JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 26 No. 1, pp. 45-55, January 2021
https://doi.org/10.9708/jksci.2021.26.01.045

Ensemble Deep Network for Dense Vehicle Detection in Large Image

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

*Dr, School of Electronic Engineering, Soongsil University, Seoul, Korea
**Professor, Dept. Smart Systems Software, Soongsil University, Seoul, Korea
***Professor, Global future Education Institute, Soongsil University, Seoul, Korea
*Professor, School of Electronic Engineering, Soongsil University, Seoul, Korea

[Abstract]

This paper has proposed an algorithm that detecting for dense small vehicle in large image efficiently. It is consisted of two Ensemble Deep-Learning Network algorithms based on Coarse to Fine method. The system can detect vehicle exactly on selected sub image. In the Coarse step, it can make Voting Space using the result of various Deep-Learning Network individually. To select sub-region, it makes Voting Map by to combine each Voting Space. In the Fine step, the sub-region selected in the Coarse step is transferred to final Deep-Learning Network. The sub-region can be defined by using dynamic windows. In this paper, pre-defined mapping table has used to define dynamic windows for perspective road image. Identity judgment of vehicle moving on each sub-region is determined by closest center point of bottom of the detected vehicle's box information. And it is tracked by vehicle's box information on the continuous images. The proposed algorithm has evaluated for performance of detection and cost in real time using day and night images captured by CCTV on the road.

▸ **Key words**: Ensemble Deep-Learning Network, Voting Map, Dense Small Object Detection, High Resolution Image, Dynamic Windows

[요 약]

본 논문은 고해상도를 가지는 영상에서 겹쳐져있는 소형 물체를 효과적으로 검출하고 추적하는 알고리즘을 제안한다. Coarse to Fine 방식을 기본으로 하는 두 개의 Deep-Learning Network을 앙상블 형태로 구성하여 차량이 존재할 위치를 미리 판단하고 서브영역으로 선택한 이미지로부터 차량을 정확하게 검출한다. Coarse 단계에서는 서로 다른 다수의 Deep-Learning Network 에 대한 각각의 결과로 Voting Space를 생성한다. 각 Voting Space 의 조합을 통해 Voting Map을 만들고 차량이 존재할 위치를 선택한다. Fine 단계에서는 Coarse 단계에서 선택된 영역을 기준으로 서브영역을 추출하고 해당 영역을 최종 Deep-Learning Network 에 입력한다. 서브 영역은 Voting Map을 이용하여 영상에서의 높이에 적합한 크기의 동적 윈도우를 생성함으로써 정의되며, 본 논문에서는 원거리에서 근거리로 접근하는 도로의 이미지를 대상으로 미리 계산된 매핑테이블을 적용하였다. 각 서브 영역 간 이동하는 차량의 동일성 판단은 검출된 영역의 하단 중심점에 대한 근접성을 기반으로 하였으며, 이를 통해 이동하는 차량의 정보를 트래킹 하였다. 실제 주야간 도로 CCTV를 통해 획득한 실시간 영상에서 처리 속도 및 검출 성능을 비교 실험하여 제안한 알고리즘을 평가하였다.

▸ **주제어**: 앙상블 딥러닝 네트워크, 투표 맵, 밀집형 소형 물체 검출, 고해상도 이미지, 동적 윈도우

• First Author: Jae-Hyoung Yu, Corresponding Author: Hernsoo Hahn
*Jae-Hyoung Yu (grampusyu@naver.com), School of Electronic Engineering, Soongsil University
**Youngjoon Han (young@ssu.ac.kr), Dept. of Smart Systems Software, Soongsil University
***JongKuk Kim (kokjk91@ssu.ac.kr), Global future Education Institute, Soongsil University
*Hernsoo Hahn (hahn@ssu.ac.kr), School of Electronic Engineering, Soongsil University
• Received: 2020. 09. 17, Revised: 2020. 11. 25, Accepted: 2020. 11. 30.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

딥러닝 알고리즘이 보편화됨에 따라 오브젝트 검출 분야에서의 어려웠던 많은 문제점들이 해결되었고, 일반적인 영상 이미지의 물체 인식에 대한 기술이 한층 도약하였음을 알 수 있다. 특히, 처리속도를 향상시키기 위해 GPU를 활용함으로써 실시간 물체 검출이 가능해졌으며, 또한 다양한 종류의 물체와 다수의 물체를 동시적으로 처리가 가능한 형태로 구성되어 있어 영상처리를 기반으로 하는 산업분야에 획기적인 발전을 이룰 수 있는 토대를 마련하였다. 교통관제 및 보안 CCTV, 지능형 자동차, 위성 항공 등 다양한 산업분야에서 딥러닝을 기반으로 하는 물체 검출 알고리즘을 활용하고 있다[1-5]. 여러 딥러닝 알고리즘 중에서 가장 빠른 것으로 알려진 YOLO 는 실시간을 보장할 수 있기 때문에 더욱 활용도가 크다. 최근 PC 기반이 아닌 임베디드 환경에서의 활용도 또한 커지고 있다[6].

항공 영상과 같이 해상도가 매우 높은 영상 이미지에서는 검출해야 하는 물체의 크기가 매우 작다. 도로 교통 감시에 사용되는 CCTV 의 경우에도 일반적인 해상도가 매우 크며 높은 위치에 설치되기 때문에 영상에서의 차량이나 보행자의 크기는 매우 작게 투영된다. 딥러닝을 통해 물체의 검출률과 인식률이 매우 향상된 것은 사실이다. 하지만, 매우 작은 크기를 가지는 물체의 검출이 정확하지 않을 수 있다는 단점을 가지고 있으며 특히 작은 물체가 겹쳐져 있는 경우에는 그 위치를 정확하게 구분하기 어렵다는 단점이 있다. 이러한 단점을 극복하기 위한 연구들이 수행되고 있다.[7-9] 본 논문에서는 신뢰성 있는 영역분할을 통해 물체가 있을 위치를 선정하고 해당 위치를 또 다른 네트워크에 입력하여 겹쳐진 물체의 정확한 위치를 판단하기 위한 동적 윈도우 기반 앙상블 네트워크 알고리즘을 제시한다[10, 11].

본 논문에서 제시하는 시스템은 Coarse to Fine 구조를 가지는 앙상블 네트워크를 형성하며, Coarse 단계의 네트워크들에 대한 반응을 기반으로 Voting Map을 생성하고 검출하려는 물체가 있을만한 위치를 미리 선정한다. Voting Map에서 일정 값 이상을 가지는 서브 영역들을 선택하고 해당 영역에 대해서 Fine 단계의 딥러닝 네트워크를 수행한다. 이때 서브 영역은 영상에서의 물체 특징에 따라 가변적으로 지정할 수 있으며 이미지 전체 영역에 동일한 크기로 분포하거나 원근감의 특징에 따라 다른 크기로 배치할 수 있다. Coarse 단계의 Voting Map을 생성하기 위한 사전 Network 로 Faster R-CNN[12], SSD[13], YOLO v2[14] 를 사용하여 나온 각각의 결과를 종합하여 신뢰성을 높이는 형태로 적용하였다. Fine 단계의 정밀 검출을 위해 YOLO v2를 사용하였으며 제안한 알고리즘의 성능 판단을 위해 시내 도로를 촬영한 3392 x 2008 해상도의 CCTV 영상 이미지를 이용하여 평가하였다. 또한 YOLO 네트워크 이외의 다른 네트워크를 적용하여 일반적인 네트워크의 공통적인 특징을 비교하였다.

II. Preliminaries

1. Related works

딥러닝 알고리즘의 발전으로 인해 다양한 분야 및 환경에 적용되는 연구들이 매우 활발해지고 있다. 이중에서 고해상도 이미지 영상에서의 물체 검출에 대한 연구들을 볼 수 있는데, 특히 도로에서의 보행자 혹은 차량의 검출에 대한 연구들이 수행되고 있다.

Mingfei Gao[15] 는 고해상도 이미지에서 보행자를 검출하는 알고리즘을 제시하였다. 일반적으로 4K (2160 x 3840) 해상도를 가지는 영상 이미지를 Coarse to Fine 구조를 가지는 보행자 검출 시스템을 제안하고 있다. 이 구조는 두 개의 네트워크로 구성되며 정확성 Gain을 계산하기 위해 Coarse 와 Fine 단계의 연관성을 학습하는 R-net 과 그 출력을 분석함으로써 최적의 확대 영역을 선택하도록 학습하는 Q-net 으로 구성된다. 이 논문은 어느 영역에 물체가 있을지와 선택할지를 딥러닝 학습을 기반으로 결정하는 시스템을 제안하고 있다. 선택 영역과 크기가 고정되어 있어 검출 속도에 대해서는 보장할 수 없다는 단점을 가지고 있다.

Vit Ruzicka 는 [16] 4K, 8K 해상도의 영상에서 YOLO v2를 기반으로 보행자를 검출하는 알고리즘을 제시하고 있다. 입력되는 이미지를 고정 위치와 크기의 박스 영역으로 나누어 각 영역에서 검출된 결과를 통합한다. 통합된 결과는 최종 네트워크로 전달된다. Attention Evaluation 단계와 Final Evaluation 단계로 구성되며 각각 전체 검출과 자세한 검출을 진행하도록 구성되어 있다. PEViD-UHD 데이터 셋을 사용하였으며 일부 유효하지 않은 데이터 구간에 대한 처리를 추가하여 평가 데이터 셋을 구성하였다. 격자 형태의 고정된 영역으로 물체가 여러 영역에 걸쳐져 있는 경우 많은 윈도우의 네트워크 수행으로 처리 시간이 많이 걸릴 수 있다는 단점을 가지고 있다.

다른 한편으로는 위성사진과 같은 초고해상도를 가지는 영상에서의 물체 검출에 대한 연구가 진행되고 있다.

Hilal Tayara [17] 는 CNN를 기반으로 하여 Very High Resolution (VHR) 영상에서의 작은 물체 검출에 대한 방법을 제시한다. 기존의 여러 방법들은 두 스텝으로 구성된 구조를 제안하고 있는데, 이러한 방법들이 최적화하기 어렵다는 것과 실시간 처리가 용이하지 않다는 점을 해결하기 위한 방법을 제안하였다. 다양한 크기의 물체를 검출하는 것과 밀접하게 연결된 특징 피라미드 네트워크가 제안되었으며, 물체를 검출하기 위한 높은 차원의 정보를 갖는 하이 레벨의 멀티 스케일 특징 맵을 적용하였다. Top Down 방식과 Bottom Up 방식을 혼합하여 네트워크를 구성하였으며 Prediction Network 로 전달함으로써 최종 물체를 검출하도록 하는 클래스 구분하는 방법을 제안하였다.

Wu Zhihuan 은[18] 는 YOLO Network 를 기반으로 구글 어스 위성 영상에서의 비행기나 소형 차량 등을 검출하는 알고리즘을 제안한다. CNN 특징 추출, 영역 제한 등으로 형성된 DCNN (Deep Convolution Neural Network)을 이용하여 오브젝트 검출 파이프라인을 구성하였으며, YOLO 네트워크를 이용하여 물체를 검출하는 시스템을 구성하였다. 구글 어스로부터 얻은 시험 데이터를 이용하여 검증된 결과를 보여준다.

Yun Ren 는[19] 는 Faster R-CNN을 통해 위성 영상에 존재하는 작은 물체를 검출하는 알고리즘 연구결과를 보여준다. 기존의 Faster R-CNN 은 소형 오브젝트를 바로 검출하는데 어렵다는 단점을 가지고 있기 때문에 이를 해결하기 위해 네트워크의 구조를 보완하여 소형 오브젝트를 효과적으로 검출하는 방법을 제안하였다.

여러 연구들에서 볼 수 있듯이 고해상도 이미지에서 작은 물체의 검출을 위해서는 전체 영상의 영역을 서브 이미지로 분할하는 방법들이 연구되고 있으며, 어떻게 효율적으로 영역을 분할하고 물체의 존재 유무를 판단할 수 있는지가 중요하다.

딥러닝 기반의 물체 검출 알고리즘 중에서 속도 면에서 우수한 것으로 많이 알려진 YOLO Network 의 경우 정확성 면에서는 다소 성능이 떨어지는 것으로 평가되고 있다. 특히 작은 물체의 경우는 더욱 그러하며 작고 겹쳐진 물체의 경우는 위치를 검출하기 어렵다. 다른 딥러닝 알고리즘의 경우도 소형 물체가 겹쳐져 있는 경우 정확한 검출 결과를 얻기 어려운 점이 있다.

이러한 Deep Learning Network 의 단점을 극복하기 위한 방법으로 본 논문에서는 서브 영역 분할 및 선택을 효과적으로 할 수 있도록 단계적인 검출 Network Framework 를 제시한다. 첫 번째로, Coarse Detection Part에서는 글로벌 영역에서의 물체의 대략적인 특징을 검출하기 위해 입력된 이미지(3392x2008)에서 다양한 딥러닝 알고리즘을 적용함으로써 차량이 존재할 수 있는 대략적인 위치를 추정한다. 다양한 종류의 Deep Learning Network를 수행함으로써 검출 신뢰성을 확보하고, 이를 통해 Voting Map을 생성한다. Voting Map을 기반으로 미리 계산된 동적 이미지 테이블을 이용하여 입력된 전체 이미지 영역을 서브 영역으로 미리 구분한다. 이때 동적 이미지 테이블은 일반 도로의 특징을 고려하여 Perspective View 정보를 이용하여 생성한다.

두 번째로, Fine Detection Part 에서는 서브 영역으로 구분된 이미지를 입력으로 받아 새로운 Deep Learning 알고리즘에 전달하고 정밀한 검출을 진행한다. 각 서브 영역에서 검출된 차량의 정보를 통합하기 위해 오버랩 영역에서의 동일성 판단 알고리즘을 적용한다. 각 서브 영역에서 검출된 상대적 위치 정보를 원본 이미지의 절대 위치 정보로 변환하여 최종 차량 검출 정보를 획득한다. 이렇게 검출된 차량 정보는 도로상에서 움직이는 차량의 이동 및 속도 추정을 위해 트래킹 알고리즘에 반영된다.

본 논문에서 제안하는 알고리즘의 성능 평가를 위해 시내 도로에 설치된 CCTV에서 획득한 3392 x 2008 고해상도 이미지를 이용하였다. 원거리로부터 근거리로 이동하는 환경에서 차량의 움직임 정보를 추적한 결과를 비교하였다.

III. The Proposed Scheme

1. Framework

본 시스템의 기본적인 구조는 입력된 고해상도 영상을 두 단계로 구분하여 처리하도록 되어 있다. 첫 번째는 여러 개의 딥러닝 알고리즘을 통해 오브젝트가 있을 만한 위치를 추정하는 것으로, Voting Map을 생성하는 것이다. 두 번째는 Voting Map을 통해 얻어진 물체가 있을 위치를 선택하고 서브 영역으로 구분하여 새로운 딥러닝 알고리즘에 입력함으로써 세부적인 물체의 위치를 검출하는 것이다. 이때, 서브 영역을 선택함에 있어서 그 위치에 따른 크기가 미리 계산되어 있는 영상의 위치에 따른 동적 윈도우를 적용한다. Fig. 1 은 본 프레임워크를 보여준다.

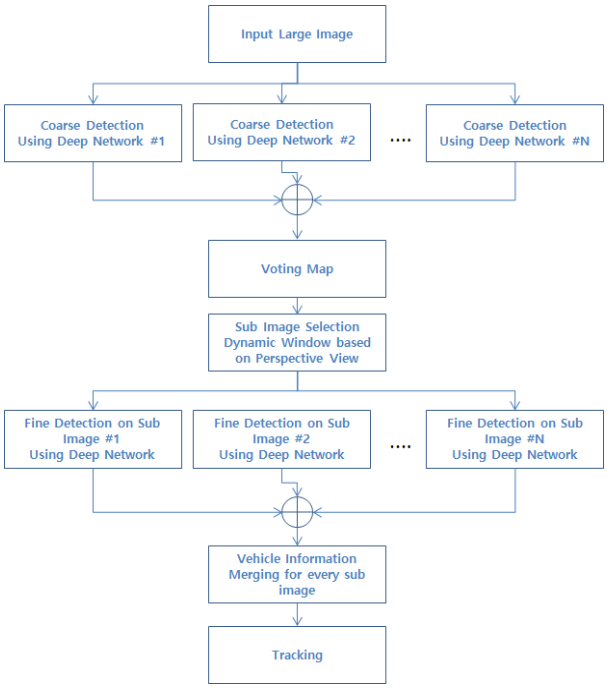


Fig. 1. System Framework

Fig. 1 에 표현된 프레임워크를 보면 Coarse Detection 파트와 Fine Detection 파트로 구분되는데, Coarse Detection 파트에서는 고해상도의 영상을 여러 개의 Deep Network에 입력하여 각 결과를 얻도록 하는데, 이때 딥러닝 알고리즘은 최대한 다양한 알고리즘을 반영한다. 본 논문에서는 Faster R-CNN, SSD, YOLO v2 등의 알고리즘을 반영하도록 구성하였다. 딥러닝 네트워크들이 가지는 특징과 성능이 각각 다르기 때문에 Voting Map을 형성할 때 최대한 서로 보완 효과를 줄 수 있다.

2. Network Structure

기본적인 네트워크의 구조는 Coarse Detection 단계에서 사용되는 네트워크와 Fine Detection 단계에서 사용되는 네트워크로 구성된다. 최종 검출 단계인 Fine Detection에서 사용되는 딥러닝 네트워크는 YOLO v2를 선택하였으며 이는 속도와 검출 성능에 대한 효율성을 바탕으로 가장 적합하다고 판단되기 때문이다. Coarse Detection 단계에서는 각기 다른 세 개의 네트워크를 선택하였으며, 검출에 있어서 가장 대표적인 YOLO v2, Faster R-CNN, SSD 네트워크를 선택하였다.

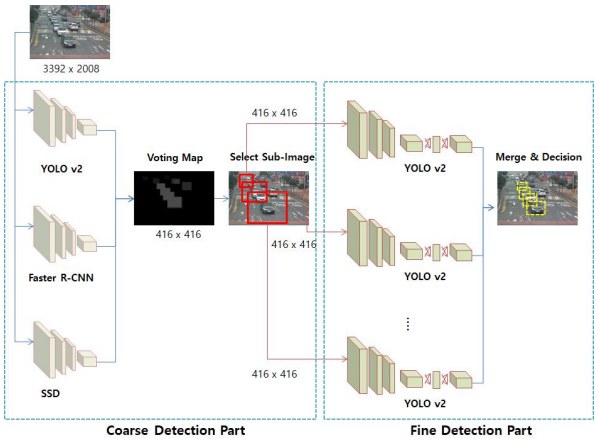


Fig. 2. Network Structure

Fig. 2 는 Coarse Detection 과 Fine Detection 단계의 구조를 예시로 보여준다. 원본 이미지는 3392 x 2008 크기의 해상도를 가지며 서로 다른 딥러닝 알고리즘의 입력으로 주어진다. 각 네트워크의 결과로 얻어진 Voting Map을 통해 서브 이미지를 추출하고 각 서브 이미지는 Fine Detection 단계의 입력으로 전달된다. 입력되는 서브 이미지는 416x416 크기로 정규화되어 전달되며 각 네트워크를 통해 서브영역에서의 차량 정보 Box를 추출한다. 각 서브 영역에서 검출된 정보를 원본 이미지에 통합하여 반영한다.

2.1 Coarse Detection

차량의 대략적인 위치를 검출하기 위한 단계로 Coarse Detection 파트를 수행한다. Coarse Detection 단계에서는 각 네트워크를 통해 검출된 오브젝트 정보를 기반으로 각각의 Voting Space 에 투영시키고 종합된 하나의 Voting Map을 생성한다. Fig. 3 은 Coarse Detection Part를 수행하는 개념도를 보여준다. Deep Learning Network #1 ~ #3 까지 각각 YOLO v2, Faster R-CNN, SSD 알고리즘을 적용하였으며, 입력 이미지는 원본 이미지 (3392x2008)를 각 네트워크의 입력 특성에 맞춰 입력한다. 각 네트워크를 통해 검출된 Box 결과를 Voting Space 에 투영한다. 검출 결과는 겹쳐진 차량의 위치를 정확하게 구분하지 않을 가능성이 크기 때문에 대략적인 차량들의 위치에 대해서만 추정하며 이들을 종합하여 Voting Map을 구성한다.

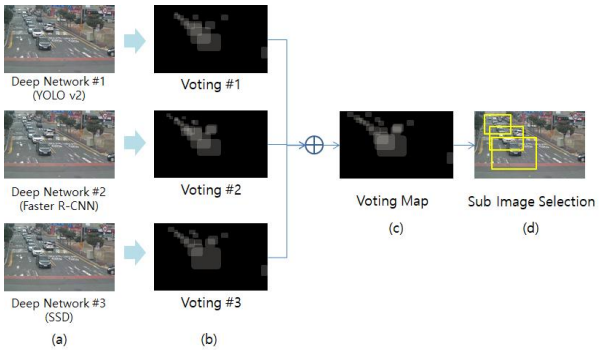


Fig. 3. Coarse Detection Part

Voting Map을 통해 대략적인 물체의 위치를 판단하며 미리 계산된 크기의 동적 윈도우를 이용하여 서브 이미지를 분리해낸다. 동적 윈도우는 도로의 특성에 따라 Perspective View 정보를 반영하여 영상의 높이에 따라 윈도우 크기를 매핑하는 테이블로 구성한다. 추출된 서브 이미지는 세부적인 검출을 위해 Fine Detection 단계의 입력들로 전달된다.

2.2. Fine Detection

Fine Detection 단계에서는 입력으로 전달된 서브 이미지들을 이용하여 오브젝트를 정밀하게 검출한다. 이때 사용된 딥러닝 네트워크는 YOLO v2를 사용하였으며 각 결과 정보를 종합하여 최종 검출 결과를 얻도록 한다.

Fig. 4 는 Fine Detection Part 의 개념도를 보여주고 있다. Coarse Detection Part에서 선택된 서브 영역들을 각각의 YOLO v2 네트워크에 입력으로 전달하여 차량을 검출한다. 이러한 서브 영역의 입력은 416x416 크기로 전달되며 이는 영상을 크게 확대하는 효과를 주기 때문에 겹쳐진 차량을 더욱 정확하게 검출할 수 있다. 이때 서브 이미지의 크기는 416x416 보다 작아지지 않도록 설정해야 하는데, 이는 원본 이미지의 왜곡을 최소화하기 위함이다.

각 서브 영역에서 검출된 차량의 정보는 마지막으로 통합되는 과정을 수행하여 각 서브 영역에서의 상대적인 위치를 원본 이미지에 대한 절대적인 위치로 변환시킨다. 이때 서브 영역간 오버랩 영역에서의 검출 결과가 중복 될 수 있으므로 이에 대한 별도의 처리가 반드시 필요하다.

Fig. 4. Fine Detection Part

3. Dynamic Sub-Window

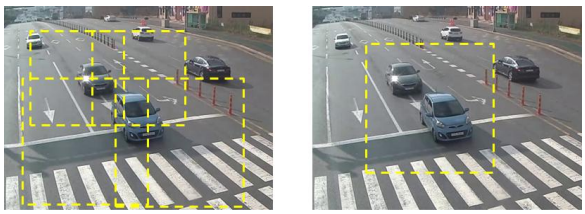
Voting Map을 통해 차량이 존재할 수 있는 위치를 추정한 이후 해당 위치를 서브 영역으로 추출하고 정밀 검출 단계로 전달해야 한다. 이를 위해서는 서브 영역을 어떻게 효율적으로 선택해야 하는지가 중요하다.

Y 축 좌표	윈도우 크기 (W x H)
411	679 x 582
572	967 x 813
733	1256 x 1045
894	1544 x 1276
1054	1831 x 1506

Fig. 5. Dynamic sub windows mapping table

기존의 방식에서는 해당 서브 영역을 일정한 간격과 동일한 크기로 분할하는 방식을 적용하거나 영상의 위치에 따라 고정된 위치에 특정 크기의 윈도우를 적용하였다. 하지만, 이러한 방식은 불필요한 영역을 검색 영역으로 포함시킴으로써 연산 속도가 증가하는 단점을 가지고 있었다. Fig. 5 는 동적 서브 윈도우를 생성하기 위해 필요한 매핑 테이

블 예시를 보여준다. 기본적으로 도로의 영상이 가지는 Perspective View 특징을 바탕으로 미리 해당 윈도우의 영상 높이에 따른 윈도우 크기를 계산한 테이블을 생성한다.



(a) (b)

Fig. 6. Compared dynamic sub window with static method

Fig. 6 은 정적 서브 윈도우와 동적 서브 윈도우의 효율성에 대한 예시를 보여준다. 해당 영상에서 차량을 검출하기 위해 필요한 최소 서브 윈도우 개수를 비교해 보면, 중앙의 두 개의 차량을 검출하기 위해 필요한 윈도우의 개수가 정적 서브 윈도우방식의 경우 4개가 고정적으로 필요한 반면, 동적 서브 윈도우 방식은 1개의 윈도우로도 해결이 가능함을 직관적으로 알 수 있다.

4. Vehicle Identify Judgement on Overlapped Region Between Sub-Windows

원본 영상을 특정 서브 영역으로 분할하고, 각 서브 영역에서 오브젝트를 검출할 경우 반드시 서브 영역들 간의 오버랩 영역이 필요하다. 만약 서브 영역의 오버랩 구간에 물체가 존재할 경우 이 물체는 오버랩 영역을 공유하는 두 개의 서브 영역에서 한 번씩 검출될 것이다. 이러한 경우 동일한 하나의 물체가 통합과정에서 두 개의 물체로 인식될 수 있기 때문에 이러한 물체를 하나의 오브젝트 정보로 인식될 수 있도록 하는 과정이 반드시 필요하다.

Fig. 7 은 서브 윈도우에서 발생하는 오버랩 영역 내부에 존재하는 하나의 차량이 각 서브 윈도우에서 검출되어 통합된 결과 예시를 보여준다. Fig. 7(a) 는 서브 영역 분할 예시이며, (b), (c) 는 각 서브 영역에서의 흰색 차량 검출 예시이다. (d) 는 (b)와 (c)의 결과를 통합한 결과 예시를 보여준다. 해당 흰색 차량의 경우 두 개의 서브 영역에서 모두 검출되었기 때문에 이 결과를 그대로 통합할 경우 하나의 차량이 두 개의 오브젝트로 인식되는 문제가 발생한다. 기본적으로 차량 검출에 있어서 차량의 하단 위치가 매우 중요하기 때문에 동일한 차량인지 여부를 판단하는 기준으로 차량의 하단 중심 위치를 지정한다.

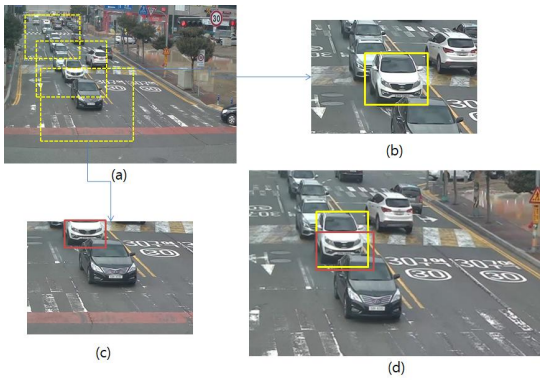
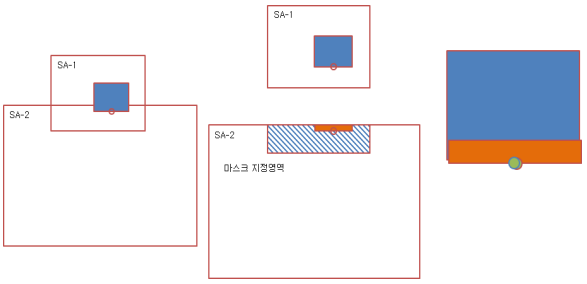
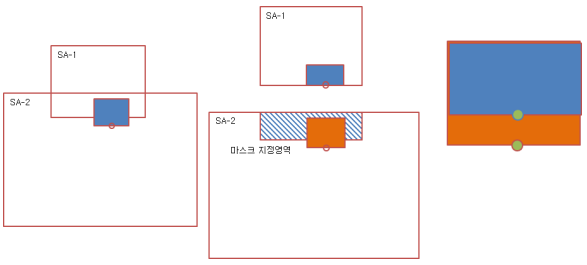


Fig. 7. Identity decision of vehicles in the overlap area for each sub windows

오버랩 구간에서 발생할 수 있는 경우는 두 가지로 구분될 수 있다. Fig. 8 은 이 두 가지의 경우를 표현한 것으로 SA-1(서브영역 1) 과 SA-2(서브영역 2) 의 서브 영역이 빗금 영역(오버랩 영역)을 공유할 때 차량이 오버랩 영역에 걸쳐 있는 경우(a)와 완전히 포함된 경우 (b)를 대표적으로 보여준다.



(a) In the case of a vehicle is lied on overlap area partially



(b) In the case of a vehicle is fully included on overlap area

Fig. 8. Relation for vehicle on each sub windows

Fig. 8 (a) 는 차량이 오버랩 영역에 일부가 걸쳐져 있는 경우로 SA-1 영역에는 차량 전체가 검출되고 SA-2 영역에는 차량의 하단 일부만 보이는 예시이다. 이때 차량의 하단 중심 위치는 두 영역에서 모두 동일하게 나타나기 때문에 중심위치의 유클리디언 거리를 측정하여 동일성 여부를 결정한다.

반면 Fig. 8 (b) 의 경우는 차량이 SA-1 영역에는 상단만 포함되어 있고, SA-2 영역에는 차량 전체가 검출되는 예시이다. 이 경우 차량의 하단 중심 위치가 두 영역에서 서로 다르게 나오기 때문에 각각 다른 차량으로 처리되기 쉽다. 차량의 하단 중심위치가 SA-1 영역의 최하위 영역 근처에 존재하는 경우 SA-2 의 마스크 지정 영역 내부에 있는 차량의 정보와 비교하여 가장 가까운 차량의 정보와 링크를 걸어주어 동일한 차량으로 인식할 수 있도록 한다. 각 서브 영역의 오버랩 구간에 대한 설정은 이미지의 형태에 따라 동적 윈도우 매핑 테이블에 비례적으로 설정하는 것이 좋다.

IV. Experimental Result

1. Environment

본 논문에서 제안한 방법을 검증하기 위한 방법으로 도로에 설치된 CCTV를 통해 획득한 고해상도 영상 250장을 대상으로 차량 검출 및 트래킹 결과를 판단하였다. 입력으로 사용한 이미지의 크기는 3392(width) x 2008(height) 이고 각 딥러닝 알고리즘에 맞는 크기로 변환하여 적용하였다. 학습을 위한 데이터는 총 5806 장을 사용하였으며, GeForce GTX 1050 TI 그래픽 카드 환경에서 구동하였다. 본 논문에서 제시한 방법을 검증하기 위해 Coarse Detection 과 Fine Detection 결과를 통해 실험 과정을 보여주며, 통합적인 결과를 통해 검출 및 트래킹 성능을 최종 비교하여 보여준다. 일반적인 단일 이미지에서의 결과와 동적 윈도우 기반에서의 결과를 비교하였다. 도로의 환경은 일반 시내도로를 대상으로 하였으며, 노면이 평평한 상태를 고려하였고, 주간 영상을 대상으로 실험하였다.

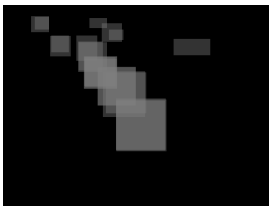
2. Coarse Detection Result

Coarse Detection 단계에서는 앞서 2-1에서 언급한대로 입력 영상에 대해 각 네트워크의 환경에 맞도록 크기를 조정하여 전달하게 된다. 본 논문에서는 Coarse Detection 단계에서 사용할 딥러닝 알고리즘으로 Faster R-CNN, SSD, YOLO v2 네트워크를 사용하였다. Faster R-CNN 와 SSD의 경우 입력 이미지의 크기는 960 x 540 을 적용하였고, YOLO v2 의 경우 416 x 416 크기를 적용하였다.

(a) Faster R-CNN

(b) SSD

(c) YOLO v2



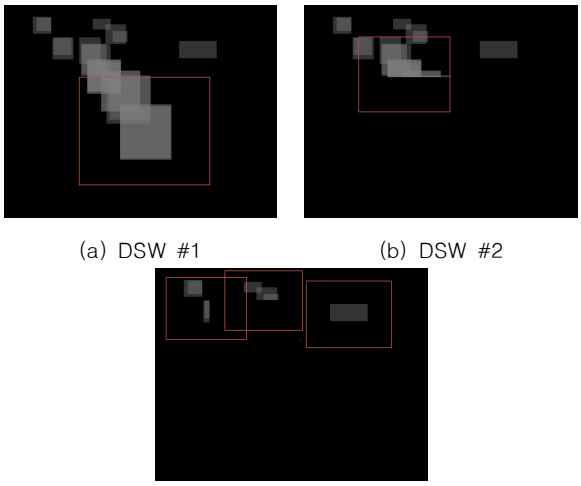
(d) Voting Map

Fig. 9. Coarse Detection Part

Fig. 9는 각 네트워크를 통해 검출된 결과 및 Voting Map 생성 결과를 보여준다. (a) 는 Faster R-CNN, (b) 는 SSD, (c) 는 YOLO v2 에 대한 Voting Space 투영 결과를 보여주며, (d) 는 이들을 통합한 Voting Map 생성 결과를 보여준다.

3. Dynamic Sub-Windows

Voting Space 의 통합을 통해 얻어진 Voting Map을 기반으로 동적 서브 윈도우를 생성할 수 있다. Dynamic Sub-Windows (DSW) 는 Voting Map에서 가장 하단을 시작으로 하여 물체가 집중되어 있는 위치의 중심을 찾아 첫 윈도우를 생성한다 이때 윈도우의 크기는 윈도우의 중심 위치에 대한 미리 정의된 매핑 정보를 이용한다.



(a) DSW #1 (b) DSW #2

(c) DSW #3~#5

Fig. 10. Integrated dynamic sub-windows

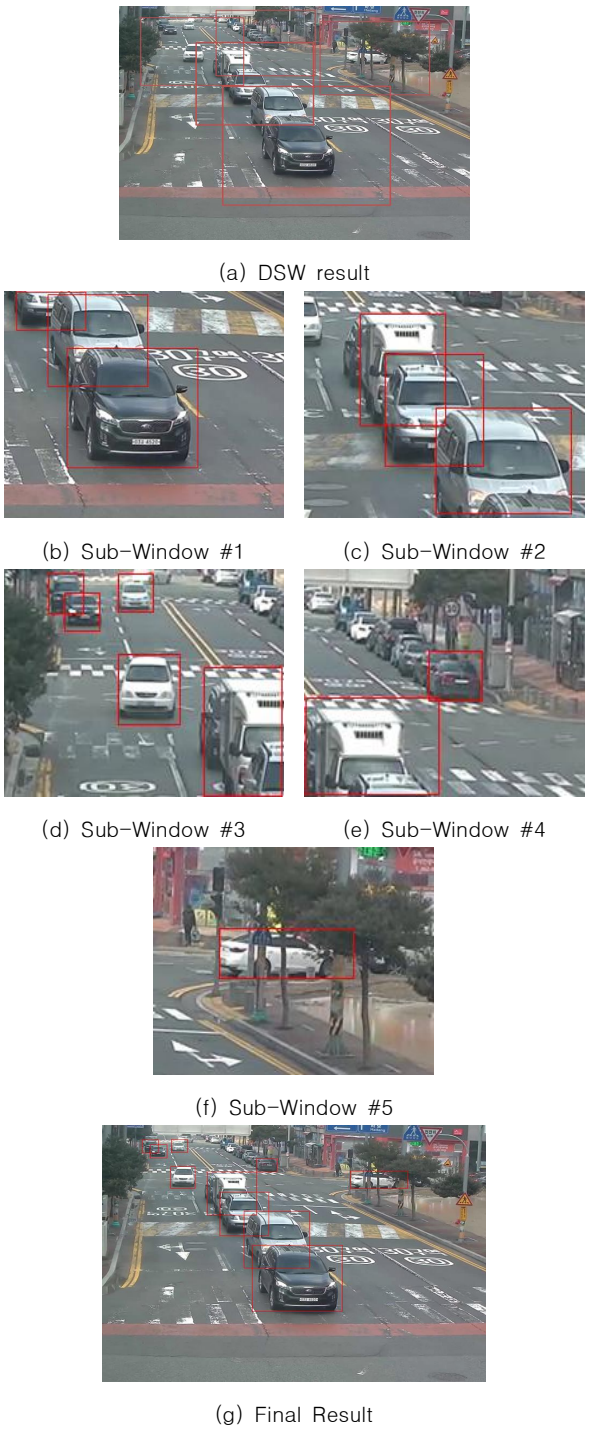
Fig. 10에서는 Voting Map을 바탕으로 동적 서브 윈도우를 생성한 결과를 보여준다. (a) 는 가장 첫 윈도우 생성 방법으로 가장 하단에 존재하는 영역의 중심을 기준으로 하는 윈도우를 생성한다. (b) 는 (a) 의 영역을 제외한 이후의 남은 영역에서 가장 하단의 위치를 중심으로 하는 위치들을 계속 반복하여 획득한다. 만약 (c) 와 같이 횡적으로 유사한 위치가 존재하는 경우는 왼쪽부터 차례대로 영역을 제거하며 윈도우를 생성하게 된다.

4. Fine Detection Result

동적 서브 영역들을 각 이미지로 분할하고, 본 논문에서 Fine Detection 단계에서 사용하는 YOLO v2 의 입력 이미지로 전달한다. 이때 입력 사이즈는 416 x 416 으로 변환하여 전달한다. Fig. 11 은 서브 영역으로 각각 분할된 결과를 보여주며 YOLO v2를 통해 검출된 각 영상의 결과를 개별적으로 보여준다.

Fig. 11 (a)는 서브 영역 분할된 각 이미지를 나타내고, (b)~(f) 는 각 서브 영역이 Fine Detection 단계의 YOLO v2를 통해 검출된 결과를 보여준다. 이렇게 검출된 각 이미지 영역들은 최종 통합되어 (g) 와 같은 결과를 보여준다.

이러한 과정을 통해 최종 차량 검출 결과를 표. 1 을 통해 확인할 수 있다. 표. 1 은 250장의 선별된 이미지에서 유효한 검출 위치에 있는 1568대의 차량들을 대상으로 단일 이미지 방식과 제안한 방식을 비교하여 실험한 결과를 보여준다. 단일 이미지 방식과 제안 방식에 대해 각각 3가지의 딥러닝 알고리즘을 적용하여 비교하였으며, 특히 제안 방식의 경우는 Fine Detection 단계에서의 검출에 적용한 결과를 비교하였다. 이 중 정상 검출된 차량의 수 (TP)와 미검출 차량의 수 (FN), 오검출 차량의 수 (FP)를 통해 Precision 과 Recall을 계산하고, Precision 과 Recall 의 Trade-Off를 통합하여 정확성을 구하기 위한 지표로 F1-Score를 계산한 결과를 보여준다.



(a) DSW result

(b) Sub-Window #1 (c) Sub-Window #2

(d) Sub-Window #3 (e) Sub-Window #4

(f) Sub-Window #5

(g) Final Result

Fig. 11. Fine detection result image using DSW

F1-Score 는 식 (1)과 같은 수식으로 얻을 수 있다[20].

$$F-Score = 2 \times (\mathrm{Precision} \times recall) / (\mathrm{Precision} + recall) \quad (1)$$

단일 이미지 방식을 이용하는 경우 Faster R-CNN 의 Precision 과 Recall 이 각각 0.91, 0.77 로 나왔고, F1-Score 는 0.83을 보여주며 SSD, YOLO v2 보다 높은 결과값을 보여주는 것을 알 수 있다. 제안한 방법의 경우는 각 딥러닝 알고리즘의 값이 Precision 은 0.96~0.98, Recall 은 0.95~0.96으로 나왔으며 F1-Score 는 0.96 의 값을 보여주었다.

Table 1. Vehicle detection result for each deep-learning method

	Obj.	TP	FN	FP	Precision	Recall	F1 Score
Single Faster R-CNN [12]	1568	1210	358	113	0.91	0.77	0.83
Single SSD [13]	1568	1128	440	86	0.93	0.72	0.81
Single YOLO [14]	1568	1019	549	158	0.86	0.65	0.74
Proposed (Faster R-CNN)	1568	1512	56	62	0.96	0.96	0.96
Proposed (SSD)	1568	1489	79	23	0.98	0.95	0.96
Proposed (YOLO v2)	1568	1507	61	45	0.97	0.96	0.96

이를 통해, 제안한 시스템이 단일 윈도우 검출 방식에 비해 검출율이 향상되는 것을 알 수 있으며, 또한 딥러닝의 알고리즘의 종류에 큰 영향을 받지 않아 균일한 결과를 보여줄 수 있다는 것도 알 수 있다.

5. Tracking Result

동적 윈도우를 이용하여 차량을 검출하는 방식은 영상의 영역을 나누어 검출된 차량의 정보를 통합한다. 이를 통해 도로상에서 움직이는 차량에 고유 ID를 부여하고 영상의 상단에서 하단까지 추적하는 시스템이 필요하다. 특히 동적 윈도우의 경우 윈도우에서 다른 윈도우로 전달되는 차량의 정보를 유지시키는 것이 매우 중요하기 때문에 이러한 차량 정보 전달이 얼마나 유효성을 가지는지를 실험하였다. 표. 2는 차량의 트래킹 성공률을 측정한 결과를 보여주며, 각각 단일 윈도우 방식과 제안한 동적 윈도우 방식에 대한 결과를 보여준다.

본 논문에서 제시하는 트래킹 방식은 검출된 차량의 박스 정보를 영상의 프레임 간 위치의 관계성을 고려하여 추적하는 방식을 사용하였으며 기존의 KCF 트래킹, TLD 트래킹과 같은 추적 방식을 사용하지 않았다. 그 이유로는 진입단에서 검출되는 차량의 정보가 명확하지 않기 때문에 그 물체를 추적하는 정보 자체가 유효하지 않을 수 있기 때문이다. 따라서 본 논문에서는 기존의 트래킹 방식과의 비교는 할 수 없으며 오로지 검출에 의한 위치 추적 방식을 기준으로 성능을 비교하였다.

Table 2. Comparison for tracking results

	Vehicle	Maintain	Missed	Changed	Accurate
Single Based	120	86	22	12	71.67%
Proposed Dynamic Window	120	117	3	0	97.50%

차량을 추적하기 위한 알고리즘은 차량 검출 결과 박스의 하단 중심 위치의 인접성과 크기를 이용하였으며, 최소 판단 범위를 적용하여 동일한 ID를 부여하도록 하였다.

표. 2의 결과를 보면, 총 120대의 차량에 대하여 실험하였다. 단일 윈도우의 경우 ID가 유지되는 경우는 86대, ID를 잃어버리는 경우는 22대, 변경되는 경우는 12대로 트래킹 정확성은 71.67%를 보여주었다. 동적 윈도우 방식은 ID를 유지하는 경우가 117대, 놓치는 경우가 3대였으며, 변경되는 경우는 없어 트래킹 정확성는 97.50%를 보여주었다. 단일 윈도우의 경우 차량 검출율이 낮기 때문에 영상 프레임간 검출 결과에 따라 사라지거나 ID가 변경되는 경우를 많이 보여주었다. 특히 차량이 겹쳐져 있는 경우 ID 가 다른 차량에 사라지거나 섞이는 경우가 많이 발생하였다. 반면 동적 윈도우 방식의 경우 ID 가 변경되는 경우는 없었으며 비교적 높은 트래킹 성공률을 보여주었다.

V. Conclusions

딥러닝 알고리즘의 발전으로 인해 물체 검출, 인식 등의 분야는 상당한 도약을 하였으며, 여러 상업화 분야에서 두드러진 결과를 보여주고 있다. 카메라의 발전에 따른 영상의 해상도 또한 많은 향상을 보여주고 있는데, 점차 고해상도의 영상이 보편화됨에 따라 물체의 검출에 있어서도 정확성이 떨어짐을 알 수 있다. 본 논문에서는 이러한 환경변화에 따라 고해상도에서도 작은 물체를 검출하기 위한 시스템을 제안하였다. 딥러닝 알고리즘에 있어서 겹쳐진 소형물체를 검출하는 문제는 또 다른 이슈이며 해결하기 위한 노력들이 많이 연구되고 있다. 본 논문에서는 입력된 영상을 차량이 존재할만한 영역들을 선별하고 이를

통해 각각의 서브 영역들에서의 딥러닝 알고리즘 수행을 적용하여 더욱 정확한 위치를 검출할 수 있도록 하였다. 이를 위해 Coarse Detection 과 Fine Detection 으로 단계를 구분하고 각 단계에 적합한 딥러닝 알고리즘을 배치하여 효과적인 시스템을 구축하였으며, 발전된 형태의 결과를 보여줌을 실험을 통해 확인하였다. 특히 소형의 겹친 물체의 정확한 위치를 검출하고 분리할 수 있었으며, 이를 트래킹을 통해 차량의 이동 정보를 함께 추출하였다. 차량의 속도를 추정하고 계도할 수 있는 시스템을 구축하기 위해 필요한 요소로서 활용될 수 있을 것으로 기대한다. 앞으로는 다양한 날씨 및 도로의 형태에 따라 적응적으로 대응할 수 있는 시스템에 대한 연구를 진행할 예정이다.

REFERENCES

[1] Q. QIN, Y. YUAN, R. LU, “A New Approach to Object Recognition on High Resolution Satellite Image,” International Archives of Photogrammetry and Remote Sensing. Vol. XXXIII, Part B3. Amsterdam 2000.

[2] M. Müller and K. Segl, “Object Recognition Based on High Spatial Resolution Panchromatic Satellite Imagery,” Proc. ISPRS Joint Workshop - Sensors and Mappingfrom Space 1999 (Hannover).

[3] M. Yang, N. Otterness, T. Amert, J. Bakita, J. H. Anderson and F. D. Smith, “Avoiding Pitfalls when Using NVIDIA GPUs for Real-Time Tasks in Autonomous Systems,” ECRTS, Euromicro Conference on Real-Time Systems. No. 20, pp. 20:1-20:21, 2018.

[4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth and B. Schiele, "The cityscapes dataset for semantic urban scene understanding." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 3213-3223, 2016.

[5] Y. Lv et al., “Traffic Flow Prediction with Big Data: A Deep Learning Approach,” IEEE Trans. Intelligent Transportation Systems, vol. 16, no. 2, 2015, pp. 865–73.

[6] L. B. Guaman, J. E. Naranjo and A. Ortiz, “Deep Learning Framework for Vehicle and Pedestrian Detection in Rural Roads on an Embedded GPU,” Electronics 2020, 9, 589.

[7] A. Marcu and M. Leordeanu, “Dual local-global contextual pathways for recognition in aerial imagery,” Computing Research Repository (CRR) abs/1605.05462.

[8] 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.

[9] S. Razakarivony and F. Jurie, “Vehicle detection in aerial imagery: A small target detection benchmark,” Journal of Visual Communication and Image Representation, 34, 187-203, 2016.

[10] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar and P. A. Muller, “Deep Neural Network Ensembles for Time Series Classification,” International Joint Conference on Neural Networks (IJCNN), arXiv:1605.05462, 2019.

[11] H. Noh, S. Hong and B. Han, “Learning Deconvolution Network for Semantic Segmentation,” Computer Vision and Pattern Recognition, arXiv:1505.04366, 2015.

[12] S. Ren, K. He, R. Girshick and J. Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks,” Computer Vision and Pattern Recognition Cite as: arXiv:1506.01497

[13] W. Liu et al., “SSD: Single shot multibox detector,” In Proc. ECCV, pp. 21-37, 2016.

[14] J. Redmon and A. Farhadi, “YOLO9000 : Better, Faster, Stronger”, Proceedings of the IEEE conference on computer vision and pattern recognition, pp.779-788, 2016.

[15] M. Gao, R. Yu, A. Li, V. I. Morariu and L. S. Davis, “Dynamic Zoom-in Network for Fast Object Detection in Large Images,” Computer Vision and Pattern Recognition (CVPR), arXiv:1711.05187, 2018.

[16] V. Růžička and F. Franchetti, “Fast and accurate object detection in high resolution 4K and 8K video using GPUs,” IEEE High Performance extreme Computing Conference (HPEC), arXiv:1810.10551, 2018.

[17] H. Tayara and K. T. Chong, “Object Detection in Very High-Resolution Aerial Images Using One-Stage Densely Connected Feature Pyramid Network,” MDPI, Sensors 2018, 18(10), 3341. 2018.

[18] Z. Wu, X. Chen, Y. Gao and Y. Li, “Rapid Target Detection in High Resolution Remote Sensing Images Using YOLO Model,” The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume XLII-3, 2018.

[19] Y. Ren, C. Zhu and S. Xiao, “Small Object Detection in Optical Remote Sensing Images via Modified Faster R-CNN,” MDPI, SCI 2018, 8(5), 813, 2018.

[20] D. M. W. POWERS, “EVALUATION: FROM PRECISION, RECALL AND F-MEASURE TO ROC, INFORMEDNESS, MARKEDNESS & CORRELATION,“ Journal of Machine Learning Technologies, Volume 2, Issue 1, pp.37-63, 2011.

Authors

Jae-Hyoung Yu received the B.S. and M.S. and Ph.D. degrees in Information and Electronic Engineering from Soongsil University, Korea, in 2007 and 2009, respectively. Dr. 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.

He received the Ph.D. degree in Electrical Engineering from Soongsil University in 2005. He is currently the Professor of Global Future Education Institute at Soongsil University. His research interests include speech synthesis, speech recognition, audio coding, and Digital Signal Processing

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.