

## Multiple Camera-Based Real-Time Long Queue Vision Algorithm for Public Safety and Efficiency

Tae-hoon Kim\*, Ji-young Na\*, Ji-won Yoon\*, Se-Hun Lee\*, Jun-ho Ahn\*\*

\*Student, Dept. of Computer Science and Engineering, Kyonggi University, Kyonggi, Korea

\*\*Professor, Dept. of Computer Science and Engineering, Kyonggi University, Kyonggi, Korea

### [Abstract]

This paper proposes a system to efficiently manage delays caused by unmanaged and congested queues in crowded environments. Such queues not only cause inconvenience but also pose safety risks. Existing systems, relying on single-camera feeds, are inadequate for complex scenarios requiring multiple cameras. To address this, we developed a multi-vision long queue detection system that integrates multiple vision algorithms to accurately detect various types of queues. The algorithm processes real-time video data from multiple cameras, stitching overlapping segments into a single panoramic image. By combining object detection, tracking, and position variation analysis, the system recognizes long queues in crowded environments. The algorithm was validated with 96% accuracy and a 92% F1-score across diverse settings.

▶ **Key words:** Multiple Camera, Queue Recognition, Real-time Vision, Tracking, Image Stitching

### [요약]

본 논문은 대기 인원이 많은 혼잡한 환경에서 대기 시간이 지체되어 관리되지 않는 상황을 효율적으로 관리하는 시스템을 제안한다. 혼잡하고 긴 대기 줄은 불편하고 안전사고를 유발할 수 있다. 기존의 시스템은 단순한 하나의 영상 기반으로 대기 줄을 관리했지만, 혼잡한 상황에 다수의 카메라를 통해서 관리해야 하는 복잡한 상황에서는 적용이 어렵다. 이러한 상황에서 효율적으로 다수의 카메라로 탐지된 하나의 줄을 관리하기 위해 다수의 비전 알고리즘을 융합하여 여러 형태의 대기 줄을 정확하게 인식하는 효율적인 멀티비전 긴 대기 줄 탐지 시스템을 개발하였다. 이 줄 인식 융합 알고리즘은 다수의 카메라의 실시간 영상 데이터를 활용하여 중첩된 부분을 이어 붙여 하나의 실시간 파노라마 영상 이미지로 가공한다. 이러한 영상 데이터를 바탕으로 비전 객체 탐지, 객체 추적, 이미지 스티칭, 각도, 간격, 위치 변화량을 융합해 Queue Recognition 알고리즘을 개발하여, 많은 군중 속에서 다양한 형태의 긴 줄을 인식한다. 본 연구는 다양한 환경에서 실시간 대기 다수의 카메라로 인식된 긴 줄을 탐지하는 융합 알고리즘을 통해서 정확도 96%와 F1-score 92%로 높은 성능을 검증하였다.

▶ **주제어:** 다중 카메라, 줄 인식, 실시간 비전, 추적, 이미지 스티칭

- First Author: Tae-hoon Kim, Corresponding Author: Jun-ho Ahn
- \*Tae-hoon Kim (201910538@kyonggi.ac.kr), Dept. of Computer Science and Engineering, Kyonggi University
- \*Ji-young Na (jiyoung@kyonggi.ac.kr), Dept. of Computer Science and Engineering, Kyonggi University
- \*Ji-won Yoon (kkuriyon5007@gmail.com), Dept. of Computer Science and Engineering, Kyonggi University
- \*Se-Hun Lee (redkye2@kyonggi.ac.kr), Dept. of Computer Science and Engineering, Kyonggi University
- \*\*Jun-ho Ahn (jha@kyonggi.ac.kr), Dept. of Computer Science and Engineering, Kyonggi University
- Received: 2024. 07. 19, Revised: 2024. 09. 30, Accepted: 2024. 09. 30.

## I. Introduction

기술의 발달로 인해 줄서기 시스템은 점차 진화하고 있으며, 모바일 예약 시스템과 SNS를 통한 대기 시간 알림 등 이용자의 대기 시간을 줄이기 위한 다양한 대안들이 등장하고 있다. 그러나 여전히 많은 사람이 몰려 직접 줄을 서야 하는 경우 대기할지와 어느 줄이 가장 효율적인지 고민하게 된다. 본 알고리즘은 이러한 문제를 해결하고자 설계되었다.

최근 코로나19 팬데믹이 완화되면서 여행객들이 다시 공항을 찾기 시작했으며, 2024년 1월 인천공항의 하루 이용객 수는 약 20만 명에 달했다[1]. 이는 코로나 이전 수준을 거의 회복한 수치이다. 그러나 인천공항의 출국 소요 시간은 코로나 전보다 증가한 것으로 나타났다[2]. 공항 측은 보안 검색 요원의 부족과 보안 검색 절차의 강화가 원인이라고 분석하고 있다[3]. 인천공항 내에는 여러 보안 검색대가 있는데, 본 알고리즘을 통해 이용자는 가장 짧은 줄의 보안검색대를 실시간으로 파악할 수 있다. 이를 통해 대기 시간을 줄이고, 이용객의 편의를 증대시킬 수 있을 것이다. 또한, 2022년부터 시행된 광역버스 입석 금지제도는 출퇴근 시간대 버스 이용자에게 큰 불편을 끼치고 있다. 좌석이 없는 경우 서서 이용할 수 없으므로, 다음 버스를 기다려야 하는 상황이 빈번하다. 이에 따라 만석인 버스를 몇 대나 놓치는 일이 자주 발생한다[4]. 본 알고리즘은 버스에 탑승한 사람의 수와 버스를 대기하고 있는 사람의 수를 계산하여, 이용자가 언제 버스에 탑승할 수 있는지 예측할 수 있게 해준다. 또한 다른 교통수단을 이용할 수 있는 사용자는 대기 시간을 확인하여 대체 교통수단을 선택할 수 있어 전체 광역버스 이용자 수를 줄이는 데 이바지할 수 있다. 놀이공원 역시 긴 대기 줄이 가장 먼저 떠오르는 문제 중 하나이다[5]. 기존 방식에서 놀이기구의 대략적인 대기 시간을 제공하지만, 이는 정확하지 않은 경우가 많다. 본 알고리즘은 놀이기구 대기 줄의 인원수와 놀이기구의 진행 시간을 계산하여 더욱 정확한 대기 시간을 이용자에게 알려줄 수 있다. 이를 통해 사람들이 빨리 이용할 수 있는 놀이기구로 분산되어 놀이공원 내 복잡도를 줄일 수 있을 것이다.

본 알고리즘은 공항, 광역버스, 놀이공원 등 다양한 대기 상황에서 실시간으로 효율적인 줄을 선택할 수 있게 하고, 대기 시간을 줄이는 데 도움을 줄 수 있다. 이는 이용자의 편의를 높이고, 시스템의 효율성을 증대시키는 중요한 역할을 할 것이다. 논문 2장에서는 대기 줄 인식 관련하여 선행 연구된 사례와 객체 탐지, Tracking, Image Stitching의

단일 알고리즘에 대한 한계점을 서술하였다. 3장에서는 객체 탐지 알고리즘, Tracking 알고리즘, Image Stitching 알고리즘, 자체 개발 Queue Recognition 알고리즘, 최종 융합 알고리즘을 설명하였다. 4장에서는 실험과 그 결과에 대해 상세히 서술하였다. 마지막으로, 5장에서는 본 연구의 결론을 논의하고 향후 연구 방향에 대해 제시하였다.

## II. Preliminaries

### 1. Queue Recognition Related Pilot Study

관련 선행연구는 빈 장면을 촬영한 후 변화를 탐지하여 사람을 인식하고, 이미지 필터링 기법을 사용해 사람의 수를 탐지했다[6]. 하나 또는 그 이상의 카메라로부터 얻은 이미지를 분석하여 Queue Module(QM)은 대기 줄에서 기다리고 있는 사람 수를 추정하는 역할을 하고, Counter Module(CM)은 서비스 계산대에서 각 사람이 서비스를 완료하기 위해 기다리는 시간을 추정하는 역할을 한다. QM은 군중 밀도 추정 알고리즘과 사람 검출 및 위치 파악 알고리즘을 사용하여 대기 줄의 사람 수를 찾고, CM은 사람 감지 및 위치 파악 알고리즘을 통해 각 사람이 서비스 완료를 위해 기다리는 시간을 계산한다. 그러나 이러한 과정을 알고리즘에 포함하면 연산 과정이 늘어나 실시간으로 군중과 사람을 탐지하는 데 어려움이 있을 수 있다. 또한 본 연구에서는 줄을 탐지하기 위해 영상을 수집하는 방법으로 상황별 단일 카메라 각도를 사용해 하나의 각도에서 사전에 통제된 줄의 형태를 탐지한다. 이는 여러 변수를 갖는 상황에서 다양한 형태와 규모의 줄을 탐지하는 데 있어 한계를 갖는다.

다른 연구에서는 모바일 기기를 이용한 데이터 수집을 통해 얼마나 대기해야 하는지를 지도에서 확인할 수 있게 한다[7]. 이 연구는 CrowdQTE라는 군중 감지 기반 시스템을 제안하며 여러 관심 지점(POI)에 대한 실시간 대기 시간을 모니터링하고 제공한다. CrowdQTE는 기획적 센싱 모드와 참여형 센싱 모드 두 가지를 사용한다. 가속도계 센서를 사용하는 기획적 센싱 모드는 모바일 기기의 가속도계 센서에 데이터를 통해 대기 줄에서 있는 상태인 정지-이동-정지-이동 패턴을 탐지하여 사람들이 줄을 서 있다고 판단한다. 대기 줄에서 있는 사람이 직접 모바일 기기를 통해 상태를 알리는 참여형 센싱 모드는 대기 줄에서 있는 사람이 “대기 시작” 버튼과 “대기 종료” 버튼을 누르면 대기 시간을 기록하고 이를 서버에 업로드한다. 그러나 본 연구는 실시간으로 영상을 통해 줄을 정밀하게 탐지하고 줄

의 형태나 규모를 파악하는 비전 알고리즘에 두는 것이 아닌, 줄을 파악한 데이터를 모바일 서버와 기기에 통신하는 전반적인 시스템에 연구를 포괄적으로 집중했다는 한계점을 갖는다. 또한 이러한 모바일 기기를 통해 데이터를 얻어 분석하려면 사용자의 동의가 필요하며, 대기 줄에 대한 구체적인 정보를 제공하는 데에 있어 연구에 난관이 있다.

## 2. Object Detection, Tracking Related Pilot Study

객체 탐지 방식은 single-stage 방식과 multi-stage 방식으로 나눌 수 있다[8]. multi-stage 탐지기는 정확도가 높다는 장점이 있지만 이미지 전체를 확인하는 방식이기 때문에 속도가 느리다. 이 방식은 주로 Region proposal 방식과 Region classification 방식을 사용하며, Region Proposal 방식을 개선하며 성능을 높였다. single-stage 탐지기는 하나의 네트워크를 사용하며 multi-stage 방식의 두 가지 단계를 하나로 줄여 연산 속도를 높일 수 있다는 이점이 있다[9]. 또한 실시간 객체 추적 알고리즘 중 하나인 SORT는 Hungarian 알고리즘과 Kalman Filter를 사용해 객체를 추적한다[10]. Hungarian 알고리즘은 문제 발생 시 최적의 매칭 값을 찾아 객체와 객체의 ID를 연결하고, 매칭되지 않은 객체들은 제외해준다. 하지만 이러한 방식에서 폐색이 발생하는 경우 객체의 ID를 잘 따라가지 못하는 단점이 있다.

이러한 한계점을 극복하기 위해 개발된 본 알고리즘은 다중 카메라 영상을 Image Stitching 하여 하나의 영상으로 합친 후, single-stage 기반의 객체 탐지 알고리즘인 YOLO와 객체 폐색에 따른 ID-Switching 문제를 개선한 알고리즘인 ByteTrack을 활용하여 사람의 얼굴을 실시간으로 정확하게 탐지하고 추적한다. 이를 통해 대기 시간을 효율적으로 관리하고, 공항, 광역버스, 놀이공원 등 다양한 대기 상황에서 사용자의 편의를 증대시키며 시스템의 효율성을 높일 수 있다.

## III. The Proposed Scheme

### 1. Overview of Research Process

본 장에서는 군중 속에서 줄을 서 있는 사람들을 효과적으로 식별하기 위한 종합적 프레임워크를 소개한다. 이 시스템은 여러 딥러닝 알고리즘과 연구를 위해 직접 설계한 알고리즘들을 통합하여 각 알고리즘이 중요한 역할을 담당한다.

먼저, YOLOV9 객체 탐지 알고리즘을 다룬다. GELAN 네트워크 아키텍처와 Programmable Gradient

Information (PGI)을 도입해 심층 신경망 처리 중에 발생하는 정보 손실 문제를 해결한다. 이를 통해 객체 탐지의 속도와 정확성을 높여 다양한 각도에서 인간의 얼굴을 인식할 수 있다. 다음으로, ByteTrack 추적 알고리즘을 설명한다. 이 알고리즘은 BYTE 연관 방식을 통해 다중 객체 추적을 향상한다. 탐지된 객체의 위치 정보를 일관성 있게 유지하며, 노이즈가 많은 환경에서도 정확한 추적이 가능하다. 이어서, 영상 스티칭 알고리즘에 대해 다룬다. OpenCV를 활용하여 여러 대의 카메라로부터 얻은 비디오 스트림을 하나의 파노라마 이미지로 결합한다. 이를 통해 여러 카메라 시야에 걸친 긴 대기 줄을 효과적으로 모니터링할 수 있다. 그리고, 직접 설계한 Queue Recognition 알고리즘은 군중 속에서 줄을 서 있는 사람들과 자유롭게 움직이는 사람들을 구분하는 데 중점을 둔다. 실시간 데이터에 기반해 객체 간의 간격과 움직임을 동적으로 조정하며, 이를 통해 줄을 정확히 식별할 수 있다. 마지막으로, 융합 AI 알고리즘을 소개한다. 앞서 언급된 알고리즘들의 출력을 종합해 실제 세계에서 줄을 서 있는 사람들을 효과적으로 식별한다. 객체 탐지, 추적, 인식 데이터를 통합해 군중 속에서 줄을 서고 있는 사람들을 신뢰성 있게 모니터링할 수 있다.

이어지는 내용에서는 각 구성 요소를 자세히 설명하며, 이들이 어떻게 융합되어 상호작용해 동적 환경에서 줄을 높은 정확도로 인식하는지 다룬다.

### 2. Object Detection Algorithm

YOLO는 'You Only Look Once'의 줄임말로, 객체 탐지를 수행하는 딥러닝 모델이다. 이 모델은 단 한 번의 신경망 실행으로 이미지 내 객체들을 탐지하는 특징을 갖고 있다. 하지만 기존 YOLO 모델들은 입력 데이터가 총별로 심층 신경망을 통과하면서 많은 정보 손실이 발생한다는 단점이 있다[11]. 이러한 정보 손실 문제는 모델의 성능 저하를 초래하며, 정확한 객체 탐지를 위해 해결이 필요한 중요한 문제이다. 이 문제를 해결하는 YOLOV9 버전은 네트워크 아키텍처인 'GELAN'과 새로운 개념인 'Programmable Gradient Information (PGI)'가 도입되었다. PGI는 추론 프로세스에서 주 경로(main branch)만 사용하기 때문에 추가적인 추론 비용을 발생시키지 않는다. 이는 PGI가 효율적인 방식으로 설계되었음을 의미하며, 모델의 추론 속도와 성능을 동시에 향상시킨다[12]. PGI는 가역 아키텍처(reversible architecture)를 기반으로 하는데, 추가적인 주 경로가 아닌 보조 가역 경로(auxiliary reversible branch)에 통합하여 비용을 절감한다. 이 구조는 역전파 과정에서 중요한 역할을 하며, 주

경로에서 발생할 수 있는 정보 손실을 보완한다. 또한, GELAN 네트워크는 CSPNet과 ELAN을 융합하여 효율성과 표현력을 강화한 객체 탐지 네트워크를 제공한다. CSPNet은 신경망의 효율성과 정보 흐름을 개선하고, ELAN은 다양한 계층의 특징을 효율적으로 결합한다[13].

2.1장에서 언급한 이전 연구에서는 사람 검출 및 위치 추정을 위해 주로 배경 변화 감지[14,15]와 마르코프 체인 몬테카를로(MCMC) 기반[16]의 탐색 최적화 방법을 사용하였다[6]. 이러한 방법들은 사람 실루엣을 검출하고, 이를 기반으로 사람 수를 추정하는 데 중점을 두었으나, 가림 현상이나 복잡한 배경에서 성능이 저하되는 한계를 가지고 있었다.

본 연구에서는 YOLOv9을 사용하여 사람 검출과 위치 추정의 정확도를 크게 개선하였다. YOLOv9의 PGI와 GELAN 네트워크 아키텍처는 기존 방법론이 가지고 있던 정보 손실 문제를 효과적으로 해결하여 더 높은 정확도를 제공한다. 아울러, YOLOV9-e 사전 학습 모델을 사용하여 직접 수집한 인간 얼굴 이미지 데이터셋을 학습시켰다. 데이터 전처리 과정에서 인간의 얼굴을 식별하고 'Front'와 'Back' 두 클래스로 분류하여 얼굴의 전면과 후면을 라벨링 했다. 라벨링이 완료된 데이터셋을 학습하여 파인튜닝된 모델은 다수의 인물이 등장하는 영상 데이터에서 얼굴의 전면과 후면을 인식하고 구분하는 성능 결과를 보여준다. 이는 영상 이미지 속 사람들이 향하는 얼굴 시선을 인식해 줄 인식에 필요한 데이터로 활용된다.

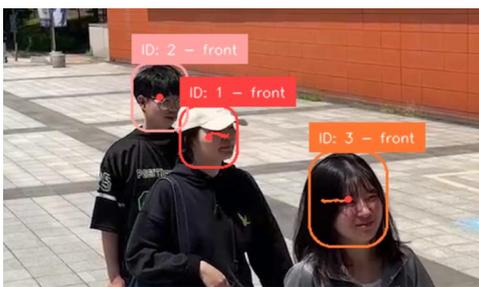


Fig. 1. Object Detection Using YOLOV9

### 3. Tracking Algorithm

ByteTrack은 YOLOX에 "BYTE" association 방식을 통합하여 향상된 다중 객체 추적(MOT) 알고리즘이다. 이 알고리즘은 객체의 연속된 위치 정보를 의미하는 'tracklet'을 활용하여 정확한 추적을 실현한다. 기존의 임계값을 기준으로 탐지된 객체들을 지우는 방식에서 벗어나, ByteTrack은 BYTE라는 새로운 기준을 사용하여 추적을 시도한다[17].

BYTE 방식은 먼저 높은 점수를 받은 객체탐지 영역을 tracklet에 연관시킨다. 이때 객체 탐지 영역 프레임 내에서 객체의 위치를 나타내는 사각형 영역을 의미한다. 높은 신뢰도를 가진 객체 탐지 영역부터 시작하여, 이를 기존의 tracklet에 맞추는 방식으로 이루어진다. 그러나 일부 tracklet은 노이즈나 기타 요인들로 인해 객체 탐지 영역과 맞지 않게 될 수 있다. 이 경우 ByteTrack은 탐지율이 낮은 객체 탐지 영역과 맞지 않게 된 tracklet을 연관 지어 연산을 수행한다. 이렇게 함으로써 노이즈에 강하고 보다 정확한 추적이 가능해진다. ByteTrack의 핵심은 높은 점수를 받은 탐지 영역을 먼저 고려하고, 그 후에 낮은 점수의 탐지 영역을 이용해 추가적인 연산을 수행하는 것이다. 이 접근 방식은 기존의 임계값 기반 방법과 비교했을 때 더 높은 정확도를 제공하며, 다양한 상황에서 객체 추적의 신뢰성을 높이는 데 기여한다[18].

2.1장에서 언급한 이전 연구에서는 주로 배경 변화 감지와 사람의 위치를 추적하기 위해 기하학적 모델과 통계적 방법[14,15]을 활용하였다[6]. 이러한 방법들은 고정된 환경에서 일정 수준의 추적 성능을 보였지만, 복잡한 군중 속에서 정확한 객체 추적이 어려웠고, 객체 간의 가림이나 빠른 움직임에 취약한 한계가 있었다.

이러한 한계를 극복하기 위해, 본 연구에서는 ByteTrack을 도입하여 추적 성능을 크게 개선하였다. ByteTrack은 높은 신뢰도를 가진 탐지 영역을 우선으로 처리한 후, 낮은 점수의 탐지 영역을 이용해 추가 연산을 수행함으로써 더욱 정확하고 안정적인 객체 추적을 가능하게 한다. 나아가, ByteTrack을 사용하여 각 객체의 ID를 식별하고, 객체의 위치 좌표가 움직이는 경로를 프레임 별로 곡선으로 표현하여 이동 방향과 거리를 시각화하였다. 이 기술을 통해 군중 속에서 줄을 서지 않고 자유롭게 움직이는 행인들의 위치 변화를 정확히 추적하였다. 이 데이터는 줄을 서는 사람들과 그렇지 않은 사람들을 구분하는 데 필요한 중요한 근거로 활용된다.

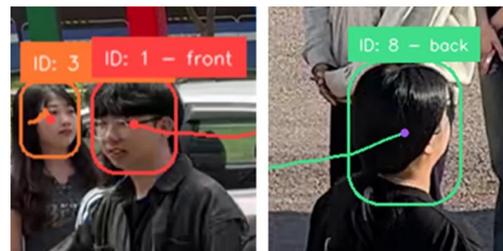


Fig. 2. Object Tracking Using ByteTrack

#### 4. Image Stitching Algorithm

실시간 영상 스티칭은 여러 대의 카메라로부터 얻은 영상들을 하나의 연속적인 이미지로 합치기 위해 OpenCV의 이미지 스티칭 알고리즘을 사용하여 다중 실시간 영상을 하나의 통합된 이미지 영상으로 가공한다[19]. 제안하는 알고리즘은 OpenCV의 기존 이미지 스티칭 알고리즘을 사용하여 실시간 처리 성능과 스티칭 품질을 최적화한다.

2.1장에서 언급한 이전 연구에서는 여러 대의 카메라를 사용해 대기열을 모니터링하고, 그 결과를 분석하는 방식으로 시야 및 커버리지를 개선하려고 하였다[6]. 그러나 하나의 장면을 여러 카메라가 다른 각도에서 관찰하는 것이 아닌, 상황별 하나의 각도를 갖는 단일 카메라로 모니터링이 진행되어 단일 카메라 간의 비중첩된 시야와 제한된 시야각으로 인해 일부 영역에서 감시의 사각지대가 발생할 수 있다는 한계가 있었다.

이러한 한계를 극복하기 위해, 본 연구에서는 OpenCV의 이미지 스티칭 알고리즘을 도입하여 다중 카메라로부터 얻은 영상을 하나의 연속적인 이미지로 통합하였다. OpenCV의 이미지 스티칭 알고리즘은 SIFT를 사용하여 각 영상에서 특징점을 추출하고 매칭한다. SIFT 알고리즘은 회전과 스케일 변환에 불변인 특징점을 찾아내어 다양한 조건에서도 안정적인 스티칭을 가능하게 한다[20]. 이후 RANSAC 알고리즘을 이용해 특징점 매칭 후 오정합을 제거하고 매칭의 신뢰성을 높인다. RANSAC은 노이즈와 외곽 점을 효과적으로 제거하여 정합 모델의 정확도를 높이며, 불필요한 오정합을 최소화하여 안정적인 스티칭 결과를 얻는 데 중요한 역할을 한다[21]. Homography 계산을 통해 영상 간의 기하학적 변환을 수행하여 각 영상의 위치를 정확히 변환함으로써 실시간 처리 성능을 강화하고 자연스러운 영상 스티칭을 가능하게 한다[22]. OpenCV의 멀티밴드 블렌딩 기법을 적용하여 경계부의 부자연스러운 부분을 최소화한다. 멀티밴드 블렌딩은 여러 해상도의 피라미드를 사용하여 경계부의 부자연스러움을 줄이고 자연스러운 스티칭 결과를 제공한다[23].

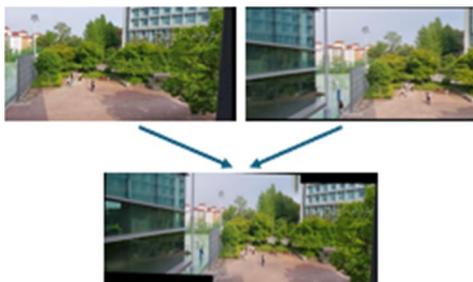


Fig. 3. Image Stitching Using OpenCV

#### 5. Queue Recognition Algorithm

본 연구에서 설계하고 개발한 Queue Recognition 알고리즘의 주요 목적은 군중 속에서 줄을 이루는 객체들을 정확하게 식별하고, 움직이는 행인을 줄로 잘못 인식하지 않는 것이다. 이를 위해 알고리즘은 객체 간의 간격과 위치 변화량을 기반으로 두 가지 임계값을 실시간으로 환경에 맞게 자동으로 업데이트하는 방식을 채택하고 있다. 이 두 임계값은 조건문을 통해 최종적으로 줄을 이루는 객체들을 식별하는 데 사용된다. 객체 간의 간격과 위치 변화량을 명시하기 위한 단위는 입력 영상 이미지의 1920\*1080프레임의 각 픽셀이 기준 단위이다.

알고리즘의 첫 번째 단계는 두 객체 간의 간격을 유클리드 거리로 계산하는 것이다[24]. 영상의 각 프레임마다 거리 리스트를 생성한 후, 이를 오름차순으로 정렬한다. 이 리스트의 길이를 특정 분할 변수인 'dist\_divider'로 나눈 값을 'MinDistNum'으로 정의한다. 'MinDistNum'은 거리 리스트에서 선택할 거리의 개수를 결정하며, 이때 선택된 가장 작은 'MinDistNum'개의 거리 값의 평균을 계산하여 'dist\_weight'라는 가중치를 적용한다. 이렇게 구한 최종 거리 임계값은 환경에 맞게 업데이트된다. 만약 'MinDistNum'이 0 이하인 경우, 거리 임계값은 0으로 설정된다. 이는 선택된 거리 개수가 없거나 매우 적을 때 발생할 수 있는 상황이다. 이 과정을 통해 도출된 변수는 'avg\_smallest\_MinDistNum'이며, 이 변수는 줄로 인식될 수 있는 사람들 간의 최소 줄 간격을 판별하는 기준의 임계값으로 사용된다. 이에 대한 수식은 다음과 같으며, 사전의 정의되어야 하는 'dist\_weight' 가중치 값은 1.4, 'bytrack\_weight' 가중치 값은 1.4, 'dist\_divider' 변수 값은 5로 사용되어 가장 적합한 성능이 나오는 파라미터를 산출하였다.

$$avg\_smallest\_MinDistNum = w_d * \frac{\sum_{i=1}^M distance_i}{M}$$

두 번째 단계에서는 ByteTrack을 사용해 객체의 현 프레임과 직전 프레임 간의 위치 좌표 변화량을 계산한다. ByteTrack의 거리 데이터를 내림차순으로 정렬한 후, 'num\_distances\_to\_average'로 정의된 특정 개수의 거리 값을 선택하여 그 평균을 계산한다. 이 평균값에 'bytrack\_weight'라는 가중치를 적용하여 최종 변화량 임계값을 업데이트한다. 만약 계산된 평균값이 1 이하인 경우, 변화량 임계값을 3으로 설정해 너무 낮은 값을 보정한다. 이 과정을 통해 얻은 변수는 'mean\_byte\_track\_dist'이며, 이 변수는 행인이 줄로 인식되지 않기 위해 위치가

변하여 지나가는 사람으로 판별될 수 있는 기준 임계값으로 사용된다. 이에 대한 수식은 다음과 같으며, 사전의 정의되어야 하는 'num\_distances\_to\_average' 변수값은 5로 사용되어 가장 적합한 성능이 나오는 파라미터를 산출하였다.

$$\lambda = \begin{cases} w_b * \frac{\sum_{i=1}^K \text{byte\_track\_dist}_{s_i}}{K} & \text{if } w_b * \frac{\sum_{i=1}^K \text{byte\_track\_dist}_{s_i}}{K} > 1 \\ 3 & \text{otherwise} \end{cases}$$

$\lambda = \text{mean\_byte\_track\_dist}$

이 연구에서는 이 두 가지 임계값을 활용해 객체들이 줄을 이루고 있는지를 판별하는 조건문을 구현했다. 객체 간의 간격이 거리 임계값보다 작고, 객체의 위치 변화량 변화량 임계값보다 적을 경우, 알고리즘은 이들이 줄을 서고 있다고 판단한다. 또한, 인식된 객체의 클래스가 'Front'인지 'Back'인지를 구분하며, 동일 클래스의 객체들에 대해서만 조건문을 적용한다. 이 접근법은 줄을 선 사람들이 앞이나 뒤로 유사한 얼굴 방향을 유지한다는 가정에 기반한다. 이러한 방식으로 줄을 서 있는 객체들을 정확하게 식별할 수 있으며, 움직이는 행인들이 줄로 잘못 인식되지 않도록 방지할 수 있다.

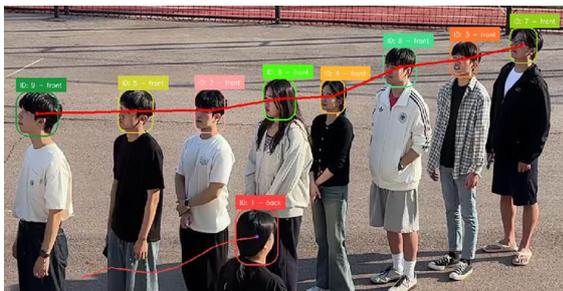


Fig. 4. Line Detection Using Queue Recognition Algorithm

### 6. Fusion AI Algorithm

본 논문에서 제안하는 최종 융합 알고리즘은 다수의 군중 속에서 줄을 서 있는 사람들을 효과적으로 식별하는 데 중점을 둔다. 이 알고리즘은 실시간 영상 데이터의 전처리, 객체 탐지 및 클래스 구분, 추적(Tracking) 및 객체 ID 부여, 줄 인식(Queue Recognition)의 네 가지 주요 단계를 통해 이루어진다.

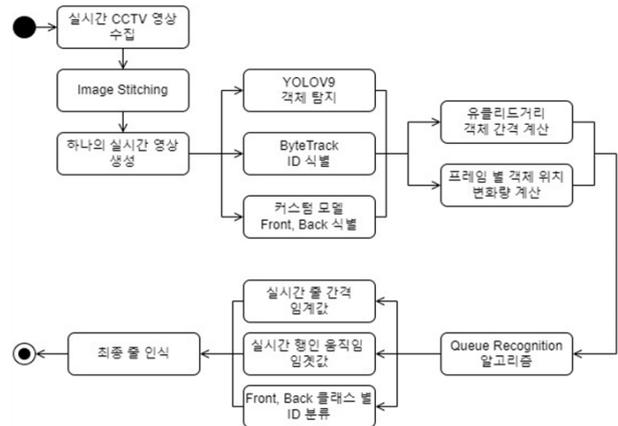


Fig. 5. Fusion Algorithm State Diagram

첫 번째 단계는 실시간 영상 데이터의 전처리이다. 여기서는 다중 입력으로부터 획득된 2개 이상의 실시간 영상 데이터를 하나의 통합된 실시간 영상 데이터로 가공한다. 이 과정을 통해 긴 대기 줄 전체를 하나의 카메라가 탐지하는 데에 발생하는 한계를 극복하여 다수의 카메라 영상을 하나의 실시간 파노라마 영상 이미지로 생성한다. 이에 따라 후속 단계인 객체 탐지 및 추적 알고리즘이 효율적으로 작동할 수 있도록 준비한다. 두 번째 단계는 객체 탐지와 클래스 구분이다. 통합된 실시간 영상 데이터에서 사람의 얼굴을 객체 탐지 알고리즘을 사용하여 식별한다. 탐지된 얼굴 객체는 'Front'와 'Back' 클래스로 구분되며, 딥러닝 기반의 YOLOV9 알고리즘을 활용해 파인튜닝 학습을 거쳐 정확도 높은 커스텀 모델을 생성한다. 세 번째 단계는 Tracking 및 객체 ID 부여이다. 탐지된 얼굴 객체는 ByteTrack 추적 알고리즘을 통해 고유의 ID가 부여되고 프레임 간 일관성을 유지한다. 이를 통해 과거 영상 프레임에서 탐지된 사람과 현재 영상 프레임에서 탐지된 사람이 같은 인물인지 식별할 수 있다. 네 번째 단계는 줄 인식이이다. 객체 탐지 및 추적 알고리즘을 통해 추출된 데이터를 기반으로 줄을 인식한다. 줄을 서 있는 사람들 간의 간격과 움직이는 행인을 구분하는 과정에서 실시간으로 줄 간격과 행인의 움직임 변화량에 대한 임계값을 설정한다. 이 임계값은 가중치와 수식을 사용한 Queue Recognition 알고리즘을 통해 동적으로 산출된다. 탐지된 객체들의 클래스가 같고, 임계값 기준 허용 범위 내의 데이터를 가진 객체들은 줄을 서 있는 사람으로 판별된다. 마지막으로, 이러한 모든 단계를 종합하여 최종적으로 줄을 서 있는 사람들을 식별한다. 이를 위해 다양한 알고리즘의 결과를 융합하고, 실시간으로 데이터의 일관성을 검증한다.

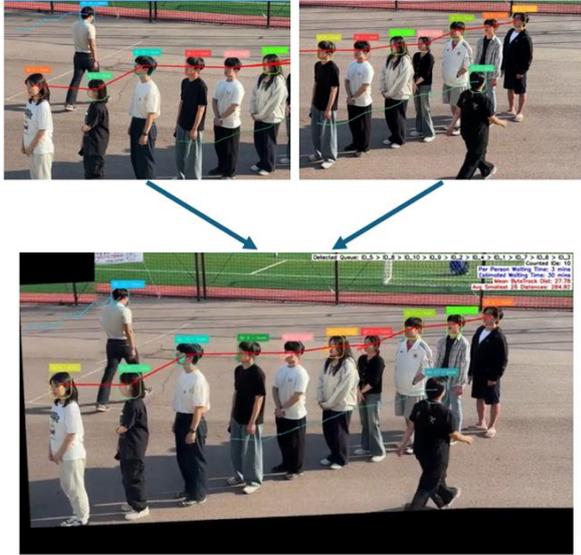


Fig. 6. Line Detection Using Fusion Algorithm

## IV. Experiment

### 1. Experimental Environment and Scheme

본 연구의 실험을 위해 약 5,000장의 사진 이미지와 68개의 영상 이미지를 직접 촬영하고 수집하였다. 데이터의 일관성을 유지하기 위해, 영상 이미지는 MP4 형식의 1080p 해상도로 촬영하였고, 사진 이미지는 JPG 형식의 1080p 해상도로 촬영하였다. 수집된 데이터셋은 Image Stitching, 객체 탐지, ByteTrack, Queue Recognition 알고리즘에 적용하여 사람들의 얼굴 탐지 여부와 줄 인식 여부를 실험하였다.

연구의 특성상 특정 방향으로 줄을 서 있는 사람들의 모습을 담은 영상과 사진을 수집하였으며, 야외와 실내 등 다양한 주변 환경에서 데이터를 확보하였다. 데이터 수집 과정에서 얼굴이 노출되는 사람들에게 사전 동의를 구하고 연구 목적을 상세히 설명함으로써 개인정보 보호 문제를 철저히 관리하였다[25]. 또한, 데이터에 사용되는 사람들의 성별, 옷차림 등 다양한 특징을 고려하여 포괄적인 데이터를 수집하였다.

실험은 세 가지로 분류해 진행되었다. 첫 번째는 최종 융합 알고리즘에 사용된 Image Stitching 기법의 성능을 검증하였다. 이를 위해 다양한 환경에서 촬영된 영상 이미지들을 Image Stitching 적용 여부에 따라 성능 결과를 비교하였다. 두 번째는 융합 알고리즘의 성능을 Confusion Matrix 기법을 통해 평가하였으며, 알고리즘의 Accuracy, Precision, Recall, F1-Score 값들을 계산하여 성능을 종합적으로 검토하였다[26]. 세 번째는 설계

된 다양한 상황에서 줄 간격 임계값만 적용된 알고리즘, ByteTrack 임계값만 적용된 알고리즘, 그리고 융합 알고리즘을 비교 분석함으로써 각 알고리즘이 상황별 줄 인식에 어떻게 반응하는지를 검증하였다.

### 2. Comparison of Results between Single Image and Stitched Image

Image Stitching의 성능 평가를 위해 Image Stitching을 하지 않은 단일 영상 데이터와 2개의 영상 데이터를 Image Stitching 하여 하나의 파노라마 영상 데이터로 가공한 영상 데이터를 대상으로 세 가지의 산출 값을 비교하였다. 첫 번째로 각 영상 데이터에서 탐지되는 사람 얼굴 수를 세어 평균 탐지 성공 인원수를 산출해 비교하였다. 두 번째로 각 영상 데이터에서 탐지되지 못한 사람 얼굴 수를 세어 평균 탐지 실패 인원수를 산출해 비교하였다. 세 번째로 영상 각 영상 데이터에서 등장하는 모든 사람 얼굴 수 중에서 탐지한 얼굴 수에 대한 성공률을 산출해 비교하였다.

정확한 비교를 위해 Image Stitching을 하지 않은 단일 영상 데이터 A와 B에 대한 성능을 평가하고, 데이터 A, B를 Image Stitching 하여 하나의 영상 데이터로 가공해 성능을 평가하였다. 이러한 방법 총 12개의 영상 이미지 데이터를 대상으로 두 개의 단일 영상 데이터를 한 쌍으로 묶어, Image Stitching의 사용 유무에 따른 성능 차이를 비교 분석하였다.

Table 1. The Experimental Result for the Comparison of Object Detection Rates with Image Stitching

	none-stitching single image	image stitching processed images
Num of Detection Success Avg	10.2	10.33
Num of Detection Fail Avg	1.92	1.33
Total Success Percentage	83.93%	88.24%

표1의 결과를 살펴보면 Image Stitching이 적용되지 않은 단일 영상 이미지보다 Image Stitching이 적용된 파노라마 영상 이미지에서 탐지되는 사람 얼굴 수의 평균 수치가 더 높으며, 탐지에 실패한 사람 얼굴 수의 평균 수치는 더 낮은 것을 알 수 있었다. 또한 탐지되어야 하는 총 사람 얼굴 수에서 실제로 얼굴을 탐지한 성공률을 비교했을 때, Image Stitching 가공된 영상의 성공률이 88.24%로 Image Stitching이 사용되지 않은 단일 영상의 성공률인 83.93%와 비교했을 때 높은 성능을 보이는 것을 알 수 있었다.

### 3. Experimental Results of Fusion AI Algorithm

여러 알고리즘을 융합하여 대기 줄 탐지 성능을 평가하는 실험을 진행하였다. 이를 위해 Confusion Matrix 기법을 활용하여 True Positive(TP), True Negative(TN), False Positive(FP), False Negative(FN) 값을 도출하고, 이를 바탕으로 정확도(Accuracy), 정밀도(Precision), 재현율(Recall), F1 Score를 산출하였다[27].

실험 과정은 다음과 같다. 융합 알고리즘을 적용하여 테스트 영상 데이터를 입력받고, 각 영상 프레임별로 탐지 결과를 저장하였다. 이 탐지 결과를 평가하여 네 가지 범주로 분류하였다. 대기 줄을 정확하게 탐지하고 인식된 대기 인원을 명확한 하나의 줄로 인식하는 경우를 True Positive(TP)로, 대기 줄의 인원 중 일부를 누락하여 정확한 하나의 줄로 탐지하지 못한 경우를 True Negative(TN)로, 대기 줄에 포함되지 않은 행인을 대기 인원으로 잘못 포함한 경우를 False Positive(FP)로, 대기 줄을 전혀 인식하지 못한 경우를 False Negative(FN)로 분류하였다. 이처럼 분류된 결과값을 활용하여 산출한 정확도, 정밀도, 재현율, F1 Score는 다음과 같다.

Table 2. The Experimental Result for the Fusion Recognition Algorithm by Features

	Accuracy	Precision	Recall	F1-Score
Fusion Algorithm	0.96	0.89	0.95	0.92

표 2의 수치를 살펴보면, 부분 항목의 값이 0.9 이상으로 산출되어 융합 알고리즘의 높은 신뢰도를 보여준다. 대기 줄 인원들을 정확히 탐지하는 TP와 대기 줄은 인식하지만, 일부 대기 인원 탐지를 누락하는 TN의 수치가 높게 나타났으며, 대기 줄을 전혀 인식하지 못하는 FN 수치는 낮게 측정되었다. 이에 따라 정확도가 0.96, 재현율이 0.95, F1 Score가 0.92로 모두 0.9 이상의 유의미한 값을 나타내어 높은 신뢰도와 성능을 입증하였다. 다만, 테스트 영상 데이터에서 대기 줄이 아닌 사람들을 대기 인원으로서 잘못 인식하는 오차로 인해 FP 수치가 높아졌으며, 이에 따라 정밀도 값이 0.89로 다소 낮게 나타난 점이 아쉬운 결과로 분석되었다. 하지만 0.89의 정밀도 수치도 신뢰성이 저하되는 절대적으로 낮은 수치는 아니기에, 유의한 결과 수치라고 볼 수 있다고 시사한다.

### 4. The Experimental Result for the Fusion Recognition Algorithm in various coverage

본 연구에서 개발한 Queue Recognition 알고리즘의 실시간 임계값 설정 기능을 단일 줄 간격 임계값 알고리즘과 ByteTrack 임계값 알고리즘으로 나눠 최종 융합 알고리즘과 함께 총 3가지의 알고리즘을 평가하고자 수집한 다양한 상황에 따른 실험을 진행하였다.

Table 3. The Experimental Result for the Fusion Recognition Algorithm in various coverage

	only with dist method	only with ByteTrack method	Fusion Algorithm
form Front Queue	○	△	○
/ form Front Queue	○	△	○
\ form Front Queue	○	△	○
\ form Back Queue	○	△	○
- form Left Queue	○	△	○
- form Right Queue	○	△	○
S form Front Queue	X	△	△
> form Front Queue	X	△	△
< form Front Queue	X	△	△
~ form Front Queue	△	△	○
~ form Back Queue	△	△	○

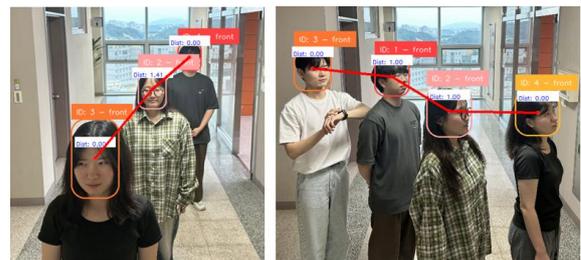


Fig. 7. Experimental results of the 'Front' class in various scenarios of the Fusion algorithm

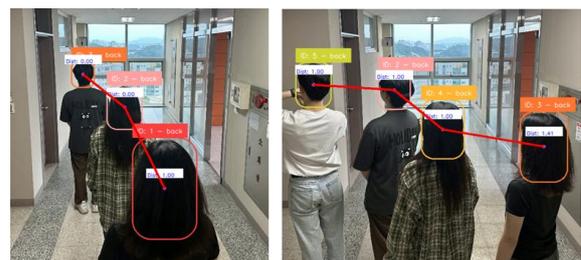


Fig. 8. Experimental results of the 'Back' class in various scenarios of the Fusion algorithm

실험 결과를 정리한 표 3을 살펴보면, 단일 줄 간격 임계값 설정을 탑재한 Queue Recognition 알고리즘은 단순한 형태의 대기 줄은 정확하게 인식하는 반면, S 모양이나 < 모양과 같이 복잡한 형태의 대기 줄은 탐지에 실패하는 것을 알 수 있었다. 다음으로 단일 ByteTrack 임계값 설정을 탑재한 Queue Recognition 알고리즘은 모든 상황에서 대기 줄을 탐지했으나, 인식하는 대기 인원수와 줄 형태에 대한 정확도가 현저히 낮아 아쉬운 성능을 보였다. 마지막으로 최종 융합 알고리즘은 전반적으로 모든 상황에서 대기 줄을 탐지하는 결과를 나타내었다. 특히 단순 형태의 대기 줄은 정확한 대기 인원 파악까지 가능하였으며, S 모양과 < 모양 등 복잡한 형태의 대기 줄은 잠시 특정 영상 프레임에서 대기 인원을 누락하는 오차는 존재했으나, 대체로 대기 줄 형태와 대기 인원을 정확한 결과와 유사하게 탐지하는 유의한 성능을 보이는 것을 확인할 수 있었다.

## 5. Overall Discussion of Experimental Results

앞서 제시한 실험을 통해 2.1장에서 언급한 이전 연구의 알고리즘에서 드러난 한계점들, 즉 실시간으로 다수의 군중을 탐지하는 데 어려움이 있다는 점, 통제된 제한적인 상황에서만 대기 줄을 인식할 수 있다는 문제, 특정 각도의 단일 카메라 시야에서만 인식 가능하다는 점, 그리고 다양한 형태의 유동적인 대기 줄을 효과적으로 탐지하지 못한다는 문제를 본 연구는 보완하고자 하였다. 이러한 한계는 실제 환경에서 다양한 변수에 대응하기 어렵다는 점에서 매우 중요한 개선 과제로 인식되었으며, 이를 해결하기 위해 본 연구는 새로운 접근 방식을 제안하였다.

본 연구에서 제시하는 최종 융합 알고리즘은 다수의 카메라로 촬영된 영상을 중첩하여 넓은 시야의 이미지를 한 번에 입력할 수 있도록 설계되었다. 이를 통해, 특정 각도에만 의존하는 기존 알고리즘의 한계를 극복하고, 카메라가 설치된 환경에서는 다양한 각도와 복잡한 상황에서도 군중 속 대기 줄을 정확하게 인식할 수 있다. 또한, 본 연구는 단순히 정면을 향하는 일직선 형태의 대기 줄만을 탐지하는 것이 아니라, 방향과 관계없이 유동적인 형태의 대기 줄도 효과적으로 탐지할 수 있음을 실험을 통해 검증하였다.

## V. Conclusions

대기 줄을 서는 사람들은 대기 상황이 빨리 해소되지 않을 때 불편함을 느낀다. 이는 클라이언트 관점에서 중요하

게 관리해야 할 부분이다. 이에 관해 본 연구의 알고리즘은 버스나 지하철 대기 시에도 활용할 수 있다. 기존 방식과 달리 비전 기술과 융합하여 사전 요구사항 없이 다수의 카메라 영상만으로 필요한 영상을 생성하고 이를 기반으로 행인의 얼굴을 탐지하여 대기 인원수와 대기 시간을 추출할 수 있다. 또한, 화면에서 어디에 줄을 서 있는지, 어디가 가장 혼잡한지 파악할 수 있다. 하나의 알고리즘만 사용했을 때 줄에 대한 인식률이 떨어지고, 여러 노이즈가 발생하는 것을 실험에서 확인했다. 이러한 문제를 해결하기 위해 YOLO, BYTE TRACK, STITCHING, Queue Recognition 알고리즘을 통합하여 사용했다. 이 통합 접근법을 통해 노이즈를 효과적으로 제거하고 줄을 정확히 인식할 수 있었다. 각 알고리즘은 서로의 약점을 보완하며, 종합적으로 더 높은 인식률과 안정성을 제공함으로써 대기 줄 관리를 위한 솔루션을 제공한다.

이를 통해 2.1장에서 언급한 이전 연구에서 제시한 알고리즘의 한계점으로 분석된 실시간 다수의 군중 탐지, 통제된 제한적인 상황의 대기 줄만 인식 가능, 특정된 각도의 단일 카메라 시야에서만 인식 가능, 여러 형태의 유동적인 대기 줄 상황 인식 불가와 같은 문제점들을 개선할 수 있다. 본 연구에서 제시하는 최종 융합 알고리즘은 다수의 카메라에서 촬영된 영상 이미지를 중첩해 넓은 시야의 이미지를 한 번에 입력할 수 있다. 또한 카메라만 설치되어 있다면 다양한 각도와 상황에서 본 연구의 알고리즘을 활용해 군중 속 대기 줄을 정확하게 인식할 수 있다. 나아가, 단순히 정면을 향하는 일직선 형태의 대기 줄이 아닌, 방향과 무관하게 유동적인 형태의 대기 줄도 파악할 수 있는 것을 실험을 통해 검증하였다.

반면, 본 연구 과정에서 한계점 또한 발견할 수 있었다. 제안된 줄 인식 알고리즘은 긴 형태의 단일 줄을 인식하는데에는 높은 정확도를 보였지만, 인파가 밀집된 환경에서 다양한 곡선 형태의 줄을 이루는 경우, 성능이 다소 저하되는 문제가 있었다. 이는 줄 인식 시 사람들 간의 거리와 변화량을 주로 활용한 방식이 다양한 형태와 환경 변수를 충분히 고려하지 못한 결과로 판단된다. 이러한 한계를 극복하기 위해, 향후 연구에서는 더 복잡한 줄 형태와 환경에서도 높은 정확도를 유지할 수 있는 알고리즘 개선을 목표로 할 것이다. 특히, 다양한 시나리오를 반영한 실험과 검증을 통해 알고리즘의 범용성을 높이고, 더욱 안정적인 줄 인식을 가능하게 하는 방법을 모색할 계획이다.

향후 기대되는 내용은 해당 알고리즘을 사용해 대기 고객들이 사용할 수 있는 서비스 플랫폼을 구축하는 것이다. 클라이언트 입장에 혼잡도를 한눈에 파악하고 이를 해소

하는 방안을 앱을 통해 바로 알려줄 수 있다. 사용자는 이러한 정보를 모바일 기기로 확인하고, 적절한 시기에 대기 줄을 서거나 이동하는 결정을 내릴 수 있다. 이는 사용자의 경험을 크게 향상할 수 있을 것이다.

## ACKNOWLEDGEMENT

“This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the National Program for Excellence in SW supervised by the IITP(Institute of Information & communications Technology Planning & Evaluation)”(2021-0-01393)

## REFERENCES

- [1] Policy Research Team, Incheon International Airport Corporation, "Time Series Statistics," <https://www.airport.kr/co/ko/cpr/statistic/CategoryOfTimeSeries.do>.
- [2] Jung Da-min, "Feature: K-pop Concerts as Cultural Exports," The Korea Times, 2024, [https://www.koreatimes.co.kr/www/culture/2024/06/135\\_365480.html](https://www.koreatimes.co.kr/www/culture/2024/06/135_365480.html).
- [3] JC Park, "Incheon Airport Free from COVID-19... But 'Queueing' Has Worsened," *Kyunghyang shinmun*, 2024. <https://www.khan.co.kr/economy/market-trend/article/202401081543001>.
- [4] Soo Kim, "I Want to Ride Too: Commute Shuttles," *Journal of Transportation Technology and Policy*, vol. 20, no. 2, pp. 70-71, 2023, DOI: <https://www.dbpia.co.kr/pdf/pdfView.do?nodeId=NO DE11409532>.
- [5] Hyo-Jin Kim, "Domestic Tourism Rebounds in Summer," *Hankook Ilbo*, 2023, <https://www.hankookilbo.com/News/Read/A2023072617250003934>.
- [6] Parameswaran, Vasu, Vinay Shet, and Visvanathan Ramesh, "Design and Validation of a System for People Queue Statistics Estimation," *Video Analytics for Business Intelligence*, Berlin, Heidelberg: Springer Berlin Heidelberg, vol. 409, pp. 355-374, 2012, DOI: 10.1007/978-3-642-28598-1\_11.
- [7] Jiangtao Wang, et al., "Real-time and generic queue time estimation based on mobile crowdsensing," *Frontiers of Computer Science*, vol. 11, pp. 49-60, 2017, DOI: <https://doi.org/10.1007/s11704-016-5553-z>.
- [8] Nicolai Wojke, Alex Bewley, and Dietrich Paulus, "Simple online and realtime tracking with a deep association metric," 2017 IEEE International Conference on Image Processing (ICIP), IEEE, 2017, DOI: 10.1109/ICIP.2017.8296962.
- [9] Juan Du, "Understanding of object detection based on CNN family and YOLO," *Journal of Physics: Conference Series*, vol. 1004, IOP Publishing, 2018, DOI: 10.1088/1742-6596/1004/1/012029.
- [10] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing (ICIP), Phoenix, AZ, USA, Sep. Feb 2016. DOI: 10.1109/ICIP.2016.7533003.
- [11] J. Terven, D.-M. Córdova-Esparza, and J.-A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS," *Machine Learning and Knowledge Extraction*, vol. 5, no. 4, pp. 1680-1716, Nov. 2023. DOI: 10.3390/make5040083.
- [12] Ultralytics, "YOLOv9," Ultralytics Documentation. <https://docs.ultralytics.com/ko/models/yolov9/>.
- [13] C.-Y. Wang, I.-H. Yeh, and H.-Y. M. Liao, "YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information," *arXiv preprint arXiv:2402.13616*, Feb 2024. DOI: <https://arxiv.org/abs/2402.13616>.
- [14] V. Parameswaran, M. Singh, and V. Ramesh, "Illumination compensation based change detection using order consistency," 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 1982-1989. DOI: 10.1109/CVPR.2010.5539873.
- [15] M. Singh, V. Parameswaran, and V. Ramesh, "Order consistent change detection via fast statistical significance testing," 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 2008, pp. 1-8. DOI: 10.1109/CVPR.2008.4587668.
- [16] J. S. Speagle, "A Conceptual Introduction to Markov Chain Monte Carlo Methods," *arXiv preprint arXiv:1909.12313*, Sep 2020. DOI: <https://arxiv.org/abs/1909.12313>.
- [17] Zhang, Yifu, et al., "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," GitHub repository, 2021. Available: <https://github.com/ifzhang/ByteTrack>.
- [18] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-Object Tracking by Associating Every Detection Box," *arXiv preprint arXiv:2110.06864*, Apr 2022. DOI: <https://arxiv.org/abs/2110.06864>.
- [19] Nebula4869, "Real-time-Panoramic-Stitching," GitHub. <https://github.com/Nebula4869/Real-time-Panoramic-Stitching>.
- [20] T. Lindeberg, "Scale Invariant Feature Transform," *Scholarpedia*, vol. 7, no. 5, pp. 10491, 2012. DOI: 10.4249/scholarpedia.10491
- [21] M. A. Fischler, and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, Vol. 24, No. 6, pp. 381-395, June 1981. DOI: 10.1145/358669.358692.
- [22] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Deep Image

Homography Estimation," arXiv preprint arXiv:1606.03798, June 2016. DOI: <https://arxiv.org/abs/1606.03798>.

- [23] Dong-Bin Xu, He-Meng Tao, Jing Yu, Chuang-Bai Xiao, "Real-Time Multi-camera Video Stitching Based on Improved Optimal Stitch Line and Multi-resolution Fusion," *Lecture Notes in Computer Science*, pp. 124-133, ICIG, Beijing, China, December 2017, DOI: 10.1007/978-3-319-71598-8\_12.
- [24] Yanfang Zhang, Xinyu Zhang, "Real-time Image Stitching with Improved Seam Cutting and Blending," *2015 IEEE International Conference on Image Processing (ICIP)*, pp. 3509-3513, Quebec City, Canada, Sep 2015. DOI: 10.1109/ICIP.2015.7351332.
- [25] Kailee Kodama Muscente, "Audio and Video Recording in Remote Research," Teachers College, Columbia University. <https://www.tc.columbia.edu/institutional-review-board/irb-blog/2022/audio-and-video-recording-in-remote-research/>.
- [26] M. Vakili, M. Ghamsari, and M. Rezaei, "Performance Analysis and Comparison of Machine and Deep Learning Algorithms for IoT Data Classification," arXiv preprint arXiv:2001.09636, Jan 2020. DOI: <https://arxiv.org/abs/2001.09636>.
- [27] Mihaela Muntean, Florin-Daniel Militaru, "Metrics for Evaluating Classification Algorithms," in *Education, Research and Business Technologies, Smart Innovation, Systems and Technologies*, vol 321, Springer, pp. 307-317, Singapore, Jan 2023. DOI: 10.1007/978-981-19-6755-9\_24.



Ji-won Yoon will receive a Bachelor's degree in Artificial Intelligence from the Department of AI Computer Engineering at Kyonggi University in 2025. Her interests are computer vision, data analysis, and artificial intelligence.



Se-Hun Lee is expected to receive his Bachelor's degree in Artificial Intelligence from the Department of AI Computer Engineering at Kyonggi University in 2025. He plans to pursue a M.S. degree afterward

in Computer Science from the Kyonggi University. He is a research assistance at the VisionAI Lab in the Department of AI Computer Engineering at Kyonggi University. His research interests include computer vision, deep learning, pattern analysis and artificial intelligence.



Jun-ho Ahn received the B.S. degree in Computer Engineering from Hongik University in 2006 and the M.S. degree in Computer Engineering from Yonsei University in 2008. He earned the Ph.D. in Computer Science

from the University of Colorado, Boulder, in 2013. He served as a senior researcher at the National Security Research Institute of ETRI in South Korea until 2017. He was an associate professor at Korea National University of Transportation until 2023, and he is currently an associate professor at Kyonggi University. He is also the supervising professor leading the VisionAI Lab in the Department of AI Computer Engineering at Kyonggi University. His research interests include the development and application of computer vision, deep learning, pattern analysis, artificial intelligence, and real-time detection.

## Authors



Tae-hoon Kim is currently completing his B.S. of Science in Artificial Intelligence at the Department of AI Computer Engineering, Kyonggi University, with graduation expected in 2025. Upon completion of his undergraduate

studies, he intends to pursue a M.S. degree. He is a research assistance at the VisionAI Lab in the Department of AI Computer Engineering at Kyonggi University, with his primary research interests focused on deep learning, computer vision, pattern analysis, and real-time detection.



Ji-young Na will receive a Bachelor's degree in Artificial Intelligence from the Department of AI Computer Engineering at Kyonggi University in 2025. Her interests are computer vision, data analysis, deep learning,

and artificial intelligence.