

LPR 시스템 트리거 신호 생성을 위한 딥러닝 슬라이딩 윈도우 방식의 객체 탐지 및 추적

김진호*

Deep-learning Sliding Window Based Object Detection and Tracking for Generating Trigger Signal of the LPR System

Kim Jinho

〈Abstract〉

The LPR system's trigger sensor makes problem occasionally due to the heavy weight of vehicle or the obsolescence equipment. If we replace the hardware sensor to the deep-learning based software sensor in order to generate the trigger signal, LPR system maintenance would be a lot easier. In this paper we proposed the deep-learning sliding window based object detection and tracking algorithm for the LPR system's trigger signal generation. The gate passing vehicle's license plate recognition results are combined into the normal tracking algorithm to catch the position of the vehicle on the trigger line. The experimental results show that the deep learning sliding window based trigger signal generating performance was 100% for the gate passing vehicles including the 5.5% trigger signal position errors due to the minimum bounding box location errors in the vehicle detection process.

Key Words : Vehicle Detection, Vehicle Tracking, Deep-learning, LPR Trigger Signal

I. 서론

주차 관제용 LPR(License Plate Recognition)[1, 2] 시스템에서는 게이트를 열고 닫기 위해 트리거 신호를 발생시키는 하드웨어 센서를 도로 바닥에 매설하여 차량의 게이트 진입 및 진출 시점을 판단하고 있다. LPR 시스템은 차량 번호판 인식 결과와 더불어 진입 트리거 신호가 발생하면 차량이 게이트 입구에 도착한 것

으로 판단하여 게이트를 열고 진출 트리거 신호가 발생하면 차량이 게이트를 통과한 것으로 판단하여 게이트를 닫는 동작을 수행한다. 트리거 신호를 발생시키기 위해 도로 바닥에 매설한 하드웨어 센서는 차량의 중량으로 인해 파손이 되거나 자연 노후로 인한 고장 등이 발생할 수 있어서 교체해야하는 상황이 가끔씩 발생한다. 하드웨어 교체에 따른 유지 보수 및 운용에 대한 비용을 줄이기 위해 LPR 카메라기반 인공지능 딥러닝 영상 추적 알고리즘을 이용하여 트리거 신호를

* 경일대학교 전자공학과 교수(단독저자)

발생시킬 수 있다면 추가적인 하드웨어 비용 부담 없이 유지 보수에 대한 부담을 획기적으로 줄일 수 있다. 주차 관제용 LPR 시스템에 기본적으로 탑재된 차량 번호판 인식용 카메라를 이용하여 통과 차량을 탐지한 다음 이를 번호판 문자열 인식 결과와 결합하여 차량의 위치를 추적하면 진입 또는 진출 트리거 신호 발생 지점에 도착했는지 여부를 자동으로 판단하고 차단기 개폐를 위한 트리거 신호를 발생시킬 수 있다.

실시간 객체 탐지 및 추적에는 딥러닝 기반의 YOLO 라이브러리[3, 4]가 많이 사용되고 있다. 미리 학습된 80개 클래스의 객체를 탐지할 수 있는 YOLO를 SORT[4] 알고리즘과 결합하면 차량 객체의 추적에도 효과적으로 활용할 수 있다. 일반적으로 도로를 주행하는 차량을 원거리에서 촬영할 경우 이동 속도와 방향이 일정하기 때문에 인접 프레임들 사이에 크기와 형태의 변화가 작지만 번호판 인식결과의 신뢰도가 낮아 질 수 있다. 하지만 주차 관제용 게이트를 통과하는 차량의 경우 근접 촬영으로 인해 연속되는 인접 프레임들 사이에 크기와 형태 변화가 심해져서 추적이 어려울 수 있지만 번호판 인식결과의 신뢰도가 높다는 특징이 있다. 따라서 주차관제용 게이트 출입 차량의 위치 추적을 위해 번호판 인식결과를 연동시킨 전용 차량 추적기를 개발하면 기존 추적 알고리즘보다 더 효율적으로 차량의 위치 정보를 구할 수 있어서 정확하게 트리거 신호를 발생시킬 수 있다.

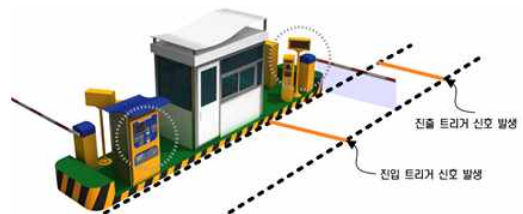
본 논문에서는 주차 관제용 LPR 시스템에서 YOLO 알고리즘으로 차량을 탐지하고 딥러닝 신경망을 이용한 번호판 인식 결과와 연동시킨 슬라이딩 윈도우 기반의 차량 객체 추적 알고리즘을 구현하여 소프트웨어적으로 트리거 신호를 생성하는 방법을 제안하였다. 주차 관제용 게이트를 통과하는 차량의 크기와 위치 정보를 번호판 인식결과와 연동시킨 게이트 출입 차량 전용 객체 추적 알고리즘을 구현함으로써 게이트 출입 차량의 위치 추적 성능을 높일 수 있도록 하였다. YOLO를 이용한 게이트 통과 차량의 위

치 탐지 결과와 LPR 시스템의 번호판 인식 결과를 활용해서 하드웨어 센서를 사용하지 않고 소프트웨어적으로 차단용 게이트 개폐용 트리거 신호를 생성할 수 있도록 하였다.

본 논문에서 제안한 게이트 트리거 신호 생성용 딥러닝 슬라이딩 윈도우 기반 차량 탐지 및 추적 알고리즘을 상용 LPR 시스템에 적용하여 차량 출입제어 실험을 수행하고 그 결과를 분석하였다.

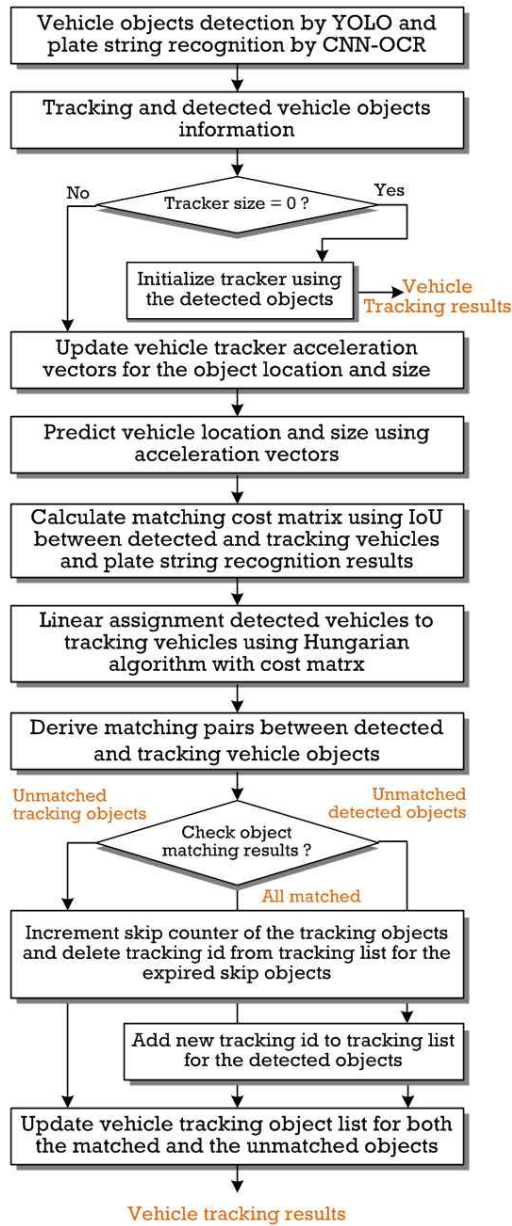
II. 딥러닝 기반의 LPR 트리거 신호 생성 알고리즘 개요

기존의 상용 LPR 시스템은 <그림 1>과 같이 카메라가 장착된 번호판 인식 시스템과 더불어 차량의 출입을 제어하는 차단기 그리고 차단기 개폐용 신호를 발생시키기 위해 도로에 매설한 트리거 신호 발생 장치 등으로 구성되어 있다.



<그림 1> LPR 시스템 차단기 개폐용 트리거 신호 발생 구조

본 논문에서는 하드웨어적인 트리거 신호 발생 장치 대신에 딥러닝 기반으로 차량을 탐지하고 CNN 기반[5-8]의 번호판 인식 결과를 연동시킨 추적 알고리즘을 구현하였다. 차량의 위치를 추적하면서 진입 및 진출 위치를 판단한 다음 게이트 제어용 트리거 신호를 발생시키는 알고리즘을 제안하였다. 딥러닝 기반 YOLO 라이브러리를 이용하여 게이트 통과 차량들을 탐지하고 위치를 추적 알고리즘을 <그림 2>에 도시하였다.



〈그림 2〉 LPR 시스템의 게이트 진입 및 진출 차량 탐지 및 추적 알고리즘

알고리즘에서는 먼저 카메라로부터 연속적으로 입력되는 영상 프레임에서 객체를 탐지하고 차량 번호판 문자열을 인식한 다음 그 결과를 차량 추적기로

전달한다. 추적기에서는 이전 영상 프레임까지의 객체 추적 결과를 바탕으로 다음 영상 프레임에서의 객체 위치를 예측한다. 추적기에서는 현재 프레임에서의 탐지 객체 위치와 크기 그리고 번호판 문자열 인식 결과를 바탕으로 비용 행렬을 구한 다음 선형 할당 문제 풀이 알고리즘을 이용하여 추적 객체들과 탐지 객체들 사이의 매칭을 시도한다. 추적 객체들 중에서 탐지 객체와 정합이 되지 않은 객체들에 대해서는 일정 횟수만큼 계속 추적을 진행하고 임계 횟수를 넘으면 추적 대상에서 제외시킨다. 탐지 객체들 중에서 추적 객체와 정합되지 않은 객체는 위치 정보와 번호판 문자열 인식 정보들을 이용해서 새로운 추적 객체로 등록한다. 마지막으로 객체 추적 결과인 차량의 위치와 차량 번호판 문자열을 출력한다.

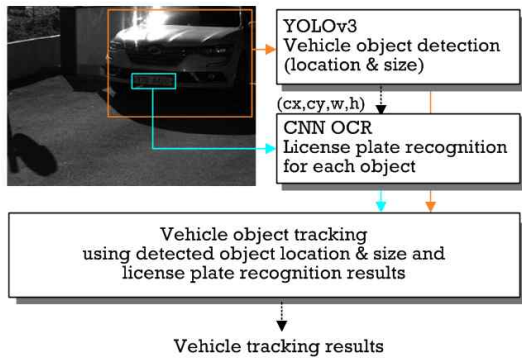
차량 객체 추적 결과를 이용해서 차량 객체가 미리 설정한 게이트 트리거 신호 발생 선의 접근 여부를 판단하고 게이트 개폐 제어용 트리거 신호를 발생시킨다. 이상의 LPR 시스템의 게이트 진입 및 진출 트리거 신호 생성을 위한 차량 탐지 및 추적 알고리즘의 구현 과정과 실험결과를 3절 및 4절에 자세하게 설명하였다.

III. 딥러닝 기반의 차량 탐지 및 추적 기반 LPR 트리거 신호 생성

딥러닝 YOLO 기반으로 차량 객체를 탐지하고 탐지된 차량 객체의 번호판을 인식한 다음 차량의 위치 정보와 번호판 인식 결과를 이용하여 게이트 통과 차량의 위치를 추적한다. 게이트를 통과하는 차량이 진입 또는 진출 트리거 라인을 통과하는지 여부를 판단하여 LPR 게이트 개폐를 위한 트리거 신호를 발생시킨다. 트리거 신호 발생을 위한 각 단계별 알고리즘을 아래에 설명하였다.

3.1 차량 객체 탐지 및 번호판 문자열 인식

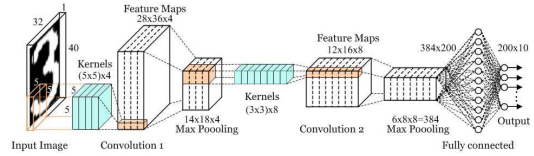
게이트 통과 차량의 위치를 추적하기 위해 먼저 LPR 카메라로 촬영한 1280x1024 크기의 영상을 대상으로 YOLOv3[3]를 이용하여 차량 객체의 위치 및 크기를 탐지하였다. GPU 기반으로 CUDA 및 cuDNN 라이브러리를 이용하여 YOLO 실행 환경을 구현하고 실시간 차량 객체 탐지를 수행한 다음 CNN 기반의 OCR에서 탐지된 차량 객체의 번호판을 인식한다 [5-7]. 차량 객체 탐지 결과와 번호판 문자열 인식 결과를 이용하여 차량을 추적하는 개념도를 <그림 3>에 도시하였다.



<그림 3> 차량 객체 탐지 및 번호판 인식 결과를 기반으로 한 차량 객체 추적 모델

번호판 문자열을 인식하기 위해 차량 객체 영상을 흑백 영상으로 변환한 다음 블롭 인덱싱을 수행하고 연속으로 나열된 문자열 블롭들을 해석하여 번호판 영역을 추출하였다. 추출된 번호판 영역의 각 문자들을 세그멘테이션 한 다음 개별 문자들의 크기를 정규화하고 문자 인식기를 이용하여 문자 인식을 수행하였다. 번호판에 기록된 문자열은 70개의 한글과 17개의 지역명 그리고 10개의 숫자들로 구성되어있다. 본 연구에서는 한글, 지역명 및 숫자 인식을 위해 각각 별도의 CNN기반 문자 인식기[6-8]를 구현하였다.

<그림 4>에 번호판에 기록된 숫자 인식을 위한 CNN 구조를 도시하였다.

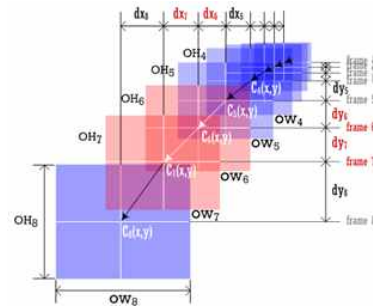


<그림 4> 번호판에 기록된 숫자 인식을 위한 CNN 구조

제안한 문자 인식용 CNN 구조는 두 개의 컨볼루션 층, 두 개의 풀링 층 그리고 두 개의 완전 결합 층들이 연결되어 있고 36x40 크기의 입력 문자 영상에 대해 출력 뉴런으로 표현되는 각 신경망의 출력 클래스 중 한 개로 문자 인식 결과를 얻을 수 있다.

3.2 주차 게이트 출입 차량 객체 추적 모델

주차 관제용 LPR 카메라에 연속적으로 입력되는 프레임들에 대한 슬라이딩 윈도우 방식의 차량 객체 이동 모델을 <그림 5>에 도시하였다.



<그림 5> 주차 관제용 게이트 출입 차량의 객체 이동 모델

프레임이 증가할수록 차량 객체의 중심 $C(x, y)$ 은 선형 특성을 갖고 dx, dy 변량으로 이동되며 객체의 크기 (OW, OH) 의 변화율도 선형 특성을 갖는다. 원거리에서는 차량의 전체 모습이 촬영되어 객체 탐지 및 추적

용이하지만 카메라에 가까워질수록 차량 객체의 크기는 매우 커지고 차량의 일부 영역이 촬영 프레임을 벗어날 수 있어서 추적 객체의 특징이 달라질 수 있게 된다. 그리고 근접 차량의 특성상 번호판 문자열 인식 신뢰도가 높아서 추적 객체의 탐지 위치 및 크기 정보에 번호판 문자열 정보를 동시에 고려하면 차량 객체의 위치 판단 및 추적의 신뢰도를 높일 수 있다. 따라서 주차 관제용 게이트 출입 차량의 이동 특성이 반영된 별도의 차량 객체 추적 알고리즘이 제안될 필요가 있다.

주차 관제용 게이트 통과 차량에 대해 YOLO를 이용한 객체 탐지 결과로 얻은 차량의 위치 및 크기 그리고 번호판 문자열 인식 결과를 이용하여 객체를 추적하는 알고리즘의 각 단계별 동작 과정을 아래에 설명하였다. 탐지 객체 목록에는 현재 프레임에서 탐지된 차량 객체들의 위치 및 크기 그리고 차량의 번호판 문자열 정보가 저장되어 있고 추적 객체 목록에는 추적 프레임 번호와 추적 중인 차량 객체들의 위치, 크기 및 변위 정보 그리고 차량의 번호판 문자열 정보가 저장되어 있다.

(1단계) 추적 차량 객체의 이동 위치 및 크기 추정

주차장 게이트 출입 차량을 연속적으로 촬영할 경우 인접한 프레임들 사이에서 차량 객체의 중심 위치 및 크기는 선형 변환 특성을 갖고 있기 때문에 이전의 경로를 참고로 다음 프레임에서의 위치와 크기를 추정하기 위한 파라미터들을 선형적인 변화율로 계산한다. 차량 객체의 이동 특성에서 객체의 이동 변위와 객체의 크기 변화율이 선형 특성을 갖는다는 점을 활용하면 다음 프레임에서의 객체 위치를 추정하는데 필요한 위치 가속 파라미터(a_x, a_y) 및 크기 가속 파라미터(a_w, a_h)들은 다음과 같은 식으로 계산할 수 있다.

$$a_x = (p) \times (dx_t/dx_{t-1}) + (1-p) \times (dx_{t-1}/dx_{t-2}) \quad (1)$$

$$a_y = (p) \times (dy_t/dy_{t-1}) + (1-p) \times (dy_{t-1}/dy_{t-2}) \quad (2)$$

식 (1) 및 (2)는 $t+1$ 번째 프레임에서의 중심 위치 추정에 사용되는 가속 파라미터(a_x, a_y)를 계산하기 위해 $t, t-1$ 및 $t-2$ 번째 프레임에서 위치 변위 정보를 활용하였다. dx_t 및 dy_t 는 각각 $t-1$ 번째에서 t 번째 프레임으로 이동할 때 x 방향 및 y 방향으로의 위치 변화율이다. P 는 한 프레임 이전과 두 프레임 이전의 변화율 반영 비율로서 값이 클수록 최근 프레임의 비중을 크게 반영하게 된다.

$$a_w = (p) \times (ow_t/ow_{t-1}) + (1-p) \times (ow_{t-1}/ow_{t-2}) \quad (3)$$

$$a_h = (p) \times (oh_t/oh_{t-1}) + (1-p) \times (oh_{t-1}/oh_{t-2}) \quad (4)$$

식 (3) 및 (4)는 $t+1$ 번째 프레임에서의 객체 크기 추정에 사용되는 가속 파라미터(a_w, a_h)를 계산하기 위해 $t, t-1$ 및 $t-2$ 번째 프레임에서 크기 변위 정보를 활용하였다. ow_t 및 oh_t 는 각각 $t-1$ 번째에서 t 번째 프레임으로 이동할 때 가로 및 세로 길이의 변화율이다.

t 번째 프레임에서 $t+1$ 번째 프레임으로 이동했을 때의 중심 위치 $C_{t+1}(x|cx_{t+1}, y|cy_{t+1})$ 는 t 번째 프레임에 도달할 때까지의 중심 위치 추정을 위한 가속 파라미터들을 이용하여 다음과 같은 방식으로 추정할 수 있다.

$$cx_{t+1} = cx_t + a_x \times (cx_{t-1} - cx_t) \quad (5)$$

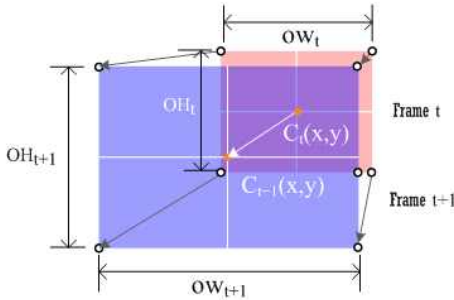
$$cy_{t+1} = cy_t + a_y \times (cy_{t-1} - cy_t) \quad (6)$$

t 번째 프레임에서 $t+1$ 번째 프레임으로 이동했을 때의 객체의 크기 (ow_{t+1}, oh_{t+1})는 크기 가속 파라미터를 이용하여 다음과 같이 구할 수 있다.

$$ow_{t+1} = a_w \times ow_t \quad (7)$$

$$oh_{t+1} = a_h \times oh_t \quad (8)$$

이와 같이 추적 중인 객체에 대해 이전 프레임들에서의 위치 및 크기를 바탕으로 구한 가속 파라미터를 이용해서 $t+1$ 번째 프레임에서의 위치와 크기를 추정한 결과를 <그림 6>에 도시하였다.



<그림 6> 인접 프레임 사이의 객체의 위치 및 크기 추정

(2단계) 탐지 객체와 추적 객체 사이의 정합

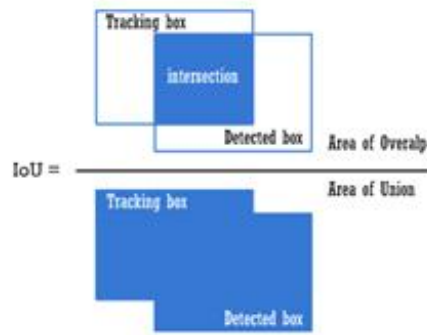
이전 프레임들 사이의 추적 객체 이동 정보를 바탕으로 다음 프레임 $t+1$ 에서의 추적 객체 위치 및 크기를 추정한 결과와 번호판 문자열 인식 결과를 바탕으로 현재 추적 중인 객체가 어느 탐지 객체에 속하는지 정합을 시도한다.

$t+1$ 번째 프레임에서 식 (5)~(8)로 추정된 추적 객체의 위치 및 크기를 이용하여 탐지 객체와 최적 정합을 시키기 위한 비용 행렬(cost matrix)을 구한다. 비용 행렬 C 는 추적 중인 n 개의 객체들을 m 개의 탐지 객체에 각각에 정합시켰을 때의 비용(cost)으로 식 (9)와 같이 행렬로 표현할 수 있다.

$$C = \begin{pmatrix} c_{11} & \dots & c_{1m} \\ \vdots & c_{ij} & \vdots \\ c_{n1} & \dots & c_{nm} \end{pmatrix} \quad (9)$$

비용 행렬 C 의 성분 c_{ij} 는 i 번째 추적 객체를 j 번

째 탐지 객체와 정합시켰을 때의 정합 비용(cost)이다. 정합 비용은 아래 <그림 7>과 같이 두 객체 사이의 IoU(Intersection of Union)을 구하고 객체 탐지 단계에서 구한 각 객체의 번호판 문자열 인식 결과를 이용하여 구할 수 있다.



<그림 7> 탐지 객체와 추적 객체 사이의 IoU

추적 객체와 탐지 객체의 합으로 이루어진 면적 대비 겹친 면적의 비율로 계산되는 IoU는 두 객체가 완전히 겹쳤을 때 최고 값인 1.0을 갖고 겹치지 않았을 때 최소값인 0.0을 갖는다. IoU가 큰 값일수록 정합 비용 c_{ij} 는 작은 값이 되도록 한다. 추적 객체와 탐지 객체의 정합 비용 c_{ij} 는 다음과 같이 구할 수 있다.

$$\begin{aligned} \text{if } s_i = s_j, \quad c_{ij} &= \begin{cases} 0_{i=j} \\ 1_{i \neq j} \end{cases} \\ \text{else } c_{ij} &= 1 - [\text{IoU between } o_i \text{ and } o_j] \end{aligned} \quad (10)$$

추적 객체 o_i 와 탐지 객체 o_j 사이의 정합 비용 c_{ij} 는 두 객체 사이의 IoU와 두 객체 사이의 번호판 문자열 인식 결과 s_i 및 s_j 를 결합해서 얻을 수 있다. 식 (10)에서 추적 객체 o_i 를 기준으로 번호판 문자열 인식 결과 s_i 가 존재하고 탐지 객체 o_j 에 동일한 문자열 인식 결과 s_j 가 있을 경우 추적 객체 o_i 를 탐지 객체 o_j 에 완전 정합시켜 정합 비용 $c_{ij} = 0$ 으

로 하고 나머지 탐지 객체들에 대해서는 $c_{i \neq j} = 1$ 로 한다. 추적 객체의 번호판 문자열 인식 결과와 동일한 탐지 객체의 번호판 문자열이 없을 경우 각 탐지 객체들과의 IoU를 계산한 결과를 이용하여 정합 비용 c_{ij} 를 구한다. 추적 객체와 탐지 객체는 중복되지 않게 1:1 정합이 이루어지므로 기존에 알려진 Hungarian 알고리즘[9]을 이용해서 정합 비용이 최소가 되는 정합 쌍을 구할 수 있다.

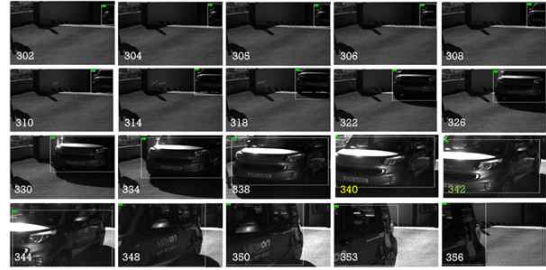
(3단계) 추적 객체의 업데이트

추적 객체가 탐지 객체와 정합된 경우 탐지 객체의 위치 및 크기 그리고 번호판 인식 결과를 추적 객체의 파라미터로 업데이트한다. 만일 추적 객체 중에서 탐지 객체와 정합 되지 않은 객체가 있다면 추적 객체의 소실 카운트(missing counter)를 증가시키고 추정한 객체의 위치와 크기로 다음 상태를 업데이트한다. 추적 객체의 소실 카운트는 추적 객체가 탐지 되지 않았을 경우 몇 프레임까지 추적을 허용할 것인지를 결정하는 파라미터이다. 소실 카운트가 미리 설정한 임계치에 이르면 더 이상 추적을 하지 않고 해당 객체를 추적 객체 목록에서 삭제한다. 탐지 객체 중에서 추적 객체에 정합 되지 않은 객체에 대해서는 이전 추적 중인 객체가 아닌 새로운 객체가 나타난 것으로 판단하고 새로운 추적 객체로 등록한다.

YOLOv3를 이용하여 게이트 통과 차량을 탐지하고 탐지된 차량 객체에서 번호판 문자열을 인식한 다음 차량의 위치 및 크기 정보 그리고 번호판 인식 결과를 이용한 객체 추적 예를 <그림 8>에 보였다.

3.3 LPR 게이트 개폐용 트리거 신호 생성

주차 게이트에 진입 및 진출하는 차량을 추적해서 객체의 하단부가 미리 설정한 트리거 신호 발생 선에 도달했을 경우에 게이트 개폐를 위한 트리거 신호를



<그림 8> LPR 카메라를 이용한 게이트 진입 차량 추적 예

발생시킨다.

차량이 게이트로 진입하는 경우에는 차량 객체가 트리거 선 밖에서 트리거 선에 겹쳐지게 되고 진출하는 경우에는 차량 객체가 트리거 선에 겹쳐져 있다가 트리거 선을 벗어나는 동작을 하게 된다.

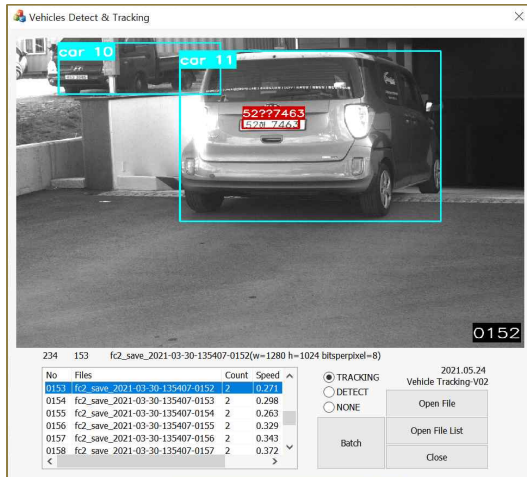
<그림 8>은 주차 게이트에 진입하는 차량의 각 영상 프레임에서 추적한 예를 보인 것으로 객체의 최소 외접 사각형 위치를 이용하여 미리 설정한 트리거 신호 발생 선에 도달했는지 여부를 판단할 수 있다.

IV. 실험 및 결과 고찰

본 논문에서 제안한 LPR 시스템 트리거 신호 생성을 위한 딥러닝 기반의 차량 객체 탐지 및 추적 알고리즘을 구현하고 상용 LPR 시스템에 적용하여 그 성능을 분석하였다. 차량 객체 탐지를 위한 YOLOv3는 GPU RTX2080 데스크탑 컴퓨터에 설치하고 CUDA 10.1 및 cuDNN v7.6.3 라이브러리들을 이용하여 개발환경을 구축하였다. 게이트 출입 차량의 번호판 문자열을 인식하기 위한 CNN 기반의 OCR을 VS2015 개발환경에서 C/C++ 코드로 직접 구현하고 상용 LPR 시스템에 탑재하였다.

제안한 알고리즘의 검증을 위해 주차 게이트 출입 차량들의 객체 추적 성능에 대한 실험과 게이트 개폐를 위해 발생시킨 트리거 신호의 정확도에 대한 실험

을 별도로 수행하였다. 이를 위해 YOLOv3를 이용한 차량 객체 탐지 및 차량 번호판 문자열 인식 결과를 이용한 차량 객체 추적 시뮬레이션 인터페이스를 <그림 9>와 같이 구현하였다.

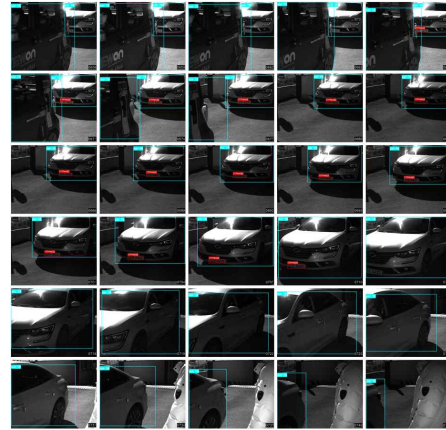


<그림 9> 객체 탐지 및 번호판 문자열 인식 결과를 결합한 차량 객체 추적 시뮬레이션 인터페이스

주차 게이트 통과 차량들의 객체 추적 실험을 위해 입차 장면 975 프레임, 출차 장면 235 프레임을 촬영하여 데이터베이스를 구축하고 차량 객체를 추적한 결과를 검토하였다. 트리거 신호 발생 알고리즘의 검증은 기존 상용 LPR 시스템에 제안한 트리거 발생 알고리즘을 구현하고 하드웨어 센서 신호와 차량 객체 추적을 통해 발생시킨 트리거 신호를 비교하는 방법으로 진행하였다.

<그림 10> 및 <그림 11>은 입차 및 출차 장면을 촬영한 영상 프레임을 대상으로 객체 추적 실험을 수행하고 3프레임 단위로 추적 결과를 도시한 것이다.

<그림 10>에서 게이트에 연속적으로 입차 되는 차량들에 대해 번호판 인식 결과를 연동시킨 차량 추적 알고리즘이 원거리에서 입차가 완료되는 과정까지 추적하는 모습을 확인할 수 있다. <그림 11>에는 다



<그림 10> 진입 차량 추적 결과를 3프레임 간격으로 출력



<그림 11> 진출 차량 추적 결과를 3프레임 간격으로 출력

른 차량이 주차되어 있는 상황에 차량의 일부가 보이는 출차 시작점에서 원거리로 출차가 종료되는 과정까지 추적한 결과를 확인할 수 있다.

주차 게이트 통과 차량들의 객체 추적 실험을 위해 게이트 진입 및 진출하는 차량들을 촬영한 각각 975개 및 235개 영상 프레임에 포함된 463대 및 116대의 차량 객체들을 대상으로 차량 탐지 및 추적 실험을 수행한 결과를 <표 1>에 도시하였다.

〈표 1〉 주차 게이트 통과 차량 탐지 및 추적 성능 분석

구분		차량 진입	차량 진출
프레임 수		975	235
차량 대수		463	116
차량 탐지 (YOLOv3)	정상 차량 탐지	446(96.3%)	113(97.4%)
	False Positive	5	3
차량 추적	Ours	442(95.5%)	112(96.6%)
	YOLOv3(SORT)[5]	435(94.0%)	108(93.1%)

YOLOv3를 이용한 객체 탐지 실험에서 진입차량의 96.3%, 진출 차량의 97.4%를 탐지하였다. 차량 탐지에 실패한 경우는 영상에 차량이 처음 나타나는 초기 프레임들에 차량의 일부 모습만 보이기 때문에 차량이 탐지 되지 않은 것으로 판단된다. 차량 탐지 결과를 이용한 차량 객체 추적 실험은 YOLO와 SORT 알고리즘을 결합[4]한 추적 결과와 본 논문에서 제안한 추적 알고리즘의 성능을 비교하였다. 제안한 알고리즘을 이용한 추적 실험에서는 진입 차량에 대해 95.5%, 진출 차량에 대해 96.6%의 추적 성능을 보여 Darknet SORT[7]를 이용하여 구현한 결과보다 비교적 높은 추적 성능을 보였다.

트리거 신호 발생 실험을 위해 상용 LPR 시스템에 제안한 알고리즘을 구현하고 실제 하드웨어 센서 트리거 신호와 비교한 결과를 <표 2>에 도시하였다.

〈표 2〉 하드웨어 센서 트리거 신호 대비 딥러닝 기반 트리거 신호 발생 결과 분석

구분		센서 신호	딥러닝 신호	
게이트 진입 차량 대수		147	272	
게이트 진출 차량 대수		125		
신호 발생 횟수	진입	147	147(100%)	272(100%)
	진출	125	125(100%)	
오차 발생 횟수	진입	-	7(4.8%)	15(5.5%)
	진출	-	8(6.4%)	

게이트 통과 차량의 추적 결과를 이용하여 게이트 개폐를 위한 트리거 신호를 발생시키기 위해서는 트

리거 라인 발생 구간에서의 차량 탐지 및 추적이 정확하게 이루어져야한다. 본 알고리즘에서는 트리거 라인 전후의 근접 차량들에 대해 번호판 문자열을 100% 인식할 수 있어서 번호판 문자열 인식 결과를 차량 추적 알고리즘에 결합시켜 진입 및 진출 차량들의 위치를 추적함으로써 통과 차량들에 대해 100% 트리거 신호를 발생시킬 수 있었다. 트리거 신호의 정확도를 분석하기 위해 일반적인 타이어 직경의 길이 60cm를 기준으로 트리거 라인 전후 60cm를 벗어난 경우 하드웨어 센서 트리거 신호에 대비해서 오차가 발생한 것으로 판단하였다. YOLOv3 객체 탐지 결과 차량 객체 외접 사각형의 위치에 오차가 발생한 경우 트리거 신호 발생에도 영향을 미쳐 하드웨어 센서 트리거 신호 대비 진입 차량들에 대해 4.8%, 진출 차량들에 대해 6.4%의 오차가 발생하여 전체 통과 차량의 5.5%에 오차 신호가 발생하였다.

실험 결과 주차 게이트 통과 차량들에 대해 LPR 번호판 인식 결과를 차량 추적 알고리즘에 결합시킬 수 있었기 때문에 비교적 정확하게 차량의 위치를 추적하고 게이트 제어용 트리거 신호를 발생시킬 수 있음을 확인하였다. 상용 LPR 시스템에 적용하기 위해서는 객체 탐지 결과 외접 사각형의 정확도를 높여 트리거 신호의 발생 오차를 최소화 하는 연구가 계속 되어야 할 것으로 보인다.

V. 결론

본 논문에서는 주차 관제용 LPR 시스템의 차단 게이트 개폐를 위해 도로 바닥에 매설한 하드웨어 센서 트리거 신호 발생 장치 대신 딥러닝 기반 YOLO 알고리즘으로 차량을 탐지하고 번호판 인식 결과와 연동시킨 차량 객체 추적 알고리즘을 구현하여 소프트웨어적으로 게이트 트리거 신호를 생성하는 방법을 제안하였다. 제안한 방법으로 차량 객체 탐지 및 추

적 실험을 수행하고 트리거 신호 발생 알고리즘을 상용 LPR 시스템에 구현하여 하드웨어 센서 트리거 신호와 신호 발생 오차를 비교하는 실험도 수행하였다. YOLOv3를 이용한 객체 탐지 실험에서 진입 및 진출 차량에 대해 각각 96.3%, 97.4%를 탐지하였고 추적 실험에서는 진입 및 진출 차량에 대해 각각 95.5%, 96.6%의 추적 성능을 보여 Darknet SORT를 이용하여 차량을 추적한 결과보다 비교적 높은 추적 성능을 보였다. 트리거 신호 발생 실험을 수행한 결과 모든 게이트 통과 차량들에 대해 트리거 신호를 발생할 수 있었으나 YOLO 차량 탐지 결과 차량의 외접 사각형 위치의 오차로 인해 5.5%의 통과 차량들에 대해 트리거 신호 생성 위치에 대한 오차가 발생하였다.

상용 LPR 시스템에 적용하기 위해 차량 객체 탐지 정확도를 더욱 높여 트리거 신호의 발생 오차를 최소로 하는 연구를 계속 진행할 예정이다.

Research, Vol. 8, Issue 11, 2019, pp.705-708.

[5] 김진호, "딥러닝 신경망을 이용한 문자 및 단어 단위의 영문 차량 번호판 인식," 디지털산업정보학회 논문지, 제16권, 제4호, 2020, pp.19-28.

[6] T. Liu, S. Fang, Y. Zhao, P. Wang and J. Zhang, "Implementation of Training Convolutional Neural Networks," arXiv:1506.01195v2, 2015.

[7] G. Sarker and S. Ghosh, "A Convolution Neural Network for Optical Character Recognition and Subsequent Machine Translation," Int. Journal of Computer Application, Vol. 182, No. 30, 2018, pp.23-27.

[8] 김진호, "딥결합 신경망을 이용한 여권 MRZ 정보 인식," 디지털산업정보학회 논문지, 제16권, 제4호, 2020, pp.19-28.

[9] The Hungarian Maximum Matching Algorithm, <https://brilliant.org/wiki/hungarian-matching>

참고문헌

[1] Hendry, R. Chen, "Automatic License Plate Recognition via sliding-window darknet-YOLO deep learning," Image and Vision Computing, Vol. 87, 2019, pp.47-56.

[2] Y. Choong, L. Keong and T. Cheaf, "License Plate Number Detection and Recognition Using Simplified Linear-Model," Journal of critical reviews, Vol. 7, Issue 3, 2020, pp.55-60.

[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," May 2016, DOI: <https://arxiv.org/pdf/1506.02640.pdf>

[4] A. Bathija and G. Shama, "Visual Object Detection and Tracking using YOLO and SORT," International Journal of Engineering

■ 저자소개 ■



김진호
Kim, Jin Ho

1992년 3월~현재
경일대학교 전자공학과 교수
2002년 2월 경북대학교 전자공학과(공학박사)
1994년 2월 경북대학교 전자공학과(공학석사)
1992년 2월 경북대학교 전자공학과(공학사)
관심분야 : 인공지능 딥러닝, 패턴인식
E-mail : anyface2009@gmail.com

논문접수일 : 2021년 11월 26일
수정일 : 2021년 12월 10일
게재확정일 : 2021년 12월 14일