DBpia

합성영상과 딥러닝 모델을 이용한 크레인의 자세추정

Crane Pose Estimation Using Deep Learning Models and Synthetic Images

저자 (Authors)
박규하 , 홍효성, 정현호, 강호준, 원문철
Gyuha Park, Hyosung Hong, Hyeonho Jeong, Hojun Kang, Mooncheol Won

출처 (Source)
제어로봇시스템학회 논문지 27(4), 2021.4, 312-319 (8 pages)
Journal of Institute of Control, Robotics and Systems 27(4), 2021.4, 312-319 (8 pages)

발행처 (Publisher)
제어로봇시스템학회
Institute of Control, Robotics and Systems

URL
http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE10543295

APA Style
박규하 , 홍효성, 정현호, 강호준, 원문철 (2021). 합성영상과 딥러닝 모델을 이용한 크레인의 자세추정. 제어로봇시스템학회 논문지, 27(4), 312-319.

이용정보 (Accessed)
부산대학교
164.***.126.245
2021/09/10 15:00 (KST)

저작권 안내

DBpia에서 제공되는 모든 저작물의 저작권은 원저작자에게 있으며, 누리미디어는 각 저작물의 내용을 보증하거나 책임을 지지 않습니다. 그리고 DBpia에서 제공되는 저작물은 DBpia와 구독계약을 체결한 기관소속 이용자 혹은 해당 저작물의 개별 구매자가 비영리적으로만 이용할 수 있습니다. 그러므로 이에 위반하여 DBpia에서 제공되는 저작물을 복제, 전송 등의 방법으로 무단 이용하는 경우 관련 법령에 따라 민, 형사상의 책임을 질 수 있습니다.

Copyright Information

Copyright of all literary works provided by DBpia belongs to the copyright holder(s)and Nurimedia does not guarantee contents of the literary work or assume responsibility for the same. In addition, the literary works provided by DBpia may only be used by the users affiliated to the institutions which executed a subscription agreement with DBpia or the individual purchasers of the literary work(s)for non-commercial purposes. Therefore, any person who illegally uses the literary works provided by DBpia by means of reproduction or transmission shall assume civil and criminal responsibility according to applicable laws and regulations.

Journal of Institute of Control, Robotics and Systems (2021) 27(4):312-319
https://doi.org/10.5302/J.ICROS.2021.21.0015 ISSN:1976-5622 eISSN:2233-4335

합성영상과 딥러닝 모델을 이용한 크레인의 자세추정

Crane Pose Estimation Using Deep Learning Models and Synthetic Images

박 규 하, 홍 효 성, 정 현 호, 강 호 준, 원 문 철*
(Gyuha Park[1], Hyosung Hong[1], Hyeonho Jeong[1], Hojun Kang[1], and Mooncheol Won[1,*])
[1]Department of Mechatronics Engineering Chungnam National University

Abstract: This paper suggests a deep learning algorithm for estimating the poses of cranes in an industrial site. A CAD model-based image dataset generation and automatic annotation methods are developed to train deep learning–based bounding box detection and UV mask prediction algorithms for estimating crane pose. Most deep learning frameworks require large amounts of data for training, and it is labor-intensive to make the human annotation data. Since there are no datasets for estimating the poses of cranes in an industrial site, we generate a synthetic crane image dataset using a CAD model. The generated synthetic image dataset is used in a deep learning model after hard augmentation through a method called domain randomization. The algorithm is verified by a real test image dataset made using a crane miniature and ArUco markers.

Keywords: Cane, Object detection, Pose estimation, Synthetic data, Domain randomization, Blender

I. 서론

현재 많은 항만, 산업현장에서 크레인 장비를 많이 사용하고 있다. 하지만 크레인 충돌로 인해 사고가 발생할 경우 큰 재산피해와 인명피해로 이어지게 된다. 2017년 삼성 중공업에서 크레인 충돌 사고로 인해 6명이 사망하는 사고가 있었으며 2020년에는 부산신항에서 화물선과 크레인이 충돌해 큰 재산피해를 입었다. 일반적으로 충돌 사고를 방지하기 위해서는 크레인 운전수에게 신호를 보내는 인원이 배치된다. 하지만 복잡한 작업환경에서는 사람의 시야가 제한될 수 있다. 이러한 위험성으로 인해 2018년부터는 크레인을 운용할 때 크레인 충돌방지 시스템을 반드시 사용하게끔 법이 개정되었다[1].

크레인 충돌방지 시스템에는 종류가 두 가지가 있다. 첫 번째 방식은 IMU (Inertial Measurment Unit), GPS (Global Positioning System) 센서를 이용해 크레인의 움직임을 감지함으로 크레인 간의 충돌을 방지하는 시스템이다[2]. 크레인 마다 센서를 부착하기 때문에 크레인 간의 충돌은 방지할 수 있지만 다른 외부 물체와의 충돌에는 취약하다는 단점이 있다. 두 번째 방법은 Lidar (Light Detection and Ranging) 센서를 사용해 크레인과 물체의 거리를 측정하여 충돌을 방지하는 시스템이다[3]. 하지만 이러한 방식은 눈과 비와 같은 악천후에 취약하고 크레인 간의 정상작동임에도 거리가 가깝다는 이유로 잘못된 알람이 울릴 수 있다는 단점이 있다.

본 연구는 새로운 크레인 충돌방지 시스템을 제안한다. 크레인의 정확한 6자유도 자세를 RGB 영상 데이터와 딥러닝(deep learning) 모델을 통해 예측함으로 크레인 간의 충돌을 방지한다. 기존의 크레인 충돌방지 시스템은 관성 측정 장치, GPS, 라이다 센서와 같이 고가의 센서들이 사용되었다. 하지만 본 연구에서 제안한 방법은 RGB 영상 데이터를 얻을 수 있는 저가의 단안 카메라만을 필요로 한다. 그러나 딥러닝 모델을 학습하기 위해서는 많은 양의 표시작업(annotation)이 수행된 RGB 영상 데이터가 요구되지만 보안 문제로 인해 항만, 산업현장에 위치한 크레인의 RGB 영상 데이터를 얻기는 매우 어렵다. 이러한 문제를 해결하기 위해 본 연구에서는 크레인의 3차원 CAD (Computer Aided Design) 모델을 렌더링(rendering)하여 얻은 합성영상(synthetic image)을 학습 데이터로 사용하였다. 또한, 3차원 CAD 모델의 정보를 이용하여 자세의 표시작업의 자동화가 이루어지기 때문에 표시작업의 어려움도 해결할 수 있다.

*Corresponding Author
Manuscript received February 3, 2021; revised March 2, 2021; accepted March 11, 2021
박규하: 충남대학교 메카트로닉스공학과 대학원생(khp3927@naver.com, ORCID 0000-0001-5972-2559)
홍효성: 충남대학교 메카트로닉스공학과 대학원생(hyosung.hong@cnu.ac.kr, ORCID 0000-0003-4642-4506)
정현호: 충남대학교 메카트로닉스공학과 대학원생(zz4979@cnu.ac.kr, ORCID 0000-0002-7606-8330)
강호준: 충남대학교 메카트로닉스공학과 대학원생(ghwns8720@naver.com, ORCID 0000-0001-8990-1558)
원문철: 충남대학교 메카트로닉스공학과 교수(mcwon@cnu.ac.kr, ORCID 0000-0002-9730-4291)
※ 이 연구는 충남대학교 학술연구비에 의해 지원되었음.

Copyright© ICROS 2021

부산대학교 | IP:164.***.126.245 | Accessed 2021/09/10 15:00(KST)

II. 선행연구

일반적으로 6자유도 자세를 추정하는 딥러닝(deep learning) 모델의 연구들은 LineMOD[4]와 같은 오픈 데이터세트(open dataset)를 이용해 학습을 함과 동시에 알고리즘의 성능을 평가하는 지표로 사용하고 있다. 본 연구에서는 오픈 데이터세트를 그대로 사용할 수 없어 LineMOD[4]와 같은 데이터 생성 방식을 사용하여 데이터세트를 생성하고 학습 데이터로 대체하였다.

6자유도 자세 추정 오픈 데이터세트들을 만드는 과정은 다음과 같다. 먼저 카메라 보정(camera calibration)을 수행한 후 배치된 아루코 마커(aruco marker)[5]들의 중앙에 대상체인 작은 모형을 놓아 단안 카메라로 RGB 영상을 얻는다. 그리고 RGB 영상에서 아루코 마커[5]의 6자유도 자세를 예측함으로 모형의 자세의 표시작업을 수행함으로 실험 데이터를 얻는다. 또한, 모형에 대한 3차원 CAD 모델을 갖고 있기 때문에 합성영상(synthetic image)을 생성하여 학습 데이터로 사용할 수도 있다.

초창기의 자세 추정 알고리즘들은 SIFT (Scale Invariant Feature Transform)[6]와 같은 딥러닝 모델을 사용하지 않고 영상처리 알고리즘만을 사용하는 방법에 의존하였으나 최근에는 복잡한 환경에도 높은 성능을 얻기 위해서 PoseCNN[7]과 같이 RGB 영상에서 직접 회전행렬(rotation matrix)를 추정하는 딥러닝 모델 기반의 방법이 제안되었다. 하지만 이러한 방식은 영상의 깊이 정보가 주어지지 않아 한계가 있었다.

이후에는 딥러닝 모델로 직접 회전행렬을 구하는 대신에, 영상에서 객체의 특징점(key point)들을 먼저 찾아내고, 2차원 영상과 3차원 모델의 특징점들의 대응 관계를 통하여 PnP (Perspective-n-Point)[8] 알고리즘으로 전이행렬(translation matrix)과 회전행렬을 추정하는 방식이 제안되었다. RGB 영상에서 직접 객체의 자세를 추정하는 것 보다는 객체의 자세 변화에 따른 특징점들을 추정하는 것이 보다 쉽기 때문에 성능을 향상시킬 수 있었다.

특징점을 찾아내고 PnP[8] 알고리즘을 수행하는 두 단계 방식은 객체에서 어떠한 특징점들을 찾는가에 따라서 나뉠 수 있다. BB8 (8 Corners of The Bounding Box)[9]과 YOLO6D (You Only Look Once 6D)[10]에서는 3차원 공간에서 객체에 맞닿은 직육면체의 꼭짓점들을 특징점으로 한다. 하지만 직육면체의 꼭짓점들은 객체에서 거리가 있는 곳에 위치하여, 특징점이 RGB 영상의 경계를 넘어서 위치하는 경우가 있다. 또한 객체의 일부가 가려지는 경우에는 특징점을 잘 예측하지 못하여 성능이 떨어지는 문제점이 있다.

이에 대한 해결책으로 DPOD (6D Pose Object Detector and Refiner)[11]에서는 객체의 모든 픽셀이 3차원 모델의 UV map의 값을 추정하도록 하여, 모든 픽셀이 2차원과 3차원의 대응 관계를 추정하는 특징점 역할을 하게 함으로써 자세 추정의 두 번째 단계인 RANSAC (Random Sampling Consensus)[12] 기반의 PnP[8] 알고리즘의 성능을 높였다.

본 연구에서는 DPOD[11]의 방식을 채택하였다. 그리고 UV map을 통해 2차원과 3차원의 대응 관계를 더욱 더 잘 추정하기 위해 기존 방식과 달리 UV 값을 좀 더 효율적으로 생성하는 방법을 도입하였다.

III. 크레인 자세추정 알고리즘

1. 크레인 자세추정 알고리즘의 구성

크레인 자세추정 알고리즘의 구조는 그림 1에 나타나 있듯이 경계상자(bounding box) 탐지 딥러닝 모델인 YOLO-v3 [13]로 크레인의 영역을 찾고 정사각형으로 잘라내어 입력으로 들어가게 된다. 그리고 UNet[14]의 UV 마스크(mask)를 예측한다. 예측된 UV 마스크는 그대로 PnP[8], RANSAC [12] 알고리즘에 적용되지 못한다. 입력 영상의 UV 마스크가 아니라 잘라낸 크레인 영상의 것이기 때문이다. 예측한 UV 마스크는 YOLO-v3[13]로 얻은 크레인의 영역 크기만큼 다시 축소시키고 나머지 배경의 값은 0으로 처리한 후 PnP[8], RANSAC[12] 알고리즘에 들어가 전이행렬(translation matrix)과 회전행렬(rotation matrix)을 얻는다.

2. YOLO-v3 딥러닝 모델

크레인 자세추정 알고리즘의 구조에서 YOLO-v3[13]를 이용해 크레인의 영역을 찾아 정사각형으로 잘라내 UNet [14]에 입력으로 들어가게 된다. YOLO-v3[13]는 Darknet-53 [13]이라는 구조를 따르고 있다. Darknet-53[13]은 이전 버전에 사용된 Darknet-19에 skip connection을 적용하여 합성곱 신경망(CNN, Convolution Neural Net)의 층을 더욱 많이 쌓은 모델이다.

그림 2를 보면 경계상자 출력이 세 개인 것을 확인할 수 있다. Darknet-53[13] 이후에는 FPN (Feature Pyramid Network)

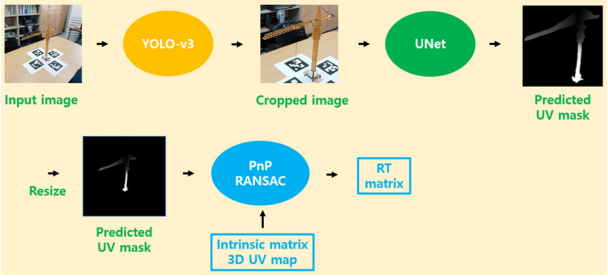


그림 1. 구조의 개요.

Fig. 1. An overview of the architecture.

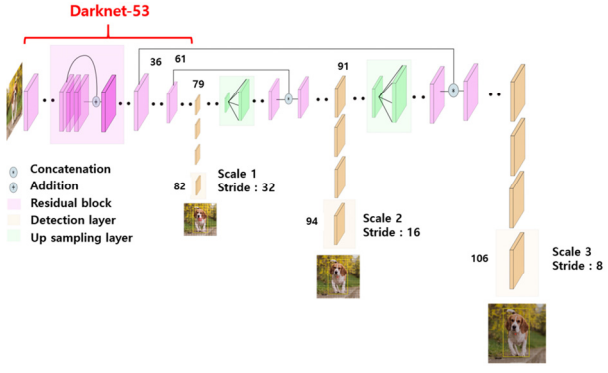


그림 2. YOLO-v3의 구조[13].

Fig. 2. YOLO-v3 architecture[13].

부산대학교 | IP:164.***.126.245 | Accessed 2021/09/10 15:00(KST)

[15] 형식을 따라 정확성과 속도를 높였다. 즉, 각 스케일의 출력마다 그리드 셀(grid cell)의 크기를 다르게 주었으며 앵커박스(anchor box를) 3가지씩 부여했다. 즉, 총 앵커박스는 9개가 되는 것이다. 그리드 셀의 크기가 다른 세 가지 출력을 통해 얻을 수 있는 효과는 Scale 1에서는 그리드 셀이 크기 때문에 큰 물체를 찾게 되며 Scale 3에서는 그리드 셀이 작기 때문에 작은 물체를 잘 찾게 되는 효과를 볼 수 있다.

3. UNet 딥러닝 모델

본 연구에서 사용된 크레인 자세추정 알고리즘의 구조에서는 UV 마스크를 예측하기 위해 UNet[14]를 사용하고 있다. UNet[14]은 시멘틱 세그멘테이션(semantic segmentation) 분야에서 가장 많이 쓰이는 대표적인 딥러닝(deep learning) 모델이다.

그림 3을 보면 일반적인 UNet[14]의 구조는 엔코더(encoder)에 대응되는 디코더(decoder)가 하나이나 본 연구에서는 3개를 사용하였다. PnP[8], RANSAC[12] 알고리즘에 필요한 입력이 마스크, U 마스크, V 마스크이다. 그러므로 디코더도 세 개가 필요하다. 처음에 입력하는 영상의 크기를 줄여 256×256×3의 영상이 들어오게 된다. 그 후에 세 갈래로 나뉘어 출력이 되는데 마스크를 출력하는 디코더는 256×256×2의 채널이 2개인 영상이 출력되고 U, V 마스크를 출력하는 디코더는 모두 256×256×256의 디코더가 출력이 된다. 출력되는 U, V 마스크에서 (i,j,k)의 값은 (i,j)위치의 픽셀에 대응되는 U, V map의 값이 k가 될 확률을 의미한다. 이후에 채널 중에서 (i,j)위치의 최댓값만 남겨서 하나의 채널로 만든 뒤 Ground truth와 비교해 교차 엔트로피(cross entropy) 손실함수(loss function)를 계산한다.

4. PnP, RANSAC 알고리즘

UNet[14]에서 예측한 UV 마스크는 PnP (Perspective-n-Point) [8] 알고리즘에 적용이 된다. PnP[8] 알고리즘은 카메라의 위치와 자세를 알아내기 위해 3개 이상의 특징점(Key point)을 이용하여 계산하는 알고리즘이다. PnP[8] 알고리즘에는 다양한 방법들이 있지만 쉽게 설명할 수 있는 방법인 최소자승법(least square method)으로 PnP[8] 알고리즘을 설명한다.

$$y = f(x, P_M, K) \tag{1}$$

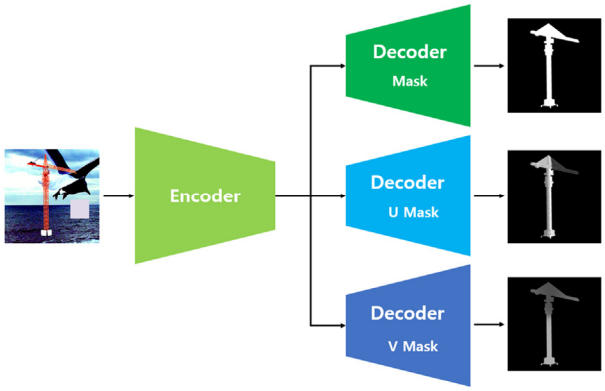


그림 3. UNet의 구조.

Fig. 3. UNet architecture.

식 (1)에서 y는 카메라의 사영함수를 나타낸다. 3개의 변수가 있는데 x는 구하고자 하는 변수 카메라의 회전 각도와 위치를 나타내며 P_M은 특징점들의 3차원 좌표이다.

알고리즘의 순서는 다음과 같다. 먼저 변수 x의 초기값을 무작위로 설정한다. 그리고 식 (2)와 같은 잔여 오류(residual error) E를 계산한다.

$$E = |y - y_0|^2 \tag{2}$$

여기서 y는 x를 입력으로 하여 얻은 값이며 y_0는 영상에서 얻은 특징점을 나타낸다. 계산된 잔여 오류 E는 다음 단계인 함수 미분 근사화 과정에서 dy로 사용된다. 잔여 오류 즉, dy를 계산하였다면 자코비안 행렬(jacobian matrix) J를 식 (3)과 같이 계산한다. 자코비안 행렬을 구하는 이유는 함수 미분 근사화에 필요한 dx를 구하기 위함이다.

$$\begin{aligned} &J = \partial f/\partial x \rightarrow dy = Jdx \\ &dx = J^{+}dy \ (J^{+} = (J^{T}J)^{-1} : pseudo\ inverse\ of\ J) \\ &x \leftarrow x + dx \end{aligned} \tag{3}$$

위와 같은 과정을 잔여 오류 E가 수렴할 때까지 반복하면 x 즉, 카메라의 회전각과 위치를 얻을 수 있다.

최근에는 PnP[8] 알고리즘의 정확도를 높이기 위해서 추가로 RANSAC (Random Sampling Consensus)[12] 알고리즘이 같이 사용되고 있다. 최소자승법은 잔여 오류 E를 최소화하도록 모델을 찾아가지만 RANSAC[12]은 가장 많은 수의 데이터들로 지지를 받는 모델을 선택하는 방법이다. RANSAC[12] 알고리즘은 특히 본 연구와 같이 딥러닝(deep learning) 모델로부터 예측한 특징점을 사용하는 알고리즘에는 필수적이다. 그 이유는 딥러닝 모델에서 예측한 특징점은 이상점(outlier)이 많기 때문이다.

IV. 데이터 생성

1. 실험 데이터 생성

산업현장의 크레인 영상을 실험 데이터로 사용하면 좋지만 보안 문제로 인해 구하기가 어려운 경우가 많다. 그리고 본 알고리즘은 실제 크레인의 CAD (Computer Aided Design) 모델을 활용해 만든 합성영상(synthetic image)만을 학습 데이터로 사용하기 때문에 정확한 수치의 CAD 모델이 필요하지만 실제 현장의 크레인의 CAD 모델을 구하기는 매우 어렵다. 이러한 문제점들을 고려해서 직접 수치를 측정해서 CAD 모델을 만들기 편리한 크레인 모형을 대상으로 선정하였다. 크레인 모형은 움직이지 않고 카메라를 움직여가며 총 100장의 영상을 단안 카메라로 촬영 후 표시작업(annotation)을 진행하였고 표시작업은 마스크(mask), 경계상자(bounding box), 자세(pose) 세 가지로 분류된다.

1.1. 마스크, 경계상자 표시작업

마스크 표시작업은 Labelme[16]라는 오픈소스(open source) 프로그램을 사용하였다. 사람이 직접 점을 찍어 다각형을 만들어 표시작업을 해줄 필요가 있다. 마스크 표시작업이 완료되면 흰색 픽셀(pixel) 좌표의 정보를 이용해 경계상자의 표시작업도 동시에 수행된다. 그림 4는 실제 영상의 마스크 표시작업을 나타낸다.

부산대학교 | IP:164.***.126.245 | Accessed 2021/09/10 15:00(KST)

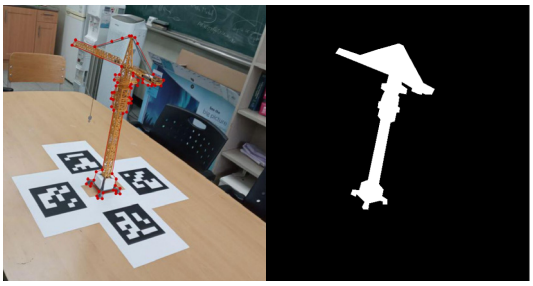


그림 4. 실험 데이터의 마스크 표시작업.

Fig. 4. Mask annotation of test data.

1.2. 자세 표시작업

크레인의 6자유도 자세를 사람이 직접 측정해서 표시작업을 수행하는 것은 불가능하다. 이러한 문제점을 해결하기 위해서 아루코 마커(aruco marker)[5]의 자세를 예측하고 평균을 내어 크레인의 자세 표시작업을 수행한다. 자세 표시작업 예시가 그림 5에 나타나 있다.

카메라 보정(camera calibration)과 카메라 렌즈 보정(camera lens distortion correction)을 수행 후 PnP (Perspective- n-Point) [8] 알고리즘으로 아루코 마커[5]의 6자유도 자세를 추정한다. 그리고 추정한 마커의 6자유도 자세의 평균을 구해 크레인의 6자유도 자세를 구한다.

2. 학습 데이터 생성

산업현장의 크레인의 CAD 모델을 얻는 것 또는 직접 치수를 측정해서 CAD 모델을 만드는 것은 매우 어렵다. 본 연구에서 크레인의 모형을 실험 데이터 대상으로 선정한 이유이다. 산업현장의 크레인보다 크기가 작기에 치수를 측정하기가 편리해 CAD 모델을 만들기가 쉽기 때문이다. 블렌더(blender)[17]라는 오픈소스 렌더링 프로그램으로 크레인 모형의 CAD 모델을 불러오고 렌더링작업과 객체 영역, UV 영역, 자세의 표시작업을 수행하였다. 마지막으로 여러 알고리즘을 거쳐 데이터의 양과 다양성을 확보하였고 이를 통해 학습 데이터 도메인과 실험 데이터의 도메인 간의 거리를 줄였다.

2.1. 학습 데이터의 종류

그림 6에서 첫 번째 행의 영상들은 블렌더[17]에서 생성한 기본 합성영상이다. 기본 합성영상을 생성함과 동시에

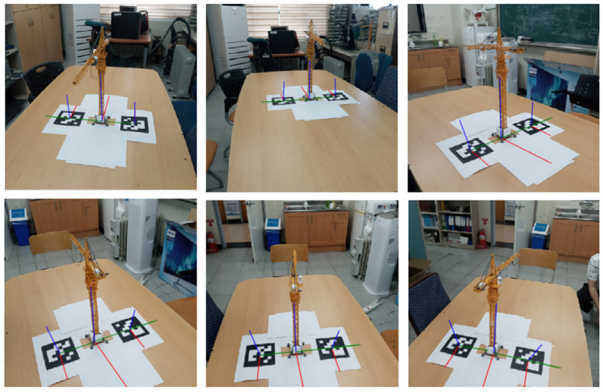


그림 5. 실험 데이터의 자세 표시작업.

Fig. 5. Pose annotation of test data.

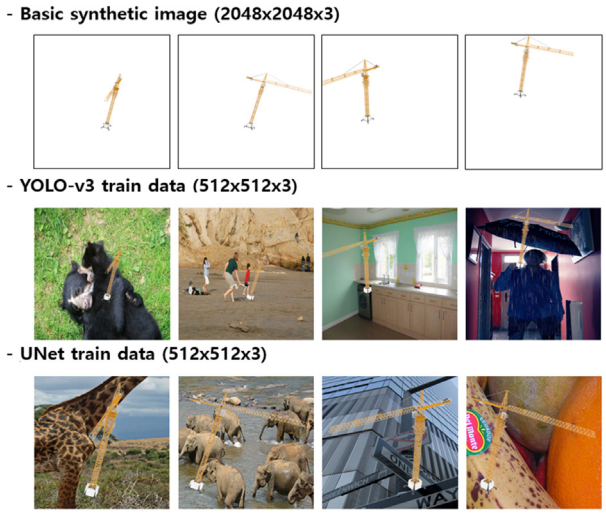


그림 6. 학습 데이터의 종류.

Fig. 6. Type of train data.

CAD 모델의 정보를 이용하여 경계상자, 마스크, 자세의 표시작업이 동시에 수행된다. 두 번째 행은 YOLO-v3[13]의 학습 데이터이며 세 번째 행은 UNet[14]의 학습 데이터이다. UNet[14]의 학습 데이터에서 크레인이 영상에 딱 맞게 들어오도록 만들어준 이유는 YOLO-v3[13]에서 경계상자를 예측함으로 얻은 크레인의 영역이 동일한 비율로 UNet[14]의 입력으로 들어가기 위함이다.

2.2. UV 마스크 표시작업

본 연구에서는 UV map을 통해 2차원과 3차원 간의 대응관계를 추정하기 위해서 PnP (Perspective-n-Point)[8] 알고리즘과 RANSAC (Random Sampling Consensus)[12] 알고리즘을 적용하였다. UV map이란 2차원의 영상을 3차원의 모델에 투영할 때의 근사값에 해당된다.

그림 7은 UV 마스크의 생성과정을 나타낸다. 크레인의 포인트 클라우드(point cloud) 데이터에 크레인의 현재 시점의 전이행렬(translation matrix), 회전행렬(rotation matrix), 그리고 카메라 내부행렬(intrinsic matrix), 외부행렬(extrinsic matrix)을 이용해 2차원 투영(2D projection)을 수행한다. 이때 포인트 클라우드의 좌표 (x, y, z)와 대응되는 투영된 픽셀에 UV 값을 대입한다. 그림 7에서 (d_x, d_y, d_z)는 3차원 모델 임의의 점 P에서 물체의 중점을 향하는 단위벡터이다. 이때 x_m, y_m, z_m을 구하는 이유는 단위벡터의 성분의 값의 크기가 차이가 크기 때문에 평균값을 똑같게 맞추기 위해서이다. d_{x_m}, d_{y_m}, d_{z_m}은 각 단위벡터의 평균이다. DPOD[9]에서는 arctan2, arcsin 함수에 단위벡터인 (d_x, d_y, d_z)가 그대로 입력되지만 본 연구에서는 그 대신 $(d_x x_m, d_y y_m, d_z z_m)$이 입력된다. 그림 8에서 본 연구와 DPOD[9] 방식의 UV 마스크의 차이를 확인할 수 있다.

2.3. 도메인 무작위화

본 연구는 도메인 간의 거리를 줄이기 위해 도메인 무작위화(domain randomization)[18,19] 기법을 적용하였다. 데이터의 다양성을 주기 위해 블렌더[17]에서 생성한 기본 합성영상에서 MS-COCO[20] 영상의 검증 데이터를 무작위로 선

부산대학교 | IP:164.***.126.245 | Accessed 2021/09/10 15:00(KST)

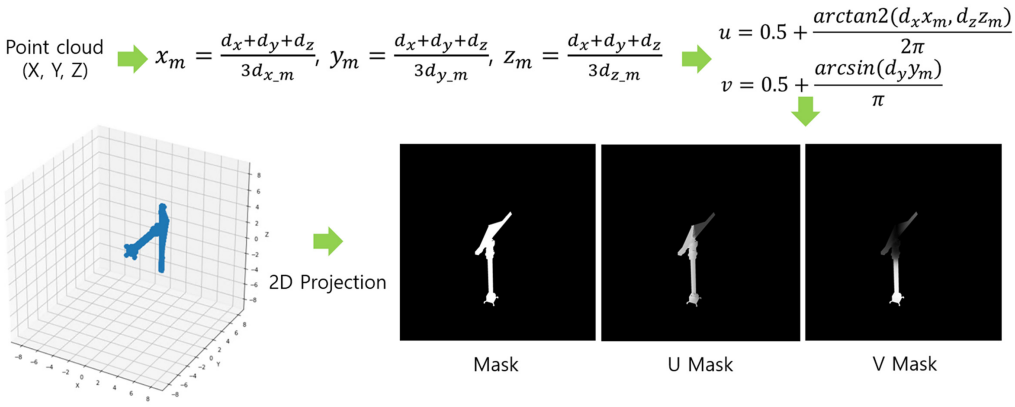


그림 7. UV 마스크 생성과정.

Fig. 7. Process of creating UV mask.

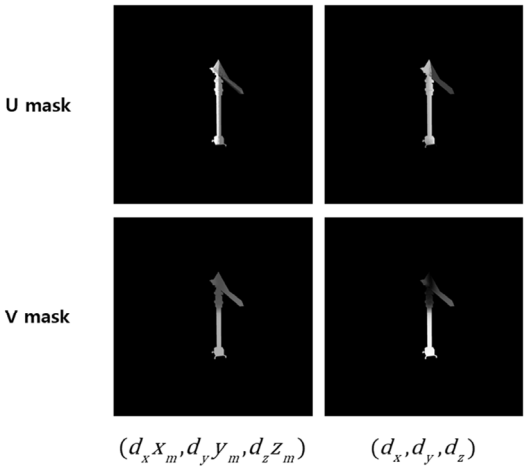


그림 8. UV 마스크 비교.

Fig. 8. Comparison of UV mask.

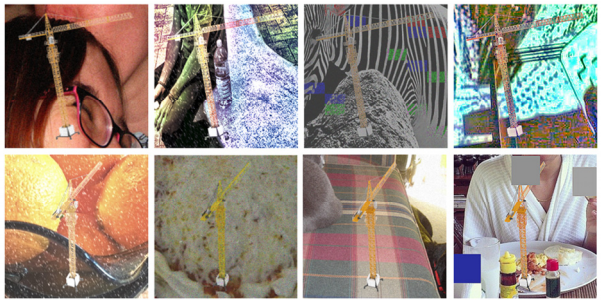


그림 9. 학습 데이터의 도메인 무작위화 예시.

Fig. 9. Domain randomization of train data example.

택하여 배경으로 변경하여 사용하였다. 또한, 효과적으로 도메인 무작위화를 적용하기 위해서 데이터 확장기법(data augmentation)을 사용하였다. 이를 적용하기 위해 오픈소스 라이브러리인 imgaug[21]를 사용하였다. 노이즈(noise) 추가, 밝기, 색상 반전, 컷아웃(cutout)등 총 36가지의 기법이 사용되었고 이 중에서 1개에서 5가지의 기법을 무작위로 선택해 매 영상 마다 적용해 데이터의 개수를 기본 합성영상에 비해 10배로 늘렸다. 적용 예시가 그림 9에 나타나 있다.

V. 학습 및 결과

크레인 모형의 CAD (Computer Aided Design) 모델을 렌더링(rendering)한 합성영상(synthetic image)을 학습 데이터로 사용하여 UNet[14]와 YOLO-v3[13]를 학습하였다. 또한, 실험은 아루코 마커(aruco marker)[5]를 이용하여 크레인 모형의 6자유도 자세의 표시작업(Annotation)을 수행한 영상을 실험 데이터로 사용하였다. 데이터세트 환경은 표 1에 나타나 있다.

1. YOLO-v3 학습결과

그림 10은 YOLO-v3[13]의 손실함수(loss function)의 곡선이다. 도메인 무작위화(domain randomization)[18,19]를 적용하지 않았을 때는 검증 손실함수 곡선이 학습중 발산하는 경향이 있으나 도메인 무작위화[18,19]을 적용하였을 때는 검증 손실함수의 곡선이 발산하지 않고 수렴한다.

표 2는 도메인 무작위화[18,19]를 적용한 경우와 하지 않은 경우의 YOLO-v3[13]의 실제 영상에 대한 실험결과를 보여준다. 도메인 무작위화[18,19]을 적용한 경우가 하지 않은 경우보다 좋은 결과를 보여준다.

2. UNet 학습결과

UV 마스크(mask)의 손실함수 곡선이 그림 11에 나타나 있다. 학습 데이터의 손실함수 값이 도메인 무작위화[18,19]를 적용한 경우 적용하지 않은 경우보다 크게 나왔다. 도메인 무작위화[18,19]를 적용한 경우 합성영상에서 UV 마스크를 예측하는 것이 어렵기 때문이다. 그러나 그림 12를 보면 실제 영상에 적용 시 도메인 무작위화를 적용한 경우 더 좋은 결과를 얻었다.

표 1. Dataset environment.

Table 1. 데이터세트 환경.

기본 합성영상 해상도	2048×2048
학습 영상 해상도	512×512
기본 합성영상 수	2000 (합성영상)
YOLO-v3 학습 영상 수	18000 (합성영상)
UNet 학습 영상 수	18000 (합성영상)
UNet 검증 영상 수	2000 (합성영상)
YOLO-v3 검증 영상 수	100 (실제영상)
최종 실험 영상 수	100 (실제영상)

부산대학교 | IP:164.***.126.245 | Accessed 2021/09/10 15:00(KST)

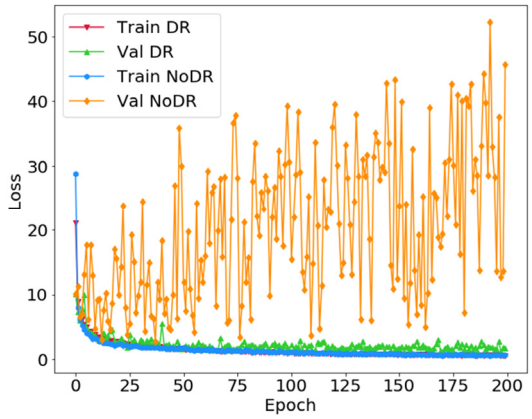


그림 10. YOLO-v3의 손실함수 곡선.
Fig. 10. Loss curve of YOLO-v3.

표 2. Bounding box detection evaluation result using YOLO-v3.
Table 2. YOLO-v3를 이용한 경계상자 탐지 실험 결과.

도메인 무작위화	Precision (IoU≥0.5)	Recall (IoU≥0.5)	AP (IoU≥0.5)
적용	0.97	1.00	1.00
적용 안함	0.87	0.93	0.90

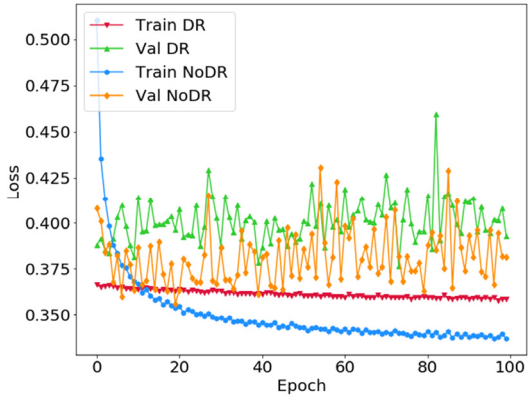


그림 11. UNet의 UV 마스크 손실함수 곡선.
Fig. 11. UV mask loss curve of UNet.

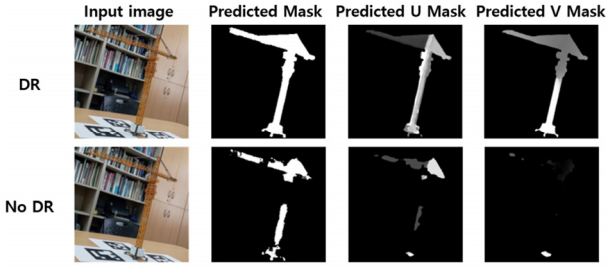


그림 12. 실제 영상에 대한 UV mask 예측 결과 예시.
Fig. 12. UV mask prediction result for real image example.

3. 크레인 자세추정 알고리즘 실험결과

3.1. ADD score 비교

크레인 자세추정 알고리즘의 성능을 평가하는 지표로는

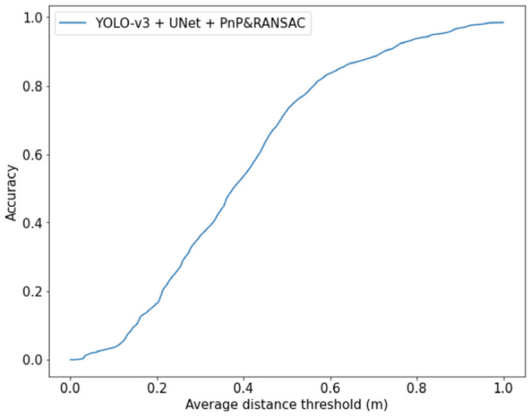


그림 13. 정확도-임계값 곡선.
Fig. 13. Accuracy-threshold curve.

표 3. DPOD와 비교한 본 연구의 ADD score 결과.
Table 3. ADD Score result of this study compared to DPOD.

	크레인 자세추정 알고리즘	DPOD
ADD score	0.60	0.56

ADD score[22]를 사용하였다. 식 (4)는 ADD가 계산되는 과정을 나타낸다. x가 포인트 클라우드(point cloud) 데이터라면 Ground truth인 전이행렬(translation matrix), 회전행렬(rotation matrix)과 예측된 전이행렬, 회전행렬을 각각 포인트 클라우드 데이터에 적용해 대응되는 포인트 간의 거리의 평균을 구하는 방식이다.

$$ADD = \frac{1}{m}\sum_{x\in M}\|(Rx+T)-(\tilde{R}x+\tilde{T})\| \quad (4)$$

ADD score[22]는 대응되는 포인트 간의 거리가 임계값(threshold) 이내이면 정답이라 가정하고 정확도(accuracy)를 구한 정확도-임계값 곡선의 넓이를 말한다. 그림 13의 곡선의 넓이를 계산한 결과 ADD score[22]는 0.60이 나왔다.

표 3은 생성한 실험 데이터에 DPOD[11]를 적용한 결과와 본 연구의 크레인 자세추정 알고리즘의 결과를 ADD score[22]로 비교하였다. DPOD[11] 이상의 결과를 보여줌으로 본 알고리즘의 성능을 입증한다.

3.2. 오일러 각(Euler angle) 기반 자세추정 오차

크레인 자세추정 알고리즘을 실험 데이터에 적용하였을 시 예측한 회전행렬을 오일러 각(roll, pitch, yaw angle)으로 변경하여 Ground truth 자세 각과의 오차값을 계산하여 보았다. 전체 100개 실험 영상 각각에 대하여 3가지 각도의 추정 오차의 평균을 구하고, 다시 이를 100개에 대하여 평균과 표준 편차를 구한 결과, 평균은 20.03도이며 표준편차는 13.68도가 산출되었다.

3.3. 크레인의 자세추정 결과 시각화

크레인의 자세추정 결과를 시각화한 것이 그림 14에 나타나 있다. 첫 번째 행의 영상이 아루코 마커(aruco marker)[5]로 표시작업(annotation)을 수행하여 얻은 Ground truth 크레인 자세를 시각화한 것이고 두 번째 행의 영상이 크레인 자세추정 알고리즘으로 예측한 크레인 자세를 시각화한 것이다.

부산대학교 | IP:164.***.126.245 | Accessed 2021/09/10 15:00(KST)

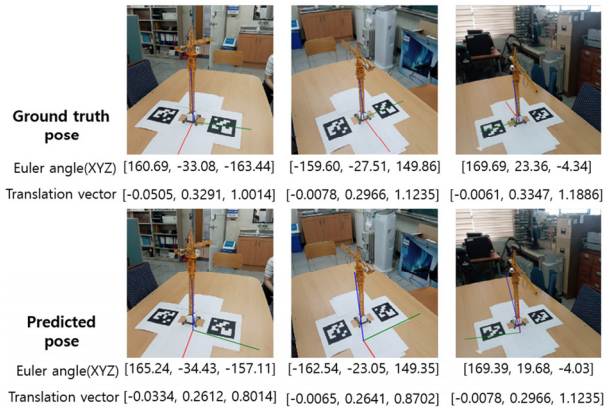


그림 14. 크레인 모형 자세추정의 시각화.

Fig. 14. Visualization of crane miniature pose estimation.

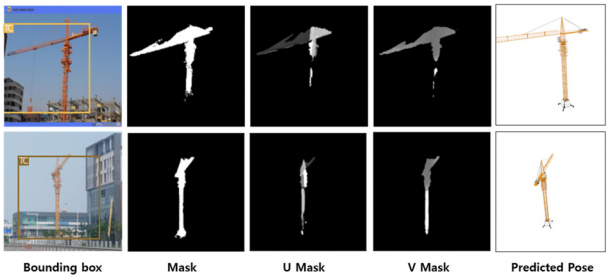


그림 15. 실제 크레인 자세추정의 시각화.

Fig. 15. Visualization of real crane pose estimation.

마지막으로, 실제 산업현장의 크레인의 자세를 예측한 것이 그림 15에 나타나있다. 표시작업이 수행되지 않은 데이터이기 때문에 예측한 자세와 Ground truth 자세의 오차를 수치적으로 비교할 수는 없었다.

VI. 결론

본 연구에서는 크레인의 충돌방지 시스템을 위한 CAD (Computer Aided Design) 모델을 이용한 합성영상(synthetic image) 생성 및 자동 표시작업(annotation)과 크레인의 6자유도 자세추정 알고리즘을 제안하였다. 실험결과 CAD 모델의 대상인 크레인 모형의 자세추정은 기존의 합성영상 기반 딥러닝 자세추정 연구 결과보다 비교적 좋은 결과를 얻었다. 또한, 도메인 무작위화(domain randomization)기법이 학습 데이터와 실험 데이터 간의 도메인 간의 거리를 줄이는 데 큰 효과가 있음을 확인하였다. 산업현장의 유사한 크레인에 대하여도 CAD 모델을 확보하고 있다면, 실제 데이터 없이 합성영상 만으로 자세추정 알고리즘을 개발할 수 있을 것이다. 추후 연구로는 자세보정(pose refinement) 딥러닝 모델을 추가하여 정확도를 높이고 도메인 무작위화를 데이터 전처리 과정에서 적용하는 것이 아니라 데이터 확장기법의 변수들을 학습을 통해 찾아가 알고리즘의 성능을 높일 계획이다.

REFERENCES

[1] Partial amendment to the enforcement regulations of the Occupational Safety and Health Act, https://www.law.go-.kr/LSW/nwRvsLsInfoR.do?lsiSeq=203066

[2] MCAS: Crane anti-collision system based on IMU and GPS, https://www.mcas.ai/solutions/heavy-equip

[3] NOCOL: Crane anti-collision system based on LiDAR, http://www.futureman.co.kr/CraneAntiCollision

[4] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model-based training, detection and pose estimation of texture-less 3d objects in heavily cluttered scenes," *Asian Conference on Computer Vision*, pp 548- 562, 2012.

[5] D. Hu, D. DeTone, and T. Malisiewicz, "Deep charuco: Dark charuco marker pose estimation," *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8436-8444, 2019.

[6] D. G. Lowe, "Object recognition from local scale-invariant features," *Proc. of the Seventh IEEE International Conference on Computer Vision*, 1999.

[7] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv: 1711.00199*, 2017.

[8] L. Shiqi, X. Chi, and X. Ming, "A robust o(n) solution to the perspective-n-point problem," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1444-1450, 2012.

[9] M. Rad and V. Lepetit, "BB8: A scalable, accurate, robust to partial occlusion method for predicting the 3d poses of challenging objects without using depth," *Proc. of the IEEE International Conference on Computer Vision*, pp. 3828-3836, 2017.

[10] B. Tekin, S. N. Sinha, and P. Fua, "Real-time seamless single shot 6d object pose prediction," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 292-301, 2018.

[11] S. Zakharov, I. Shugurov, and S. Ilic, "DPOD: 6D pose object detector and refiner," *Proc. of the IEEE/CVF International Conference on Computer Vision*, pp. 1941-1950, 2019.

[12] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the Association for Computing Machinery*, vol. 24, no. 6, pp. 381-395, Jun. 1981.

[13] J. Redmon and A. Farhadi, "Yolov3: an incremental improvement," *arXiv preprint arXiv: arXiv:1804.02767*, 2018.

[14] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention*, pp. 234-241, 2015.

[15] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object de-

부산대학교 | IP:164.***.126.245 | Accessed 2021/09/10 15:00(KST)

tection," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2117-2125, 2017.

[16] Labelme: Annotation program for mask annotation, https://github.com/wkentaro/labelme

[17] Blender: rendering program for generating a synthetic image dataset, https://www.blender.org

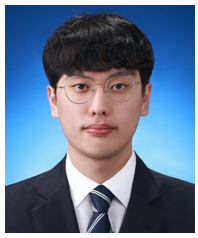
[18] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," *International Conference on Intelligent Robots and Systems*, vol. abs/1703.06907, 2017.

[19] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield. "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," *the IEEE Conference on Computer Vision and Pattern Recognition Workshop on Autonomous Driving*, pp. 969-977, 2018.

[20] T. Lin, M. Maire, S. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. Lawrence Zitnick, "Microsoft COCO: common objects in context," *European Conference on Computer Vision*, pp. 740-755, 2014.

[21] imgaug: The Python library for image augmentation, https://github.com/aleju/imgaug

[22] S. Hinterstoisser, V. Lepetit, S. Ilic, S. Holzer, G. Bradski, K. Konolige, and N. Navab, "Model based training, detection and pose estimation of texture-less 3D objects in heavily cluttered scenes," *Asian Conference on Computer Vision*, 2012.



박 규 하

2019년 충남대 메카트로닉스공학과 학사. 2019년~현재 동 대학원 석사과정 재학 중. 관심분야는 인공지능을 이용한 영상처리.

홍 효 성

2017년 충남대 메카트로닉스공학과 석사. 2017년~현재 동 대학원 박사 과정 재학 중. 관심분야는 자율 주행 차량 제어 및 인공지능을 이용한 영상처리.

정 현 호

2018년 충남대 메카트로닉스공학과 학사. 2018년~현재 동 대학원 석사과정 재학 중. 관심분야는 인공지능을 이용한 포인트 클라우드 데이터 처리.

강 호 준

2020년 충남대 메카트로닉스공학과 학사. 2020년~현재 동 대학원 석사 과정 재학 중. 관심분야는 자율 주행 트랙터 제어 및 인공지능을 이용한 영상처리.

원 문 철

1983년/1985년 서울대학교 조선공학 학사/석사. 1995년 UC Berkeley 기계공학 박사. 1995년~현재 충남대학교 메카트로닉스공학과 교수. 관심분야는 제어 및 인공지능.

부산대학교 | IP:164.***.126.245 | Accessed 2021/09/10 15:00(KST)