 게임인공지능, 딥러닝
- E-Mail : hanhokim@gmail.com

이 동 열(Dong-Lyeor Lee)

[정회원]



- 2000년 2월 : 일본 큐슈예술공과대학 예술공학과 정보전달전공(예술공학 석사)
- 2006년 4월 ~ 현재 : 공주대학교 게임디자인학과 교수

- 관심분야 : 게임캐릭터 디자인, 컴퓨터그래픽, 멀티미디어
- E-Mail : ezer@kongju.ac.kr