

Design of visitor counting system using edge computing method

Jung-Jun Kim*, Min-Gyu Kim*, Ju-Hyun Kim*, Man-Gi Lee*, Da-Young Kim*

*Researcher, Korea Institute of Robotics & Technology Convergence, Pohang, Korea

*Principal Researcher, Korea Institute of Robotics & Technology Convergence, Pohang, Korea

*Senior Researcher, Korea Institute of Robotics & Technology Convergence, Pohang, Korea

*Senior Researcher, Korea Institute of Robotics & Technology Convergence, Pohang, Korea

*Researcher, Korea Institute of Robotics & Technology Convergence, Pohang, Korea

[Abstract]

There are various exhibition halls, shopping malls, theme parks around us and analysis of interest in exhibits or contents is mainly done through questionnaires. These questionnaires are mainly depend on the subjective memory of the person being investigated, resulting in incorrect statistical results. Therefore, it is possible to identify an exhibition space with low interest by tracking the movement and counting the number of visitors. Based on this, it can be used as quantitative data for exhibits that need replacement. In this paper, we use deep learning-based artificial intelligence algorithms to recognize visitors, assign IDs to the recognized visitors, and continuously track them to identify the movement path. When visitors pass the counting line, the system is designed to count the number and transmit data to the server for integrated management.

▶ **Key words:** Application, Object Detection, Object Tracking, Image Processing, Edge Computing

[요 약]

우리 주위에 다양한 전시관, 쇼핑몰, 테마파크 등이 있으며 실제 전시하고 있는 전시물, 콘텐츠에 대한 관심도, 흥미도에 대한 분석은 설문 정도에만 이루어지고 있다. 이러한 설문은 주로 피 설문자의 주관적인 기억에 의존하고 있어서 잘못된 통계 결과를 얻을 수 있는 문제가 있다. 따라서 방문객의 동선 추적과 수를 카운팅 하여 흥미가 떨어지는 전시 공간 파악이 가능하며 이를 통해 교체가 필요한 전시물에 대해 정량적 자료로 사용이 가능하다. 본 논문에서는 딥러닝 기반의 인공지능 알고리즘을 이용하여 방문객을 인식하고, 인식된 방문객에 아이디를 할당하여 이를 지속적으로 추적하는 방식으로 동선을 파악한다. 이때 방문객이 카운팅 라인을 통과하게 되면 그 수를 카운팅 하고, 데이터는 서버에 전송하여 통합 관리할 수 있도록 시스템을 설계하였다.

▶ **주제어:** 응용소프트웨어, 객체인식, 객체추적, 영상처리, 엣지 컴퓨팅

• First Author: Jung-Jun Kim, Corresponding Author: Min-Gyu Kim, Ju-Hyun Kim, Man-Gi Lee, Da-Young Kim

*Jung-Jun Kim (jjkim@kiro.re.kr), Korea Institute of Robotics&Technology Convergence

*Min-Gyu Kim (mingyukim@kiro.re.kr), Korea Institute of Robotics&Technology Convergence

*Ju-Hyun Kim (myearch@kiro.re.kr), Korea Institute of Robotics&Technology Convergence

*Man-Gi Lee (mangi11@kiro.re.kr), Korea Institute of Robotics&Technology Convergence

*Da-Young Kim (dayoung@kiro.re.kr), Korea Institute of Robotics&Technology Convergence

• Received: 2022. 06. 03, Revised: 2022. 07. 21, Accepted: 2022. 07. 21.

I. Introduction

인공지능에 대한 기술이 발전함에 따라 기존의 다양한 사물인터넷(IoT : Internet of Things) 센서를 활용하여 공간에 대한 정보를 얻는 것뿐만 아니라 영상정보를 활용하여 다양한 정보를 분석하려는 연구가 진행되고 있다. 특히, 저가형 RGB-D 카메라 기술이 발전함에 따라 영상 내에서 객체를 탐지하고 추적하는 인공지능 기반의 연구가 많이 진행되고 있다[1,2]. 이러한 기술들은 실시간 무인감시 및 CCTV, 스마트 빌딩 등 다양한 영상 분야에서 사용되고 있으며, 국내에서도 해당 기술로 많은 스타트업 회사가 생기고 있을 정도로 연구 개발이 크게 급증하고 있다.

센서 기반의 분석 기술들은 우리 생활환경의 다양한 분야에 자리 잡고 있다. 쇼핑몰이나 백화점, 전시관 등의 많은 사람들이 지나 다니는 곳에 다양한 센서를 사용하여 방문객의 통과 여부를 카운팅 하고, 이동 패턴을 분석하는 등 다양한 데이터를 얻는데 도움을 줄 수 있다. 이러한 자료는 전시물이나 상품에 대한 효과적인 공간 배치를 하는데 근거 자료로 사용 할 수 있다[3,4].

그러나 인공지능과 사물인터넷의 발전으로 발생하는 데이터의 양은 기존의 중앙 집중화된 컴퓨팅 구조에서는 처리 할 수 없는 수준에 이르렀고, 그에 따른 네트워크 과부하와 보안 문제가 발생하고 있다[5]. 이러한 문제를 해결하기 위해 데이터가 발생하는 기기에서 실시간으로 처리하여 데이터의 양을 줄 일수 있는 컴퓨팅(Edge Computing) 기술이 대두되고 있다. 특히 GPU 하드웨어의 선두 기업인 Nvidia에서 제작한 Jetson 시리즈는 소형로봇, 지능형 드론, 스마트 카메라기반의 지능형 비디오 시스템, 센서 데이터 처리 등 다양한 임베디드 환경에서 최적의 성능을 제공하고 있다[6]. 이러한 하드웨어를 기반으로 본 연구에서는 RGB 카메라와 함께 딥러닝 연산이 가능한 다수의 Nvidia Jetson Xavier를 사용하여 방문객의 동선 추적과 그 수를 카운팅 하는 엣지 컴퓨팅 시스템을 제안 한다.

본 연구의 구성은 2장에서 방문객 인식을 위한 Object Detection 및 Tracking에 대한 배경지식을 살펴보고 3장에서 Edge Computing System설계와 카운팅 알고리즘, 데이터 통신 프로토콜에 대해 기술 한다. 4장에서는 구현한 시스템을 기반으로 실증공간에 적용한 실험에 대해 설명하고 5장에서 결과에 대해 기술한다.

II. Preliminaries

1. Object Detection

객체 탐지(Object Detection)는 이미지나 비디오에 포함되어 있는 물체에 대해 해당 물체가 어떤 물체인지를 분류하고 그 물체가 있는 위치를 찾아내는 것을 말한다. 최근 나오는 대부분의 딥러닝 모델들은 합성곱 신경망(CNN, Convolutional Neural Networks)을 사용하고 있으며 대표적인 몇 가지를 소개한다.

R-CNN(Region based Convolutional Neural Network)은 미리 ImageNet Dataset[7]을 학습시켜 놓은 CNN을 통과 시켜 특징 벡터를 추출하고 이 추출된 벡터를 가지고 각각의 클래스마다 학습시켜 놓은 SVM Classifier를 통과시켜 객체를 분류하고, Regression을 통해 위치를 조정하는 방식으로 동작한다[8]. 객체 인식 분야의 CNN기반으로 이루어진 초기 모델로 연산량이 많아 이를 보완하는 Fast R-CNN이란 모델이 제안 되었다[9]. Fast R-CNN은 R-CNN에서 Selective Search를 통해 얻어진 2천개의 영역에 대해 CNN을 수행하여 Feature Map을 얻는 대신, 원본 이미지에 대해 CNN을 통해 Feature Map을 얻은 후, 이 Feature Map으로 부터 ROI Projection을 통해 ROI(Region of Interest)를 얻는다. 각각의 ROI 영역에 부분만 max-pooling을 적용하는 ROI Pooling을 통해 고정된 길이의 Feature Vector를 추출하고, Fully Connected Layer를 거쳐 softmax classification으로 class를 분류하고 bounding box regression을 수행하여 위치를 찾는 방법을 제안했다. 그러나 여전히 region proposal을 selective search로 수행하는 것은 GPU를 사용하기엔 적합한 구조가 아니라 training, testing하는 시간이 매우 오래 걸린다는 단점이 있다. 이러한 2-Stage Detector는 regional proposal과 classification이 순차적으로 이루어져 정확도는 높지만 속도가 느리다는 문제가 있다. 반면, 하나의 신경망에서 물체의 위치와 종류를 예측하는 1-Stage Detector는 정확도는 낮지만 속도가 빠르다는 장점을 갖는다.

Yolo(You Only Look Once)는 이미지 전체에 대해서 하나의 신경망이 한 번의 계산만으로 객체의 위치와 종류를 대략 1초에 45frame을 처리하는 속도로 예측해 낸다 [10]. 기존의 2-Stage Detector와는 달리 Yolo는 예측하고자 하는 이미지를 Grid Cell 단위로 나누고 각 Cell 마다 객체를 예측한다. 그리고 미리 설정된 Boundary box를 통해 객체의 위치(x, y), 객체의 크기(w, h), confidence score를 포함한 총 5개의 인자를 통해 물체

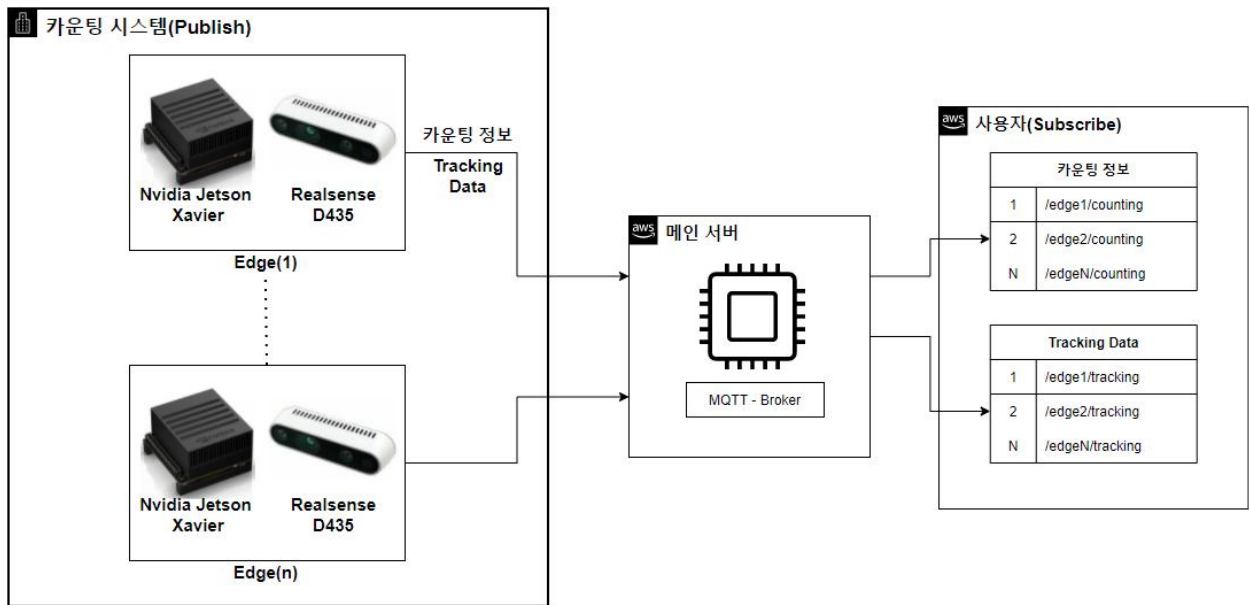


Fig. 1. System Architecture

를 검출 한다. 현재 많은 연구자들이 지속적인 성능 개선을 거쳐 현재 논문으로 게재된 버전은 Yolo v4까지 공개가 되어 있다[11].

2. Object Tracking

객체 추적(Object Tracking)은 영상 내에서 관심 물체를 인식하고 그 물체가 무엇인지를 찾아내는 것뿐만 아니라, 이전 프레임과의 비교를 통해 동일 물체를 매칭시키는 등의 여러 가지 기술들을 사용하여 매 프레임마다 추적하는 것을 말한다. 객체 추적은 크게 Point Tracking, Kernel based Tracking, Silhouette based Tracking 세 가지로 나누어진다[12].

Point Tracking은 움직이는 객체의 특징 점으로부터 Kalman Filter, Particle Filter 등을 사용하여 물체를 추적한다. Kernel Based Tracking은 한 프레임으로부터 다음 프레임 상에서 객체 영역으로 표현되는 움직임을 계산하여 추적 하는 방식으로, 대표적인 알고리즘으로는 Mean Shift Method가 있다[13]. 마지막으로 Silhouette Based Tracking은 주로 손, 손가락, 어깨와 같은 복잡한 모양을 추적할 때 사용한다. 객체의 모양을 매칭 시키는 방법을 사용하거나 외곽선을 이용하여 추적하는 기술이 이에 해당한다.

3. Edge Computing

기존에 전통적인 중앙처리 방식인 클라우드 컴퓨팅과 다르게 엣지 컴퓨팅(Edge Computing)은 종단 기기에서

발생하는 데이터를 중앙처리 기기에 보내지 않고 데이터가 생성되는 기기에서 처리하는 패러다임 이다[18]. 이러한 엣지 컴퓨팅의 경우 대용량 데이터를 전송을 필요로 하지 않아 응답 시간이 빠르고 그에 따른 네트워크 사용 비용을 줄일 수 있다. 또한 새로운 장치가 추가 될 때 전용 데이터 서버를 지속적으로 구축하거나 확장해야 할 필요가 적다. 데이터를 중앙 서버에 저장시키는 것이 아니라 각 디바이스에 분산되어 있기 때문에 쉽게 해킹이 가능한 중앙처리 방식과는 다르게 보안 측면에서 위험이 적다는 장점이 있다. 이러한 엣지 컴퓨팅을 적용한 사례로 비디오 분석에 관한 연구에서는 검증된 딥러닝 모델의 사이즈를 줄이거나 최적화 하여 기존 모델 대비 비슷한 성능을 내면서 Frame rate을 향상 시키는 연구를 통해 다양한 어플리케이션이 개발 되고 있다[20-23].

III. Implementation

본 절에서는 Nvidia Jetson AGX Xavier Development kit와 Realsense D435 RGBD 카메라를 사용하여 지나가는 방문객의 수를 카운팅 하는 알고리즘과 데이터 통신 설계 방법에 대해 설명 한다.

1. System Structure

전체 시스템 구조는 Fig. 1과 같이 카운팅 시스템(Publisher), 메인 서버, 사용자(Subscriber) 총 세 개의

파트로 나누어 설계했다. 카운팅 시스템은 다시 각각 Xavier 보드와 Realsense 카메라를 하나의 엣지(Edge)로 묶어서 구성했다. 엣지에서 나오는 카운팅 정보와 Tracking Data는 메인 서버로 전송 되는데, 이때 서버는 MQTT(Message Queuing Telemetry Transport) 통신을 사용하고, 이를 위해 메인 서버를 Broker로 구성 했다 [14]. 사용자는 이 서버에 접속하여 엣지 별로 카운팅 정보와 Tracking Data를 수신 받는다.

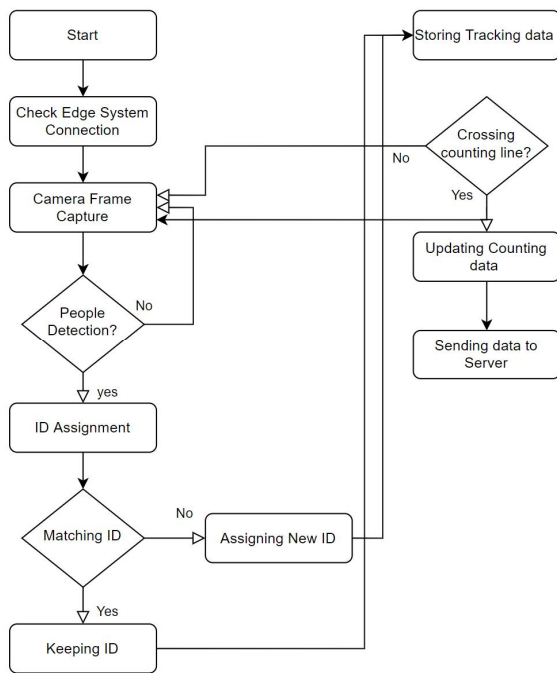


Fig. 2. System Flow Chart

2. People Detection and Tracking

피플 카운팅 시스템은 Fig. 2 와 같은 흐름도로 구성 하였다. 먼저 서버를 통해 사용자는 엣지와 서버의 현재 연결된 상태를 확인한다. 각 엣지는 카메라에서 RGB 영상을 받아 딥러닝 기반의 사람을 인식하는 알고리즘을 수행하고, 각각의 인식된 사람마다 ID를 부여 한다. 이때, 알고리즘은 MOT Challenge의 2DMOT15에서 BestModel로 선정된 FairMOT 모델을 사용하였다[15,16]. Fig 3과 같은 구조로 이루어져 있으며 backbone을 통해 feature map을 얻어 사람에 대한 bounding box를 추정한다. 각 bounding box에 있는 사람에 대해 identity feature vector를 매핑 하는 방식으로 re-identification을 수행한다. 인식된 사람에게 ID를 부여 할 때, 이미 할당된 ID를 갖는 사람과 매칭을 통해 일치하면 기존 ID를 그대로 유지하고, 일치하지 않으면 새 ID를 할당하게 된다. 이 모

델에서는 백본으로 DLA-34를 사용하고 MOT dataset을 학습한 weight를 사용하였다[17]. 할당된 ID를 갖는 사람은 Tracking 알고리즘을 통해 현재 위치를 포함한 10개의 frame에 대한 데이터를 저장하도록 구현했다. 이 데이터를 기반으로 Tracking하는 사람이 카운팅 라인을 통과 했는지 여부를 판단하여 그 수를 세고, 이 정보를 서버에 전송하는 방식으로 구현 하였다.

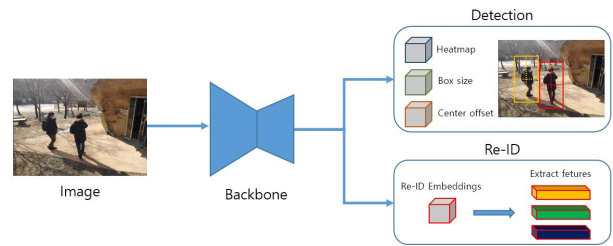


Fig. 3. FairMOT Architecture Overview

3. People Counting Algorithm

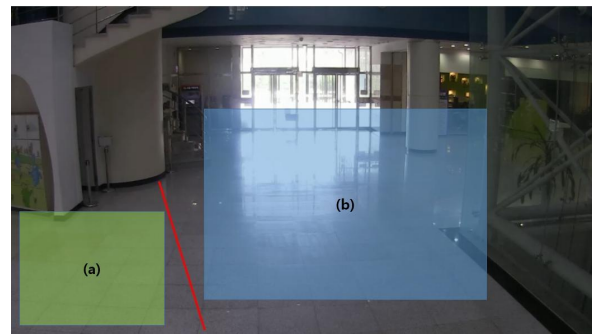


Fig. 4. People counting experimental space and setting to count people moving from (b) to (a)

사람의 수를 카운팅하기 위해서는 이미지 내에 counting line을 기준으로 사람이 통과를 했는지에 대해 확인이 이루어 져야 한다. Fig. 4은 실험 공간 내부의 사진으로 (a), (b) 두 개의 구역으로 나누었다. (b)구역에서 카운팅 라인(빨간색 선)을 지나 (a)구역으로 이동 할 때 카운팅이 되도록 구현 하였다.

Table 1. People tracking sample data

ID	Counting Flag	Tracking Position
		frame(n-10)~frame(n)
n	0, 1, -1	$[(x_{n-10}, y_{n-10}), (x_{n-9}, y_{n-9})]$
		...
		$[(x_{n-1}, y_{n-1}), (x_n, y_n)]$

위 장에서 언급했듯이, 사람이 인식되고 ID를 부여 받게 되는데 각 ID에 따라서 Table 1과 같은 데이터 구조를 갖

게 된다. 먼저 ID 인덱스는 인식된 사람에 따라 초기에 부여되는 고유한 값이고 Counting Flag는 0, 1, -1 세 가지로 분류 하였다. 0은 초기 값이고, 첫 인식된 위치를 바탕으로 (a)구역에 있으면 1로, (b)구역이면 -1 값을 주었다. 따라서 (b)에서 (a)로 이동하는 사람은 Flag값이 -1을 부여 받은 ID에 한정하여 카운팅이 되도록 하였다. 그 다음, Tracking Position의 경우 현재 frame에서 10 frame전의 위치 정보까지 저장하도록 구조를 만들었다. 이는 카운팅 라인 부근에서 사람이 겹쳐져 있어서 영상에서 보이지 않아 Tracking에 실패하여 ID를 순간적으로 잃어버리는 경우나, 빠른 속도로 사람이 움직여서 ID 매칭에 실패하여 새로운 ID를 부여 받는 경우 발생하는 중복 카운팅을 방지하기 위해 10개의 위치정보를 저장하고 있는 ID에 한해서 카운팅이 이루어지도록 구조를 설계하였다.

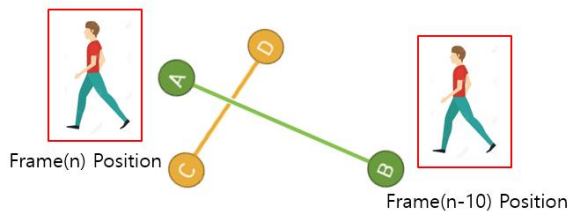


Fig. 5. People counting based on Counter Clock Wise algorithm

Fig 5에서 점C와 점D를 잇는 선분 CD를 카운팅 라인이라 하고, 인식된 사람이 점B에서 A로 이동하는 사람의 수를 카운팅하기 위해 본 연구에서는 CCW(Counter Clockwise)알고리즘을 사용하였다. CCW알고리즘은 점 C, D, B를 순서대로 봤을 때 반시계 방향으로 놓여있으면 양수를, 시계방향이면 음수를 반환한다.

$$x_1y_2 + x_2y_3 + x_3y_1 - (x_2y_1 + x_3y_2 + x_1y_3) \cdots (1)$$

식 (1)은 CCW알고리즘에서 사용되는 식으로 Fig 4에서 세 개의 점 $C(x_1, y_1)$, $D(x_2, y_2)$, $B(x_3, y_3)$ 일 때, 시계 방향에 있으므로 음수 값을 반환 한다. 따라서 Table 1에 나와 있는 Counting Flag는 -1 값을 할당해 주어 카운팅이 되기 전 상태를 나타낸다. 점B에서 점A로 사람이 이동하면 반시계 방향이 되어 CCW알고리즘 반환 값이 양수로 변하게 되고 이 때 Counting Flag는 1로 변하면서 카운팅이 되는 구조로 구현하였다. 각 사람은 고유한 ID를 부여 받기 때문에 영상 내에서 카운팅 라인을 거꾸로 통과하거나 다시 지나 치더라도 카운팅이 되지 않도록 중복 카운팅을 방지하였다.

4. Data Communication Protocol

Fig 1에서 각 엣지 별로 카운팅 정보와 Table 1에 정의된 Tracking Position 정보를 MQTT 브로커 서버에 전달하고 사용자가 필요한 정보를 Subscribe하는 방식으로 통신 구조를 설계했다.

사용자가 원하는 엣지 디바이스에서 나오는 카운팅 정보와 Tracking 데이터에 대해 선택적으로 정보를 가져올 수 있도록 Table 2와 같이 구성하였다. Counting Topic에서는 카운팅을 통과하는 사람의 수 정보를 얻고, Tracking Data Topic으로부터 사람의 동선 파악이 가능하도록 구현하였다. 이러한 Subscribe/Publish 형태의 구조를 취함으로써 각 엣지 디바이스는 다른 디바이스와 별도의 연결 없이 독립적으로 동작이 가능하다. 새로운 엣지 디바이스를 추가 할 경우 서버와 동일한 네트워크에 연결하여 Table 2와 같이 Edge Number만 정의 해 준다면 해당 디바이스에서 얻은 카운팅, Tracking 정보를 사용자가 서버로 Topic을 보내 쉽게 얻을 수 있다는 장점이 있다. 또한 각 엣지 디바이스에서 획득한 카운팅, Tracking 데이터는 실시간으로 서버에 전송되는 것이 아니라 Topic 명령을 통해 필요할 때 한번 전송 받을 수 있기 때문에 네트워크 트래픽을 감소시킬 수 있다.

Table 2. MQTT Message Protocol

Edge Number	Counting Topic
	Tracking Data Topic
1	/edge1/counting
	/edge1/tracking
2	/edge2/counting
	/edge2/tracking
n	/edge(n)/counting
	/edge(n)/tracking

IV. Experiment Results

본 연구에서 제안하는 카운팅 시스템을 실제 전시관 내에 설치하여 테스트를 진행했다. 카운팅 알고리즘이 수행되는 엣지 디바이스는 Nvidia Jetson AGX Xavier Development Kit과 Realsense D435카메라를 사용하고, Ubuntu 18.04 환경에서 python3.6, tensorflow 1.x로 사람 인식 및 알고리즘을 구성하고, 데이터를 획득하는 메인 서버는 ubuntu18.04 환경의 PC를 사용했다.

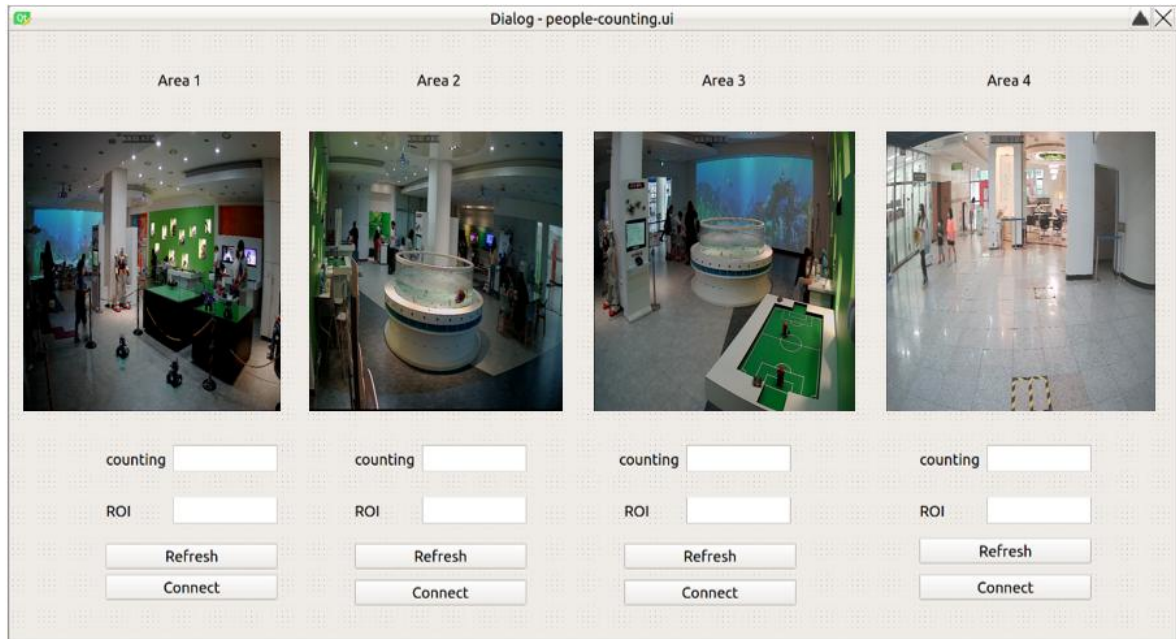


Fig. 7. People Counting System Monitoring Application GUI

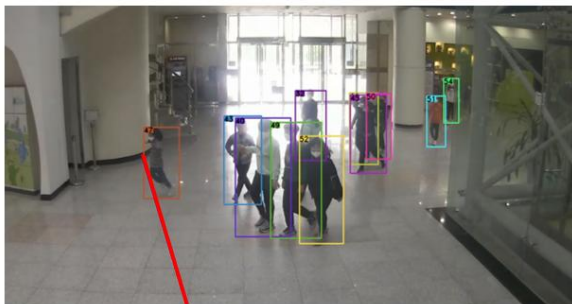


Fig. 6. People counting system demonstration

Fig. 6는 전시관 실증 공간 내 카메라를 설치하여 사람이 지나가는 영상을 녹화하고 카운팅 알고리즘을 수행한 이미지를 캡처한 것이다. 사람인식 및 추적 알고리즘을 수행할 때 필요한 최적의 파라미터를 찾기 위해 녹화 영상을 기반으로 테스트를 진행하였다.

Table 3. Accuracy and FPS results according to skip frame change

skip frame	fps	accuracy (%)	Counting People
1	15	100	39
2	19	100	39
3	21	92.3	36
4	23	84.6	33
5	25	84.6	33

Table 3은 Skip frame값의 변화에 따라 알고리즘이 수행되는 fps와 정확도를 측정된 표이다. skip frame은 실제로 사람을 검출하는데 모든 frame에서 적용될 경우 속도가

낮아 이를 높이기 위해서 일정 frame을 건너 뛰 도록 구현하였다. 이때, skip하는 frame의 수가 커질수록, fps는 증가 하지만 사람이 빠른 속도로 움직이는 경우 ID매칭에 실패하게 되어 Tracking이 제대로 이루어지지 않는 현상이 있어서 결과적으로 카운팅 정확도가 떨어지는 문제점을 발견하였다. 이를 바탕으로 총 39명의 사람이 이동했을 때 skip frame의 변화를 주어 측정되는 fps와 지나가는 사람에 대한 정확도를 측정 하였다. Table 3에서 확인 할 수 있듯이 2 frame을 skip하는 것이 정확도를 높이고 속도를 높일 수 있는 최적의 값이라는 결과를 얻을 수 있었다.

Fig. 7은 옛지 시스템에서 카운팅 한 정보와 관심 영역 내 사람의 수를 모니터링 할 수 있는 어플리케이션 GUI를 구현한 그림이다. 총 4대의 시스템을 사람이 자주 다니는 전시관에 설치하여 테스트를 진행했다. 영상을 직접적으로 실시간 스트리밍해서 영상을 출력하게 되면 네트워크의 부하가 심하고 버퍼링이 생겨 아래 Refresh버튼을 통해 현재 장면의 이미지만을 캡처해서 가져오도록 하였다. 카운팅 정보와 관심영역 정보는 숫자가 바뀔 때 마다 읽을 수 있도록 실시간으로 구현했다.

V. Conclusions

본 논문에서는 옛지 컴퓨팅 방식의 딥러닝을 활용한 방문객 수 카운팅 시스템을 구축하는 방법을 제안하였다. 실제 박물관, 전시관 내에 실증을 통해 사람들이 얼마나 방문

을 했는지, 주로 많이 이동하는 곳이 어디인지를 객관적으로 확인이 가능하여 전시물을 교체하거나 공간을 재배치할 때 중요한 정량적 자료가 될 것으로 기대 한다. 국내의 한 전시관에 설치될 예정이며 전시물에 따른 방문객 카운팅 데이터를 분석 할 예정이다. 더 나아가 방문객의 머리 위치, 자세 등의 정보를 인식하여 실제로 전시물에 얼마나 집중해서 보았는지에 대한 추가 연구를 진행 할 예정이다.

ACKNOWLEDGEMENT

This research is supported by Ministry of Culture, Sports and Tourism(MCST) and Korea Creative Content Agency(KOCCA) in the Culture Technology(CT) Research & Development Program 2020.

REFERENCES

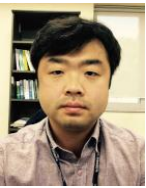
- [1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." *Advances in neural information processing systems* 25 (2012): 1097-1105.
- [2] Brasó, Guillem, and Laura Leal-Taixé. "Learning a neural solver for multiple object tracking." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- [3] Yang, Xiuzhu, et al. "Dense people counting using IR-UWB radar with a hybrid feature extraction method." *IEEE Geoscience and Remote Sensing Letters* 16.1 (2018): 30-34.
- [4] Choi, Jeong Woo, Xuanjun Quan, and Sung Ho Cho. "Bi-directional passing people counting system based on IR-UWB radar sensors." *IEEE Internet of Things Journal* 5.2 (2017): 512-522.
- [5] Yu, Wei, et al. "A survey on the edge computing for the Internet of Things." *IEEE access* 6 (2017): 6900-6919.
- [6] Wikipedia contributors, Nvidia Jetson, https://en.wikipedia.org/w/index.php?title=Nvidia_Jetson&oldid=1021671065
- [7] Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009.
- [8] Girshick, Ross, et al. "Region-based convolutional networks for accurate object detection and segmentation." *IEEE transactions on pattern analysis and machine intelligence* 38.1 (2015): 142-158.
- [9] Girshick, Ross. "Fast r-cnn." *Proceedings of the IEEE international conference on computer vision*. 2015.
- [10] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- [11] Bochkovskiy, Alexey, Chien-Yao Wang, and Hong-Yuan Mark Liao. "Yolov4: Optimal speed and accuracy of object detection." *arXiv preprint arXiv:2004.10934* (2020).
- [12] Balaji, S. R., and S. Karthikeyan. "A survey on moving object tracking using image processing." *2017 11th international conference on intelligent systems and control (ISCO)*. IEEE, 2017.
- [13] Collins, Robert T. "Mean-shift blob tracking through scale space." *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.. Vol. 2*. IEEE, 2003.
- [14] Andrew Banks, Ed Briggs, Ken Borgendale, and Rahul Gupta, MQTT Version 5.0., <https://docs.oasis-open.org/mqtt/mqtt/v5.0/os/mqtt-v5.0-os.html>
- [15] Zhang, Yifu, et al. "FairMOT: On the Fairness of Detection and Re-Identification in Multiple Object Tracking." *arXiv preprint arXiv:2004.01888* (2020).
- [16] Leal-Taixé, Laura, et al. "Motchallenge 2015: Towards a benchmark for multi-target tracking." *arXiv preprint arXiv:1504.01942* (2015).
- [17] Yu, Fisher, et al. "Deep layer aggregation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
- [18] Hu, Yun Chao, et al. "Mobile edge computing—A key technology towards 5G." *ETSI white paper 11.11* (2015): 1-16.
- [19] Mittal, Sparsh. "A Survey on optimized implementation of deep learning models on the NVIDIA Jetson platform." *Journal of Systems Architecture* 97 (2019): 428-442.
- [20] Ren, Ju, et al. "Distributed and efficient object detection in edge computing: Challenges and solutions." *IEEE Network* 32.6 (2018): 137-143.
- [21] Ran, Xukan, et al. "Deepdecision: A mobile deep learning framework for edge video analytics." *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018.
- [22] Li, Dawei, et al. "Deepcham: Collaborative edge-mediated adaptive deep learning for mobile object recognition." *2016 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2016.
- [23] Nikouei, Seyed Yahya, et al. "Smart surveillance as an edge network service: From harr-cascade, svm to a lightweight cnn." *2018 IEEE 4th international conference on collaboration and internet computing (cic)*. IEEE, 2018.

Authors



Jung-Jun Kim received the B.E., and M.E. degrees in Computer Engineering from KyungHee University, Korea, in 2015, 2017 respectively. Jung-Jun Kim joined the researcher of Korea Institute of Robotics &

Technology Convergence, Pohang, Korea in 2017. He is currently a Researcher in AI Robotics Center. He is interested in Computer Vision, Artificial Intelligent.



Min-Gyu Kim received the B.E. in Electronics Engineering and M.E. in Mechanical Engineering from Korea Aerospace University, South Korea in 2003 and 2005 respectively. He holds his PhD

degree in Intelligent Interaction Technologies from University of Tsukuba, Japan in 2012. His research mainly focuses on User-centered Engineering, in particular, System and User modeling in Human-Robot Interaction and Human Factors in Social Robotics.



Ju-Hyun Kim received the B.E., and M.E. degrees in Robotics Engineering from Tongmyong University, Korea, in 2012, 2014 respectively. Ju-Hyun Kim joined the researcher of Korea Institute of Robotics &

Technology Convergence, Pohang, Korea in 2015. He is currently an Senior Researcher in Interactive Robotics Research Division. He is interested in Robotics and Computer vision, Artificial Intelligent.



Man-Gi Lee received the B.E., and M.E. degrees in Mechanical Engineering from Yeungnam University, Korea, in 2014, 2016 respectively. Man-Gi Lee joined the researcher of Korea Institute of Robotics &

Technology Convergence, Pohang, Korea in 2017. He is currently an Senior Researcher in Autonomous Systems Center. He is interested in Robot design and analysis.



Da-Young Kim receive the B.E degree in Mechanical Engineering from Silla University, Korea in 2021. Da-Young Kim joined the researcher of Korea Institute of Robotics & Technology Convergence, Pohang, Korea in

2021. She is currently a Researcher in Interactive Robotics Research Division. She is interested in Robotics and Computer vision, Artificial Intelligent.