Journal of The Korea Society of Computer and Information
Vol. 24 No. 3, pp. 19-27, March 2019
www.ksci.re.kr
https://doi.org/10.9708/jksci.2019.24.03.019

Improving Performance of YOLO Network Using Multi-layer Overlapped Windows for Detecting Correct Position of Small Dense Objects

Jae-Hyoung Yu*, Youngjoon Han**, Hernsoo Hahn*

Abstract

This paper proposes a new method using multi-layer overlapped windows to improve the performance of YOLO network which is vulnerable to detect small dense objects. In particular, the proposed method uses the YOLO Network based on the multi-layer overlapped windows to track small dense vehicles that approach from long distances. The method improves the detection performance for location and size of small vehicles. It allows crossing area of two multi-layer overlapped windows to track moving vehicles from a long distance to a short distance. And the YOLO network is optimized so that GPU computation time due to multi-layer overlapped windows should be reduced. The superiority of the proposed algorithm has been proved through various experiments using captured images from road surveillance cameras.

▸Keyword: Multi-layer Overlapped Window, YOLO network, Small Dense Objects, Crossing Area, Small Vehicle Tracking

I. Introduction

최근 몇 년간 딥러닝 기술의 빠른 발전으로 인해 여러 산업 분야에서 인공지능 알고리즘 적용이 매우 활발하게 진행되어오고 있다[1][2][3]. 자연어 처리 분야 영역[4]이나 영상처리 분야에서 객체 검출, 영역분할, 그리고 물체 인식 알고리즘의 성능을 크게 향상시키고 있다. 딥러닝 알고리즘은 대용량의 이미지 데이터를 학습시키고 많은 수의 레이어로 구성되어 있어 많은 연산량을 요구하기 때문에 처리속도가 느린 단점을 가지고 있다. 이를 극복하기 위해 GPU를 이용하여 속도를 보완하는 연구들이 진행되고 있다. 특히, NVIDIA와 같은 업체에서 제공하는 CUDA는 연산속도를 증가시킬 수 있는 라이브러리 환경을 제공함으로써 이미지 처리 속도를 향상시키는데 도움을 준다.

대부분의 딥러닝 알고리즘은 소형 물체들의 검출 및 인식에 있어서 단점을 가지고 있으며, 특히 밀집해 있는 소형 물체들의 정확한 분리가 어렵다는 특징을 갖고 있다. 이로 인해 원거리에서 근거리로 이동하는 물체들에 관한 이동방향과 추적을 어렵게 한다. 대표적으로 일반 도심의 도로상에 설치된 CCTV를 통해 획득한 고해상도 영상에서 겹쳐진 이동 차량들이 원거리에서 근거리로 움직이는 경우 차량 분류가 어렵다.

본 논문에서는 이를 극복하기 위한 방법으로 도로 영상에 관해 다계층 중첩 윈도우를 지정하여 각 영역에서 차량을 정확하게 검출하고, 영역간의 인접성을 통해 차량의 움직임에 대한 정보를 유지할 수 있는 방법을 제시한다. 이를 위해 가상 깊이 정보를 정의하고 그에 따른 다계층 중첩 윈도우를 적용함으로써 차량의 크기 변화나 겹쳐짐에 따른 불분명한 위치 검출의 성능을 개선한다. 또한, 다계층 중첩 윈도우 영역간의 교차영역을 지정하여 원거리로부터 근거리로 이동하는 차량들을 추적한다. 하나의 영상 내에 다수 윈도우들로 인해 발생하는 처리속도 지연은 딥러닝 알고리즘의 레이어 최적화를 통해 보완한다. 이

• First Author: Jae-Hyoung Yu, Corresponding Author: Hernsoo Hahn
*Jae-Hyoung Yu (caution0@ssu.ac.kr), School of Electronic Engineering, Soongsil University
**Youngjoon Han (young@ssu.ac.kr), Department of Smart Systems Software, Soongsil University
*Hernsoo Hahn (hahn@ssu.ac.kr), School of Electronic Engineering, Soongsil University
• Received: 2019. 01. 25, Revised: 2019. 02. 22, Accepted: 2019. 02. 22.
• This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (No.2017R1A2B4012886)

와 관계된 처리 속도와 레이어 최적화에 대한 관계성을 분석하였다. 제안한 알고리즘은 도로상에 설치된 CCTV를 통해 얻은 영상을 통해 성능을 검증하고 평가한다.

II. Preliminaries

1. Related works

딥러닝 알고리즘에 대한 관심이 많아지면서 기존의 객체 인식 분야에 있어서도 상당한 기술 발전을 이루고 있다. 사람, 자동차 등의 정형적인 객체들의 경우에는 기존의 특징 추출 기반 알고리즘에 비해 월등한 성능을 보여주고 있다. 특히, 다량의 차량을 검출하고 이들에 대한 움직임을 분석하는 교통 관제 시스템은 객체 인식의 성능을 매우 중요시 하며 빠른 차량의 속도로 인해 실시간 차량 검출을 요구한다. 이러한 딥러닝 알고리즘들은 물체 인식과 영상분할[5][6][7] 분야에서 더욱 뚜렷한 성능 개선을 보여주고 있다.

객체 인식에 사용되는 딥러닝의 초기 모델이 발표된 이후 빠르게 진화되고 있으며 그 성능에 대한 비교 분석들도 상당히 많이 연구되고 있다[8]. 객체 인식 분야에서 CNN (Convolutional Neural Network)을 기반으로 하는 인공신경망 네트워크 모델들이 대다수를 차지하고 있으며, VGGNet[9], AlexNet[10], GoogLeNet [11] 을 비롯하여 R-CNN[12], Fast R-CNN[13], SSD[14], YOLO(You Only Look Once)[15] 모델들이 대표적이라고 할 수 있다. 물론 이 인공신경망 네트워크 모델들을 성능 개선하는 형태의 연구들이 상당히 많이 진전되고 있으며, 적용 환경에 따라 다양한 변형된 네트워크 모델들이 발표되고 있다.

여러 인공신경망 모델들 중에서 YOLO(You Only Look Once) 네트워크는 기존의 모델들에 비해 유사한 정확성을 유지하면서도 처리 속도를 크게 향상시킴으로써 최근 가장 주목받는 모델중 하나이다. CNN 기반 기존 모델들은 영상 내 객체 후보 영역을 먼저 추정하고 이를 기반으로 객체 후보영역을 검출하고 객체를 분류한다. 이러한 처리과정에서 발생하는 오버헤드가 매우 크기 때문에 기존 CNN 기반 네트워크 모델들은 느린 처리 속도로 인해 실시간 처리가 요구되는 응용분야에서 사용하기 어렵다.

반면 YOLO 네트워크는 경계 상자를 검출하고 클래스를 분류하는 과정을 단일 네트워크 안에서 동시에 진행함으로써 매우 빠른 처리 속도를 보인다. YOLO 네트워크는 입력 영상을 일정 크기의 격자로 분할하고, 각 영역이 검출 객체의 경계 상자(Bounding Box)가 될 점수와 특정 객체 분류에 속할 확률로 경계 상자를 표현한다. 하지만 이러한 격자 처리로 인해 물체가 겹쳐 있거나 작은 경우에 검출 성능이 떨어지는 경향을 보인다. 특히, 작은 물체가 밀집되어 있는 경우에 객체의 정확한 위치를 추정하기 어렵다.

이들은 정확성과 속도에 있어서 상충관계를 가지고 있다. 그림1 은 각 알고리즘들의 정확성과 속도에 대한 관계성을 보여주고 있다[16].

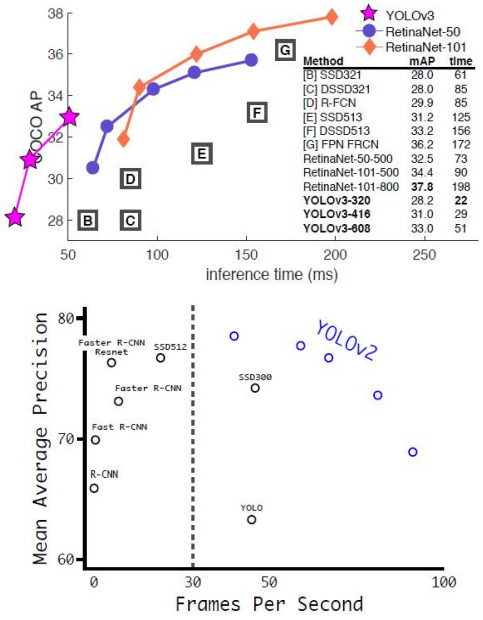


Fig. 1. (a) Comparison mAP and time of YOLO v2 with other algorithms [12] (b) YOLO v3 [17]]

이 중에서 YOLO v2의 경우는 정확성은 기존의 알고리즘에 비해 크게 떨어지지 않으면서 속도의 측면에서 우수한 성능을 갖고 있다. 이는 산업 현장에서 요구되는 실시간 처리 성능을 만족시키며 도로 교통 현장에서와 같이 검출 객체의 움직임이 빠른 경우에도 매우 적합하다.

YOLO 네트워크는 인간처럼 영상에서 물체의 종류나 위치, 관계성을 한 눈에 파악하는 방식으로 처리한다. YOLO 네트워크는 영상 내의 경계 상자와 클래스 확률을 단일 회귀 문제로 간주하여 영상을 한 번의 처리로 물체의 종류와 위치를 추정한다. 단일 컨벌루션 네트워크를 통해 멀티 경계 상자 대한 클래스 확률을 계산하는 방식이다[16].

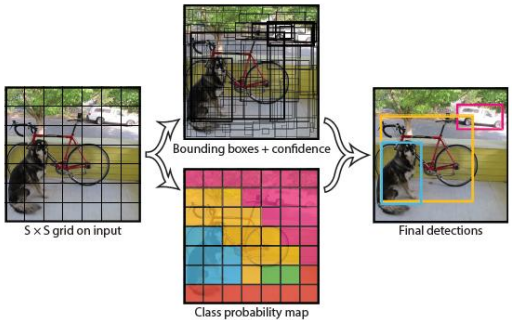


Fig. 2. Process of YOLO network model

다른 딥러닝 방식과 비교했을 때 YOLO 네트워크 모델이 보여주는 장점은 1) 간단한 처리과정으로 속도가 매우 빠르며, 2)

클래스에 대한 이해도가 높기 때문에 낮은 오검출률을 보이고, 3) 물체에 대한 일반화된 특징을 학습함으로써 높은 검출률을 보여준다. 반면, 단점으로는 작은 물체를 검출하는데 있어서 다른 알고리즘에 비해 상대적으로 낮은 정확도를 보인다.

YOLO 는 그림2 와 같이 입력 이미지를 S x S 격자 영역으로 나눈다. 각 영역은 B 개의 경계상자와 스코어 값을 갖는다. 각 격자 영역은 C 개의 상태 클래스 확률을 갖는다. 각 경계 상자는 x, y, w, h, score 로 구성된다. 여기서, (x, y) 는 상자의 중심값이며 상대적 위치 좌표를 나타내며 (w, h) 는 너비와 높이의 상대값을 나타낸다[16]. 식 (1) 은 해당 클래스의 사후 확률 값에 대한 수식을 나타낸다.

$$\Pr(Class_i|Object)*\Pr(Object)*IOU_{pred}^{truth}=\Pr(Class_i)*IOU_{pred}^{truth} \quad (1)$$

YOLO 네트워크 구조는 기본적으로 GoogLeNet 모델을 기반으로 한다[16].

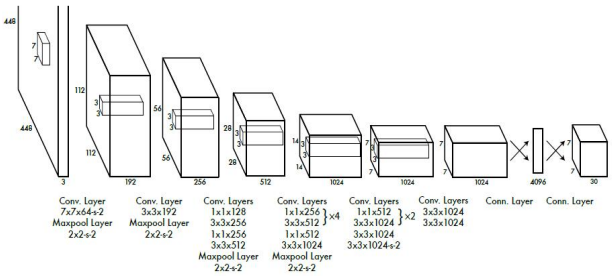


Fig. 3. YOLO Network Structure

그림 3 은 이러한 YOLO 의 기본 네트워크 구조를 보여준다. 본 논문은 소형 객체가 밀집되어 있는 상황에서 정확하게 객체들을 구분하고 그 이동 정보를 유지할 수 있도록 하는 추적방법을 제안한다. 특히 도로에 장착된 CCTV 에서와 같이 원거리 차량의 종횡 움직임 정보를 추적하기 위해 다계층 중첩 윈도우를 적용하고 각 윈도우간 교차 영역을 통해 차량의 움직임 정보를 전달하는 방법을 제시한다.

III. The Proposed Scheme

1. Vehicle Detection based on Multi-layer Overlapped Windows

YOLO 네트워크 모델은 다른 딥러닝 네트워크들에 비해 상당히 빠른 처리속도를 보여 주면서도 검출 성능 면에 있어서는 큰 차이를 보이지 않는다. 하지만 소형의 물체가 겹쳐있는 경우 그 위치를 정확하게 추정하지 못하는 단점을 가지고 있다. 이러한 단점을 해결하기 위해 다계층 중첩 윈도우를 기반으로 하는 차량 검출 방식을 제안한다.

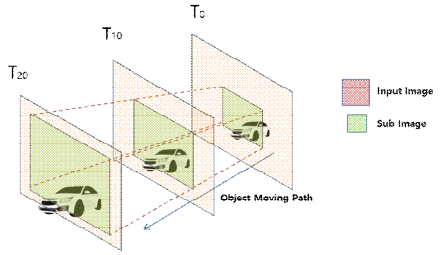


Fig. 4. Concept for multi-layer overlapped windows

그림4 에서 볼 수 있듯이 입력된 이미지의 부분 영역을 윈도우로 지정하며 이들 영역들은 서로 간 겹쳐진 공유 영역을 가진다. 일반 도로 영상에서 나타나는 가장 두드러지는 특징으로는 3차원 원근 정보가 2차원 이미지 영상에 투영되는 것이 대표적이며, 이는 고정된 각도로 장착된 CCTV 를 통해 얻어지는 영상 정보이다. 2차원 영상 정보이기 때문에 실제 3차원 깊이 정보는 없지만, 영상에서 나타나는 도로 특징인 차선 정보 등을 통해 원근 정보를 추정할 수 있다.

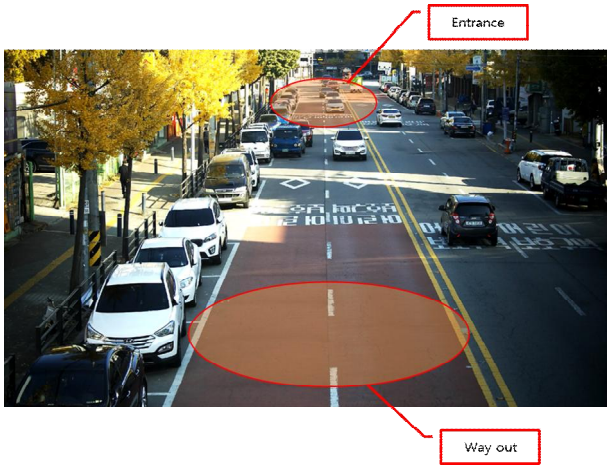


Fig. 5. Sample CCTV image captured on the road

일반 도로상에서 나타나는 차량의 경우 그림5 에서와 같이 작은 형태로 출현하여 영상을 종단하는 움직임을 보이는 경우가 많다. 또한 일반 시내 도로의 경우 신호대기로 인해 정지했다가 다수의 차량이 동시 출발하여 영상의 진입단에서 많은 차량들이 분간하기 어려운 형태로 밀집하여 나타난다.

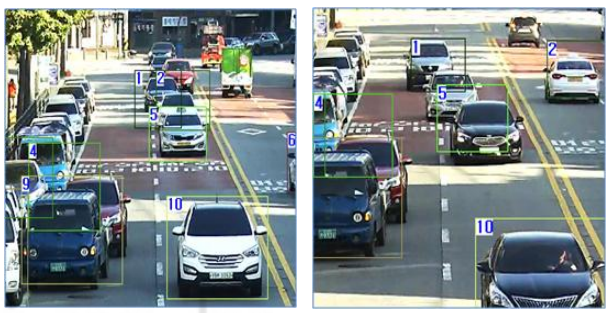


Fig. 6. Detection Result in Small Dense Vehicles case

그림6 은 밀집된 상황에서 각 차량들의 정확한 위치를 구분하지 못하는 결과들을 보여준다. YOLO 네트워크의 경우 해당 격자에서 객체의 종류와 스코어를 통해 물체의 존재 여부를 판단하게 되는데, 이때 다수의 경계 상자 정보가 겹쳐지게 된다. 이를 NMS(Non Maximum Suppression)와 같은 처리를 통해 하나의 경계 상자로 통합하고, 최종 경계 상자의 위치를 결정한다. 하지만 차량과 차량을 정확하게 구분하기 위해 NMS 값을 조정하게 될 경우 경계의 위치가 보다 정확해지는 대신 주변에서의 오검출률이 높아진다.

이러한 경우 그림 7과 같이 영상의 진입단에서 출현하는 차량들은 대부분 그 크기가 매우 작으며 겹쳐져 있는 경우가 많아 차량의 개별적인 위치를 추정하기 어렵고, 특히 차량과 차량 사이의 중간 영역을 검출하는 경우가 많아진다. 영상의 진입단에서 정확한 차량의 위치를 판별하지 못할 경우 차량의 움직임을 추적하기 어려워 차량 속도와 같은 정보를 산출하기가 어렵다. 따라서 차량이 움직이는 동선에 따라 중첩 윈도우를 배치하고 이들 윈도우에서 검출되는 차량의 이동 정보를 공유함으로써 진입단으로부터 진출단까지 종단하는 차량의 이동정보를 정확하게 판별할 수 있다. 그림7은 진입 위치에서 검출된 차량이 고유 ID를 유지하면서 진출단까지 이동하는 예시를 보여준다.

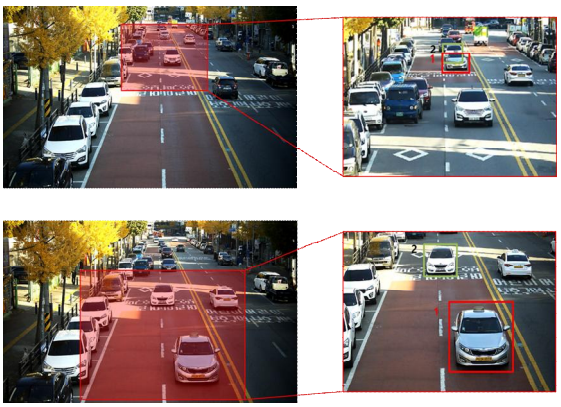


Fig. 7. Example of assign and keep ID each vehicle on entrance and exit image

각 중첩 윈도우 영역은 각 네트워크의 입력으로 주어진다. 예를 들어, 3개의 중첩 윈도우를 지정할 경우 3개의 YOLO 네트워크가 필요하다. 각 중첩 윈도우는 416 * 416 크기의 이미지로 변환되어 네트워크의 입력으로 주어지며, 각 네트워크는 각각 검출 결과를 얻는다. 각 영역에서 검출된 객체는 각각 검출 ID 를 가지며, 처음 부여된 검출 ID는 차량 추적을 위해 사용된다.

진입단 영역에서 차량 검출률을 높이기 위해 해당 영역만을 ROI(Region Of Interest) 영역으로 지정하여 YOLO 네트웍에 적용할 수 있지만, 이럴 경우 영상의 진출단 정보를 추출할 수 없어 검출된 차량의 이동에 따른 정보의 연관성을 확보할 수 없다. 따라서, 영상의 부분적인 영역 정보뿐만 아니라 영상의 전반적인 영역에서의 물체 이동 정보를 함께 고려하기 어렵다. 본 논문에서는 이를 해결하기 위해 다계층 중첩 윈도우를 적용하고 각 영역에 대한 정보 교류를 통해 영상 전체 영역에서 차량의 추적 알고리즘을 제안한다.

2. Information Sharing and Tracking between with Overlapped Windows

차량의 이동 정보를 추적하기 위해 진입단에서 처음 부여된 차량의 고유 ID를 진출단까지 동일하게 유지시켜야 한다. 하지만 영상에서 지정된 각 영역에서 검출되는 차량의 정보는 매 순간 새로운 검출 ID를 부여하기 때문에 동일한 차량인지 여부를 판단하기 어렵다. 또한, 각 영역을 지나는 차량에 각각 다른 ID가 할당된다면 진입단에서 검출된 차량이 진출단에서는 다른 ID를 가질 수 있다. 처음 부여된 차량의 고유 ID를 유지하기 위해 각 교차 영역을 지나는 차량에 대한 동일성 여부를 판단해야 한다. 이를 위해, 각 중첩 윈도우간의 경계에 교차 영역을 지정하여 차량의 동일성을 판단하고 해당 차량의 고유 ID를 다음 영역에 전달하는 알고리즘이 필요하다. 그림 8은 윈도우간 정보 전달을 위한 교차 영역의 설정 예시를 보여준다.

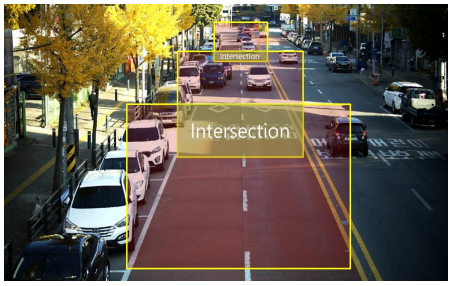


Fig. 8. Sub-Window and Intersection Area

YOLO 네트워크와 같은 딥러닝 알고리즘의 경우 검출률이 매우 높기 때문에 별도의 추적 알고리즘을 사용하지 않고 검출 결과에 대한 위치 값의 비교만으로도 차량의 추적이 가능하다. 본 논문에서 제안하는 차량추적 알고리즘은 YOLO 네트워크의 높은 검출률과 실시간 검출속도에 크게 의존한다.

중첩 윈도우간 교차 영역에서 차량의 움직임 정보를 공유하기 위해 우선 동일차량 여부를 판단한다. 동일성 판단을 위한 기준으로 물체의 부분 특징을 비교하는 방법과 위치 정보 특징 정보로 비교하는 방법으로 나눌 수 있다. 앞서 언급했듯이 YOLO 네트워크를 사용하는 경우 부분 특징의 비교 없이 단순 위치 정보의 비교만으로 차량을 추적할 수 있다.

그림 9는 첫 번째 중첩 윈도우의 설정 범위와 두 번째 중첩

윈도우의 설정 범위의 예시를 보여주고 있다. 중첩되는 윈도우는 각각 교차 영역을 포함하고 있으며, 교차 영역은 첫 번째 윈도우와 두 번째 윈도우가 겹치는 영역을 말한다. 첫 번째와 두 번째 중첩 윈도우의 교차영역의 같은 위치에서 검출된 차량은 동일하다. 이때, 두 번째 윈도우에서 차량의 일부 하단 정보만으로 차량이 검출되어 추적에 실패할 수 있다. 따라서 차량 추적의 오류를 방지하기 위해 아래 그림처럼 윈도우의 교차 영역의 일부분을 마스킹 함으로써 차량이 완전히 진입한 이후에만 검출되도록 한다.

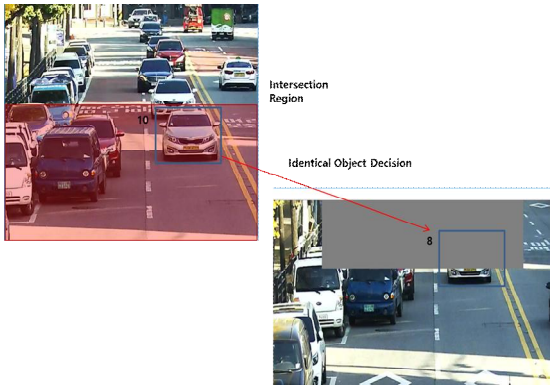


Fig. 9. Setup method for identical vehicle decision

본 논문에서는 각 프레임별로 검출된 위치 정보를 비교하는 방법을 적용하여 차량을 추적하였으며, 중첩 윈도우간의 교차 영역 내에서 이러한 정보를 공유하기 위해 검출된 하단면의 중심점의 위치 정보(BC_x, BC_y), 검출된 차량의 너비 (W) 정보를 기준으로 정하고 이를 차량 특징 정보 $F = ((BC_x, BC_y), W)$ 로 정의하였다. 중첩 윈도우간 교차 영역에서 검출된 서로 다른 프레임에서의 차량의 특징 벡터 F의 유클리디언 거리 정보 $D = |F_{t-1} - F_t|$ 를 통해 가장 작은 값을 가지는 두 차량을 동일차량으로 판단한다. 정합 후 인계 윈도우에서 차량의 ID를 인수 윈도우 영역에서 검출된 차량에 부여한다. 이 때 인계 윈도우는 상단에 존재하기 때문에 부분 차량의 정보로 보일 수 있으므로 차량의 높이 정보는 두 차량의 정합 시에 고려하지 않는다. 그림 10은 차량의 정합 알고리즘에 사용되는 특징점 및 정보 구성의 예시를 보여준다.

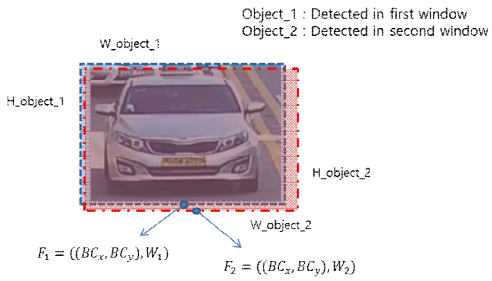


Fig. 10. Vehicle Identity Decision inside Intersection Area

3. Network Optimization

본 논문에서 제안하는 다계층 중첩 윈도우 방식은 다수의 윈도우 영상 이미지들을 네트워크에 순차적으로 입력한다. 중첩 윈도우 영상들은 각각 416x416 크기의 이미지로 균일한 크기로 변환되어 YOLO 네트워크의 입력에 전달되어 중첩 윈도우의 개수가 많아질수록 연산량과 총 수행시간은 길어진다. 따라서 이러한 단점들을 보완하기 위해 YOLO 네트워크를 최적화한다.

가장 단순한 형태의 최적화 방법은 네트워크의 각 단계에서 수행되는 컨벌루션 레이어의 필터 수를 줄이는 것이다. 보통 검출할 객체의 수가 적은 경우라면 검출률에 크게 영향을 주지 않으면서 검출 속도를 상당히 줄일 수 있다. 본 논문에서는 검출 대상 객체가 차량으로 제한되어 있기 때문에 컨벌루션 레이어의 필터수를 줄이는 방법으로 처리속도를 보완하였다. 또한, 필터의 수와 검출 성능에 대한 관계성을 알아보기 위해 컨벌루션 레이어의 필터 수의 변화에 따른 처리속도에 대한 실험을 진행하였다.

Ⅵ. Experiment

본 논문에서 제안한 방법을 검증하기 위해 도로 CCTV를 통해 획득한 10개의 동영상에서 군집 차량이 존재하는 429장의 이미지를 추출하였고, 해당 이미지에서의 이동하는 차량을 대상으로 차량의 검출 유무를 판단하였다. 입력 이미지의 원본 크기는 3392x2008이며, 24 FPS 의 동영상을 대상으로 테스트 하였다. 영상처리 및 YOLO 네트워크는 GeForce GTX 1050 TI에서 실험하였다. 본 실험은 논문에서 제시한 방법과 일반적인 단일 영상의 입력을 처리하는 방법에 대한 결과를 비교하고 있으며, 동일한 환경에서 획득한 영상들을 대상으로 진행하였다.

1. Vehicle Detection based on Multi-layer Overlapped Windows

진입단에 다수의 차량이 겹쳐있는 영상들을 총 429장 선별하였으며, 각 영상에서 움직이는 차량만을 검출 대상으로 지정하였다. 움직이지 않는 주정차 차량은 비교 대상에서 제외 하였으며, 차량이 완전히 가려지는 경우도 제외하였다. 그림 11 은 해당 환경에서 설정한 예시를 보여준다. 2차선 도로의 환경에서 원거리에서 카메라의 방향으로 접근해 오는 차량들만을 대상으로 실험하였다.

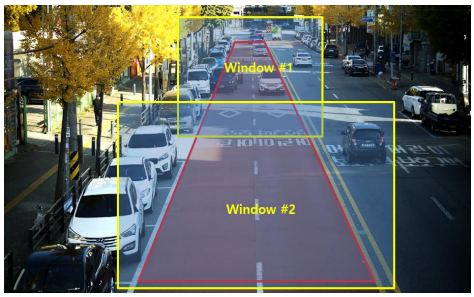


Fig. 11. Setup Multi-Windows in Experiment

다계층 중첩 윈도우의 경우 2개의 계층으로 나뉘도록 윈도우 영역을 설정하였으며, 각 윈도우의 교차영역이 존재하도록

Table. 1. Vehicle Detection Result

	Vehicle	True Positive	False Negative	False Positive	Precision	Recall	F-Score
Single Based	2079	1372	707	250	0.85	0.66	0.74
Multi-Window Based	2079	2017	62	78	0.96	0.97	0.96

지정하였다. 차량이 도로의 영역에 진입하는 위치에 첫 번째 중첩 윈도우를 설정하였고, 차량이 도로 영역을 빠져나가는 위치에 두 번째 중첩 윈도우를 지정하였다. 또한 중첩 윈도우 크기를 조절하여 첫 번째 중첩 윈도우와 두 번째 중첩 윈도우의 중간에 중첩되는 공간을 만들어 교차영역으로 지정하였다.

그림 12는 위와 같이 설정된 환경을 바탕으로 단일 영상에서 검출한 결과와 본 논문에서 제안한 다계층 중첩 윈도우를 기반으로 검출한 결과를 보여준다. 단일 영상과 다계층 중첩 윈도우는 모두 416 x 416 크기로 변환되어 YOLO 네트워크.의 입력으로 사용되었다. 모든 차량은 처음 검출 시에 고유 ID가 부여되며 검출 결과 상자의 왼쪽 상단에 표시된다. 이 고유 ID는 각 차량이 최초 검출된 시점에 부여되며 추적 알고리즘을 통해 동일 차량 정보를 알 수 있도록 지정한 정보이다.

(a) Single Image (b) Multi-Layer Image

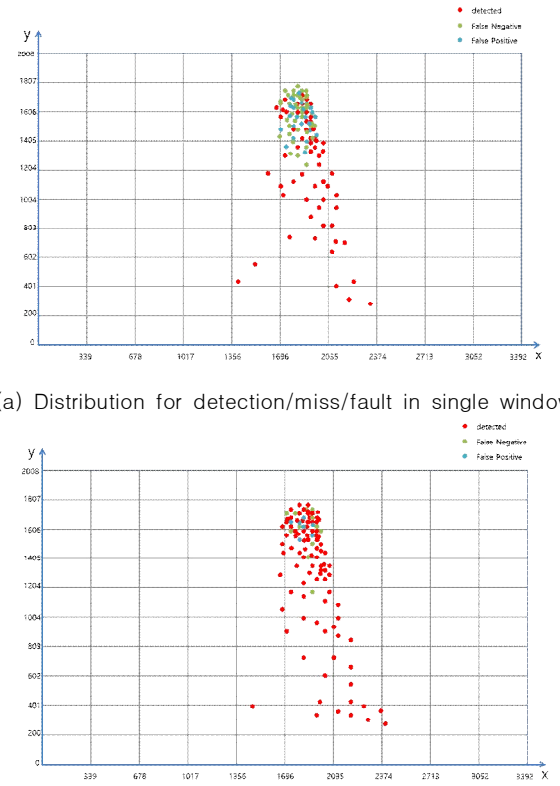
Fig. 12. Detection result for small dense vehicle

그림 12 에서 알 수 있듯이, 그림 (a)의 단일 영상 기반 차량 검출 방법은 작은 차량들의 경계선을 정확하게 검출하지 못했으며 차량과 차량 사이의 모호한 위치에서 (Convolutional Neural Network)차량들을 검출하였다. 하지만 그림 (b)의 다계층 중첩 윈도우 기반 차량 검출 방법은 동일한 조건에서 각 차량의 정확한 위치를 추정할 수 있었다. 특히, 차량들이 겹쳐져있는 상황에서도 각 차량별로 구분이 가능하여 완전히 가려지는 경우가 아니라면 차량 위치를 정확하게 추정할 수 있었다.

표 1은 단일 영상 기반 방식과 본 논문에서 제안하는 다계층 중첩 윈도우 기반 방식에서의 차량 검출 결과를 보여준다. 429장의 선별된 영상에서 유효 검출 위치에 있는 차량들을 대상으로 하였을 때, 총 차량은 2079대로, 이 중 검출 차량과 미검출, 오검출 차량의 개수를 확인하였다. 단일 영상 기반 방식의 경우 Precision은 0.8459, Recall은 0.6599를 나타냈고, 다계층 중첩 윈도우 기반 방식의 경우 Precision은 0.9628, Recall은 0.9702를 나타내었다. 이를 기반으로 좀 더 객관적인 정확성을 측정하기 위해 수식 2의 F-Measure 값을 구하였으며, 이는 Precision과 Recall을 통합하여 정확성을 한 번에 표현하는 측정 지표이다.

$$F-Measure = 2\times \frac{Precision \times Recall}{Precision + Recall} \qquad (2)$$

다계층 중첩 윈도우 기반 방식은 0.9650, 단일 영상 기반 방식은 0.7414로 계산되었다. 이를 통해 다계층 중첩 윈도우 기반 방식이 단일 영상 기반 방식에 비해 약 30% 이상의 검출 성능향상을 보였다. 특히, 단일 영상 기반 방식의 경우 미검출량이 상대적으로 많은 비중을 차지하였다.



(a) Distribution for detection/miss/fault in single window

(b) Distribution for detection/miss/fault in multi-window

Fig. 13. Voting detection result according to distance

그림 13은 랜덤하게 선택된 100개의 이미지에 대해서 차량의 위치와 검출여부에 대한 관계를 보여준다. 그림 13의 그래프는 입력된 영상에 투영된 각 영상에서 얻어진 정상검출, 오검출, 미검출에 대한 성분들의 분포를 영상의 영역 (3392 x 2008)에 색상으로 표현하였다. 그림 (a)는 단일 영상 기반 방식으로 검출된 결과를 보여주며, 그림 (b)는 다계층 중첩 윈도우 기반 방식으로 검출된 결과를 보여준다. 그림 13(a)에서 볼 수 있듯이 미검출 및 오검출의 대부분의 성분들이 하단으로부터 60% 범위 이상에서 분포되어 있으며, 하단으로부터 60% 이하에서 정상적인 검출에 대한 성분들이 분포되어 있음을 알 수 있다. 이는 단일 영상 기반 방식으로 검출하였을 경우 원거리에 속한 차량에 대한 미검출률 및 오검출률이 높다.

그러나 다계층 중첩 윈도우 기반 방식의 경우 그림 13(b)에서 알 수 있듯이 영상의 하단으로부터 80% 이상 범위에서 미검출과 오검출 성분이 다수 존재하며 그 이하의 범위에서는 정상적인 검출 성분들이 존재한다. 본 논문에서 제안하는 다계층 중첩 윈도우 기반 방식은 원거리에 있는 작고 겹쳐진 객체 검출에서도 뚜렷한 성능 개선 결과를 보여준다.

2. Vehicle Tracking Result

단일 영상 기반 방식에서 일부 영역에 관심영역을 지정하는 것은 특별한 기술은 아니다. 하지만 영상 전체에 있어서 이동하는 차량에 고유 ID를 연속적으로 부여하는 객체 추적은 단순한 관심영역의 지정만으로 구현될 수 없다. 따라서 다계층 중첩 윈도우 기반 방식은 각 중첩 윈도우간 교차영역을 두고 교차영역을 통해 차량의 고유정보를 전달함으로써 도로의 진입단에서 진출단까지 이동하는 차량을 추적할 수 있도록 하였다.

Table. 2. Vehicle Tracking Result

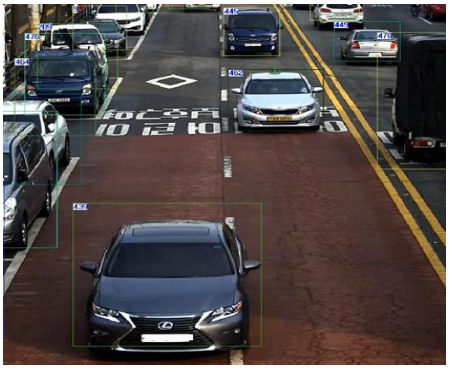
	Vehicle	Maintain	Missed	Changed	Accurate
Single Based	93	60	10	23	64.52 %
Multi-Window Based	93	85	0	8	91.40 %

표 2는 진입단에서 처음 ID를 부여 받은 차량이 진출단까지 이동하는 과정에서 고유 ID를 유지하는지 여부를 판단한 결과를 보여준다. 이는 영상 단위로 실험하였으며 유지 차량 및 변경 차량으로 구분하여 각각의 성공률을 계산하였다. 표 2의 결과에서 볼 수 있듯이, 단일 영상 기반 방식에서는 차량이 이동하면서 처음 부여 받은 고유 ID가 매우 빈번하게 변경되었다. 하지만 본 논문에서 제안하는 다계층 중첩 윈도우 기반 방식은 매우 높은 91.40% 성공률을 보여준다. 진입단에서의 차량이 미검출되거나 근접한 차량간의 모호한 위치에서 차량 검출로 인해 ID가 뒤섞이거나 사라지는 현상이 나타났다.

단일 영상 기반 추적 알고리즘은 진입할 때 부여된 차량의 고유 ID가 뒤이어 진입하는 차량의 ID로 바뀌는 현상이 자주 발생하였다. 이로 인해 처음 진입할 때 부여된 차량의 고유 ID가 진출영역에서 동일 차량의 고유 ID와 빈번하게 달랐다.

그림 14는 단일 영상 기반과 다계층 중첩 윈도우 기반 추적 알고리즘의 실험의 결과를 비교해서 보여준다. 그림 14(a)와 (b)는 단일 영상 기반 방식으로 진입단과 진출단에서 차량 추적의 실험결과를 보여주고, (c)와 (d)는 다계층 중첩 윈도우 기반방식으로 실험한 결과를 보여준다. 그림 14(a)의 진입단에서 검출된 차량의 430 ID가 그림 (b)의 진출단에서 462 ID로 변경되었다. 이는 앞서 진입한 택시가 뒤 따라 오는 소형트럭의 영향으로 고유 ID가 바뀌었다. 반면에 그림 14의 (c)와 (d)에서 진입단에서 178, 82, 그리고 187 ID의 차량들은 진출단에서도 동일하게 ID가 유지되었다.

(a) Vehicle ID in single window (entrance)



(b) Vehicle ID in single window (way out)

(c) Vehicle ID in multi window (entrance)

(d) Vehicle ID in multi window (way out)

Fig. 14. Comparison for vehicle tracking result using own ID from entrance to way out

본 실험을 통해 알 수 있듯이 차량이 진입하는 밀집 지역에서 차량의 고유 ID의 변경이 빈번하게 발생한다. 처음 진입단에서 차량을 확실하게 구분할 수 있다면 추적 알고리즘을 통해 차량의 속도나 경로와 같은 움직임 정보를 올바르게 확보할 수 있다.

3. Processing Cost Comparision with Changing Filters in Network

다계층 중첩 윈도우 기반 방식을 사용하여 겹쳐진 작은 물체에 대한 검출 성능을 향상시킬 수 있지만, 윈도우의 개수가 늘어날수록 GPU에서 처리해야 하는 연산량이 늘어난다. 따라서 윈도우의 개수를 무한정 늘릴 수 없기 때문에 네트워크의 최적화를 통해 연산량의 문제를 해결해야 한다. YOLO 네트워크의 컨벌루션 레이어의 필터 수를 줄임으로써 연산량을 줄일 수 있다.

표3은 YOLO 네트워크의 컨벌루션 레이어의 필터의 수가 검출시간과 검출률에 어떤 영향을 주는 지에 관한 실험의 결과를 보여준다. YOLO의 네트워크는 기본으로 설정된 컨벌루션 레이어의 필터의 수는 x32이다. 편의상 표에서는 첫 레이어의 필터 수를 기준으로 표시하였다. 이는 단계적으로 진행되면서 증가되게 되는데, 본 논문에서는 모든 컨벌루션 레이어의 필터 수를 첫 레이어에 대해 1/n 배수로 조정해 시간을 측정하였다.

Table. 3. Comparison accurate and time according to change filters

Filter	Time (ms)	Vehicle	True Positive	False Negative	False Positive	Accurate
x32	62.08	137	134	2	1	97.76%
x16	27.07	137	130	4	3	94.89%
x8	15.24	137	127	7	3	92.70%
x4	11.29	137	122	11	4	89.05%

결과에서 알 수 있듯이 필터의 수가 감소할수록 프레임 당 처리 시간은 감소하지만, 검출률은 다소 떨어지는 것을 알 수 있다. 그림 15는 검출속도를 초당 프레임수(FPS)로 변환하여 보여준다.

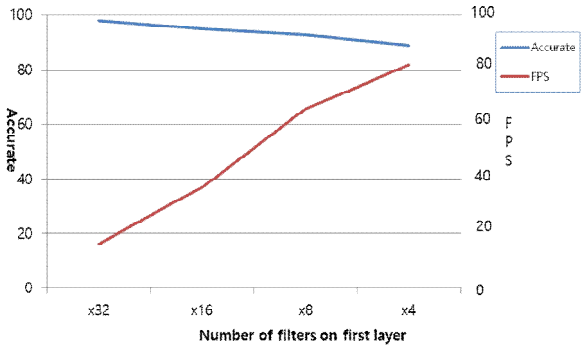


Fig. 15. Relationship accurate and FPS for changing filters

그림 15는 YOLO 네트워크에서 필터의 수를 얼마만큼 줄였을 때 기존의 검출 성능을 유지하면서 검출 속도를 개선할 수 있는지를 보여준다. x32는 처음 시작 레이어에서의 필터 수를 32개부터 시작하는 것을 의미하며, x4는 시작 필터 수를 4개로부터 시작하는 것을 의미한다. 결과로부터 알 수 있듯이 필터의 수를 감소시킬수록 검출속도의 초당 프레임수(FPS)가 증가하는 것을 볼 수 있다. 반면에 필터의 수를 감소시킬수록 검출률은 떨어지는 것을 알 수 있다. 이러한 비교를 통해 검출률과 검출 속도간의 관계를 정하고, 시스템에 적합한 YOLO 네트워크의 파라미터를 결정할 수 있다. 본 논문의 실험에서 차량과 같이 형태가 명확하고 균질한 물체에 관한 검출률을 약 90% 이상 유지하면서 처리속도를 약 60FPS 이상으로 유지할 수 있는 YOLO 네트워크의 컨벌루션 필터의 수를 정할 수 있다.

Ⅴ. Conclusions

최근 객체 검출 분야에 있어서 딥러닝 알고리즘은 없어서는 안 되는 중요한 요소이다. 이들 중에서 YOLO 네트워크는 딥러닝 네트워크의 단점인 느린 처리속도를 획기적으로 줄임으로써 주목 받고 있다. 하지만, YOLO 네트워크는 다른 딥러닝 알고리즘에 비해 검출율이 비교적 낮다는 단점을 가지고 있는데, 특히 소형 오브젝트에 대해서는 더욱 검출 성능이 낮아진다는 의견이 많다. 이에 본 논문에서는 YOLO 네트워크가 갖고 있는 소형 물체의 높은 미검출이나 밀집된 상황에서 오검출과 같은 단점들을 개선하기 위해 다계층 중첩 윈도우 기반 알고리즘을 제안하였다. 기존 YOLO 가 가지고 있는 빠른 처리 속도를 유지한 상태로 작은 물체에 관한 검출률을 높였고, 밀집되어 있는 상황에서도 오검출률을 크게 낮췄다. 이와 더불어, 객체 검출의 응용분야에서 검출 속도와 성능의 관계를 분석하여 실제 상황에 적합한 YOLO 네트워크의 파라미터를 최적화하는 방법을 제시하였다. 검출하기 위한 오브젝트의 개수가 단일하거나 적은 경우에 대해 컨벌루션 필터의 개수를 조정하여 네트워크에 입력되는 윈도우의 수가 증가함에 따른 속도 저하를 보완할 수 있었다. 필터의 수 조정에 따른 성능 저하 여부를 실험하였고 이에 대한 관계성을 파악하였다. 이를 통해 실제 환경에서 실시간성 확보를 위한 한가지 방향성을 제시할 수 있을 것으로 기대한다. 향후에는 시간 정보와 딥러닝 알고리즘의 관계에 관한 연구를 할 계획이다.

REFERENCES

[1] S. Russell and P. Norvig, “Artificial Intelligence : A Modern Approach,” NJ, USA: Prentice Hall Press, 2009.

[2] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “ImageNet large scale visual

recognition challenge," Int. J. Comput. Vis., vol. 115, no. 3, pp. 211-252, Apr. 2015.
[3] S. Ioffe and C. Szegedy, "Batch normalization : Accelerating deep network training by reducing internal covariate shift," In Proc. ICML, 2015.
[4] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent Trends in Deep Learning Based Natural Language Processing," arXiv preprint arXiv : 1708.02709. 2017.
[5] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," ICCV, pp. 2980-2988. IEEE, 2017.
[6] O. Ronneberger, P. Fischer, and T. Brox, "Unet: Convolutional Networks for Biomedical Image Segmentation," International Conference on Medical image computing and computer-assisted intervention, pp. 234-241, 2015.
[7] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation," arXiv preprint arXiv: 1511.00561, 2015.
[8] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, 1998.
[9] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," In Proc. ICLR, 2015.
[10] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, 2012, pp. 1097-1105.
[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," In Proc. CVPR, 2015.
[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 580-587, 2014.
[13] R. Girshick, "Fast r-cnn," Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440- 1448.
[14] W. Liu et al., "SSD: Single shot multibox detector," In Proc. ECCV, pp. 21-37, 2016.
[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once : Unified, real-time object detection," In Proc. CVPR, 2016.
[16] J. Redmon and A. Farhadi, "YOLO9000 : Better, Faster, Stronger", Proceedings of the IEEE conference on computer vision and pattern recognition, pp.779-788, 2016.
[17] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement", arXiv preprint arXiv : 1804.02767, 2018.

Authors

Jae-Hyoung Yu received the B.S. and M.S. degrees in Information and Electronic Engineering from Soongsil University, Korea, in 2007 and 2009, respectively Mr. Yu joined the faculty of the School of Electronic Engineering at Soongsil University, Seoul, Korea, in 2019. He is currently a student in the School of Electronic Engineering, Soongsil University. He is interested in Artificial Intelligence and Image Processing.

Youngjoon Han received the B.S., M.S. and Ph.D. degrees in Electronic Engineering from Soongsil University, Korea, in 1996, 1998 and 2003, respectively Dr. Han joined the faculty of the Department of Smart Systems Software at Soongsil University, Seoul, Korea, in 2019. He is currently a professor in the Department of Smart Systems Software, Soongsil University. He is interested in Robot Vision System, Computer Vision and Visual Servoing.

Hernsoo Hahn received the B.S. degrees in Electronic Engineering from Soongsil University, M.S. degrees in Electronic Engineering from Yonsei University, Korea, and Ph.D. degrees in Electrical Engineering from University of Southern California, USA, in 1981, 1983 and 1991, respectively Dr. Hahn joined the faculty of the School of Electronic Engineering at Soongsil University, Seoul, Korea, in 2019. He is currently a professor in the School of Electronic Engineering, Soongsil University. He is interested in Automation System, Sensor Fusion and Object Detection.