Transactions of KSAE, Vol. 29, No. 3, pp.283-288 (March, 2021)

pISSN 1225-6382 / eISSN 2234-0149
DOI http://dx.doi.org/10.7467/KSAE.2021.29.3.283

YOLO v3를 이용한 높은 정확도의 차량 계수 방법

이 태 희[1] · 박 영 석[2,3] · 김 영 모[3] · 최 두 현[*3]

지능형자동차부품진흥원 시험평가실[1] · 엠제이비전테크[2] · 경북대학교 전자공학부[3]

A Method of Counting Vehicle with High Accuracy Using YOLO v3

Tae-hee Lee[1] · Young-seok Park[2,3] · Young-mo Kim[3] · Doo-hyun Choi[*3]

[1]*Test & Evaluation Department, Korea Intelligent Automotive Parts Promotion Institute, 201 Gukgasandanseo-ro, Guji-myeon, Dalseong-gun, Daegu 43011, Korea*

[2]*MJVisionTech, 40 Yeonam-ro, Buk-gu, Daegu 41542, Korea*

[3]*Shcool of Electronic and Electrical Engineering, Kyungpook National University, Daegu 41566, Korea*

(Received 11 January 2021 / Revised 25 January 2021 / Accepted 28 January 2021)

Abstract : A method for counting the running and queuing vehicles through installed traffic surveillance cameras at intersections has been studied for a long time. Recent research via deep learning has shown many breakthroughs with high performance results that were not achieved with traditional machine learning algorithms. In the field of object detection, these algorithms have shown high accuracy in real traffic environments, but have relatively low accuracy concerning small vehicles over a long distance, the number of which is required in counting queuing vehicles. In this paper, we are proposing a method to improve detection performance by optimizing the size of the CNN network and the size of the input image by using an open source-based, deep learning framework, YOLO, to increase the detection accuracy of small vehicles over a long distance. This study aims to improve the accuracy of vehicle counting by as much as 4.6 % over the existing method.

Key words : Vehicle detection(차량 검출), Deep learning(딥러닝), Convolutional neural network(합성곱신경망), Traffic surveillance data(교통감시데이터), YOLO(You Only Look Once, 욜로)

1. 서 론

자동차 대수 및 차량 통행량이 증가함에 따라 교통 혼잡과 정체로 인해서 미세먼지 발생, 혼잡비용 증가, 교통사고 발생 빈도 증가 등 많은 문제점들이 발생하고 있다. 이러한 문제를 해결하기 위한 한 방법으로, 도로 교통 상황을 분석하여 교통 빅 데이터(통행량, 사고, 공사 등)를 구축하고 있으며, 이 데이터를 활용한 정확도가 높은 데이터 분석을 통해 교통 신호를 제어하는 효율적인 교통관리 기술이 요구되고 있다. 이러한 지능형 교통 시스템(Intelligent traffic system)을 구현하기 위해서는 실시간 교통정보를 지속적으로 수집하고 차량의 궤적(Trajectory) 및 차로별 차량대기길이(대기열)를 정확하게 분석하는 컴퓨터 비전 기술이 필요하다.[1,2] Photo. 1의 UA-DETRAC 데이터 셋은 차량 검출 알고리즘이 적용되는 다양한 실제 환경을 보여주고 있다. 그림에서 보듯이, 교통 감시 카메라로부터 차량을 검출하기 위해서는 카메라 영상의 다양한 스케일(Scale), 형태(Type), 원근(Perspective), 폐색(Occlusion), 조도 상태(Lighting/Brightness condition) 그리고, 기상(Weather) 등 매우 다양한 조건의 영상이 획득된다. 교통감시 카메라를 이용할 경우, 다양한 환경에서 정상적으로 작동할 수 있는 정확도 높은 강건한 알고리즘이 필요하다는 것을 의미한다. 특히, 차로별 대기열을 분석하기 위해서는 먼 거리에 있는 아주 작은 차량에 대해서도 정확하게 검출하여 계수(Counting)하는 컴퓨터 비전 기술이 필요하다.

최근 CNN(Convolutional Neural Network) 기반의 딥러닝 기술을 이용하여 조도 변화와 열화 된 영상에서 높은

*Corresponding author, E-mail: dhc@ee.knu.ac.kr

Photo. 1 Sample images of the UA-DETRAC benchmark dataset[3)]

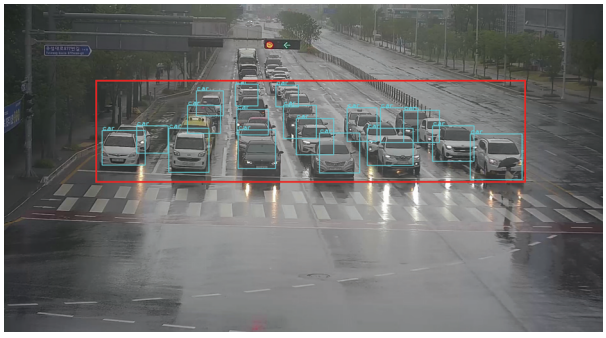

Photo. 2 Failed to detect small vehicles over long distance

정확도의 차량 검출 알고리즘이 소개되고 있다.[4-8)] 그러나 기존 알고리즘들은 차량의 크기가 큰 경우에는 높은 정확도를 보여주고 있으나 먼 거리에 있는 작은 크기의 차량에 대해서는 상대적으로 낮은 정확도를 보여주는 단점이 있다.

Photo. 2는 오픈소스 기반의 딥러닝 프레임워크 YOLO (You Only Look Once)에서 19개의 CNN 계층으로 설계한 Darknet-19 네트워크 모델을 이용하여 교통 감시 카메라로부터 획득된 영상에 대해 차량을 검출하는 모습을 보여주고 있다. Photo. 2에서 붉은색 영역은 차량 검출이 가능한 영역을 의미한다.

Photo. 2에서 보듯이, 교차로의 횡단보도에서 가까운 위치에 있는 크기가 큰 차량에 대해서는 높은 정확도의 차량 검출이 가능하지만 먼 거리의 크기가 작은 차량에 대해서는 검출하지 못하는 것을 볼 수 있다. 따라서 본 논문에서는 YOLO v3의 Darknet-53 네크워크 모델이 YOLO v2의 Darknet-19 네트워크 모델보다 먼 거리에 있는 작은 크기의 차량 검출 성능이 얼마나 향상되는지를 알아보고자 한다. 또한, 객체를 검출하기 위해서 카메라로부터 획득되는 YUV 형식의 입력 영상을 RGB 형식으로 변환하는 전처리 과정의 계산량을 줄이는 알고리즘을 적용하여 Darknet-53의 깊은 네트워크 모델을 사용하면서도 상대적으로 빠른 객체 검출이 가능한 방법을 제시한다.

2. 교통 감시 카메라를 이용한 지능형 교통 체계 구축 기술 동향

버스, 도시철도, 자율주행 차, 드론 등 오늘날 인류의 교통수단은 더욱 다양하고 복잡한 형태로 발전하고 있다. 이러한 교통수단의 다양화와 함께 교통사고, 만성적인 교통 혼잡 등의 문제를 해결할 수 있는 지능형 교통체계 기술의 필요성은 해마다 증가하고 있으며 이는 스마트시티(Smart city)의 가장 중요한 기술로 평가받고 있다. 지능형 교통체계를 구축하기 위해서는 교차로의 접근로별 정확한 교통정보 수집 기술, 수집된 정보를 이용하여 교통 혼잡도 및 위험도를 예측하는 기술 그리고, 예측된 정보를 시민들에게 다양한 형태로 제공하는 서비스 기술들이 필요하다. Fig. 1은 이러한 지능형 교통 체계 시스템의 요소 기술 및 이를 활용한 교통 혼잡도 예측 모델을 통해서 여러 가지 서비스를 제공하는 흐름을 보여주고 있다. 본 논문은 이 가운데 감시 카메라에서 획득되는 영상 데이터로부터 교통량 분석에 필요한 차량의 검출 및 대기열에 관한 내용이다.

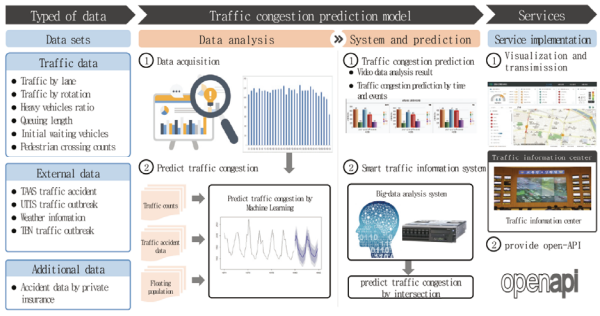

Fig. 1 The goal of intelligent traffic system

3. YOLO v3의 Darknet 53을 이용한 차량 검출 방법

3.1 학습 데이터 구축

본 논문에서 사용된 학습 데이터 집합은 교차로 20개소에 설치된 교통 감시 카메라로부터 획득하였으며 영상의 해상도는 1920×1080의 고해상도 영상을 사용하였다. Photo. 3은 각 교차로에서 획득된 다양한 형태의 학습 데이터를 보여주고 있다.

Photo. 3 Sample images taken at 20 intersections

Table 1 Number of data by time

Day			Night		
Category	Time	Count	Category	Time	Count
D1	8:00	3,200	N1	18:00	3,200
D2	9:00	3,200	N2	23:00	3,200
D3	11:00	3,200	N3	03:00	3,200
D4	13:00	3,200	N4	06:00	3,200
D5	15:00	3,200			
D6	17:00	3,200			

Table 2 Number of data by class

Class ID	Class	Count
0	Car	8,400
1	Ban	5,100
2	Bus	6,500
3	Taxi	6,200
4	Truck	5,800

학습 및 검증을 위한 데이터 집합은 차량의 전면 영상과 함께 후면 영상을 포함하고 있으며 시간대는 주간 6구간(D1~D6), 야간 4구간(N1~N4)으로 나누었다. 각 구간별 3,200장의 영상을 구축하여 전체 32,000장의 데이터 집합을 구성하였다. Table 1은 시간대 분류 및 데이터의 개수를 보여주고 있다.

32,000장의 데이터 집합은 학습 데이터(Training data) 28,800장과 검증 데이터(Validation data) 3,200장으로 구성하였다. 또한, 차량 검출 및 차량의 종류를 5종으로 동시에 분류하기 위해서 5개의 클래스로 분류하였으며 Table 2와 같다.

3.2 딥 러닝 모델

본 논문에서 사용하는 딥 러닝 프레임워크는 오픈소스 기반의 YOLO를 사용하고 있다.[6,7] YOLO는 이미지 내의 바운딩 박스(Bounding box)와 클래스 확률(Class probability)을 한 개의 회귀 문제(Single regression problem)로 간주하여, 이미지를 한 번 보는 것으로 객체의 종류와 위치를 추측하는 네트워크 모델(Single convolutional network)을 제시하였다.

YOLO에서 제시하는 Darknet 네트워크는 Fig. 2에서 설명하는 Inception 모델[8]과 이를 기반으로 하는 GoogleNet을 응용하여 높은 정확도를 가지면서 동시에 매우 빠른 처리가 가능하다.

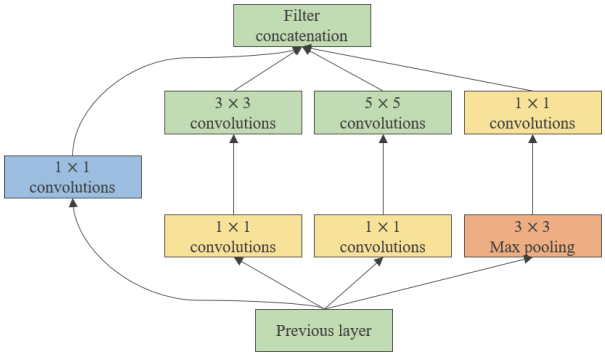

Fig. 2 Inception model with dimension reductions

Inception 모델은 다른 네트워크 모델들이 7×7 등 하나의 Convolution 필터를 사용하는 것과 달리 3×3 및 1×1의 작은 Convolution 필터 여러 개를 한 층으로 설계하였다. 이를 통해서, 네트워크 사이의 연결을 줄이면서(Sparcity) 행렬 연산에서는 최대한 높은 집적도(Density)를 가지는 네트워크 모델을 설계함으로써 다른 딥러닝 프레임워크에 비해 상대적으로 빠른 알고리즘 처리 속도를 나타내고 있다.

본 논문에서 사용한 YOLO v3는 이전의 버전에서 발생하는 작은 객체의 검출 정확도가 떨어지는 문제를 해결하기 위해서 Darknet-53의 개선된 네트워크 모델을 제시하였다.[9]

YOLO v3의 주요 특징은 다음과 같다.

3.2.1 Bounding Box Prediction and Cost Calculation

1) 각 Bounding box는 Logistic regression을 사용하여 Objectness score를 예측하며 앵크 박스가 다른 객체보다 Ground truth 객체와 더 많이 겹치면 Objectness score 1을 할당한다.
2) 앵커 박스가 기저 임계치(Default 0.5)보다 큰 값을 가

지면 0을 할당하고 앵커 박스가 할당되지 않으면 0을 할당한다.

3) YOLO v3는 Feature Pyramid Network(FPN)와 비슷한 방법으로 3개의 서로 다른 스케일(Scale)에서 특징 맵을 계산한다.

4) 마지막 Feature map layer의 이전 두 번째 layer를 2배 업 샘플(Up sample)하여 특징 맵을 계산하고 이를 반복함으로써 3개의 서로 다른 특징 맵을 계산한다.

3.2.2 Feature Extractor

1) YOLO v3는 YOLO v2에서 제시한 Convolutional Neural Network(CNN) 19계층보다 더 높은 정확도를 위해 53계층의 Darknet-53 네트워크 모델을 제시하고 있다.

2) Darknet-53은 ResNet의 Residual network와 유사한 네트워크 구조를 가지면서 동시에 계산량을 줄이기 위해서 3×3 및 1×1 필터를 설계하여 ResNet-152 네트워크와 비교하여 2배 빠른 계산 속도를 보여주고 있다. Fig. 3은 Darknet-53의 주요 계층(Layer)에서 Residual network와 필터를 이용하는 것을 보여주고 있다.

	Type	Filters	Size	Output
	Convolution	32	3×3	256×256
	Convolution	64	3×3/2	128×128
1×	Convolution	32	1×1	
	Convolution	64	3×3	
	Residual			128×128
	Convolution	128	3×3/2	64×64
2×	Convolution	64	1×1	
	Convolution	128	3×3	
	Residual			64×64
	Convolution	256	3×3/2	32×32
8×	Convolution	128	1×1	
	Convolution	256	3×3	
	Residual			32×32
	Convolution	512	3×3/2	16×16
8×	Convolution	256	1×1	
	Convolution	512	3×3	
	Residual			16×16
	Convolution	1,024	3×3/2	8×8
4×	Convolution	512	1×1	
	Convolution	1,024	3×3	
	Residual			8×8
	Avgpool		Global	
	Connected		1,000	
	Softmax			

Fig. 3 Overall architecture of the Darknet-53 network

3.3 데이터 학습

데이터 학습은 64개의 이미지를 하나의 미니배치(Mini-batch)로 정의하였으며 Momentum 학습 방법을 적용하여 500,200회를 학습하였다. 학습 결과는 모델이 예측한 답과 실제 정답의 관계를 정의하는 분류성능 평가지표를 이용하였으며 Table 3은 머신 러닝(Machine learning)에서 주로 사용하는 분류성능 평가지표인 오차행렬(Confusion matrix) 모델을 보여주고 있다.

재현율(Recall)은 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율을 의미하며 식 (1)과 같이 정의한다.

$$Recall = \frac{TP}{TP + FN} \tag{1}$$

정밀도(Precision)는 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율을 의미하여 식 (2)와 같이 정의한다.

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

F1-Score는 정밀도와 재현율의 조화평균을 의미하며 식 (3)과 같이 정의한다.

$$F1 - Score = 2 \times \frac{\text{정밀도} \times \text{재현율}}{\text{정밀도} + \text{재현율}} \tag{3}$$

데이터의 학습 결과는 재현율 96 %, F1-Score 97 % 그리고, 평균 정밀도는 98.33 %의 높은 정확도를 보였다. 이를 통해서, 학습 데이터에 대한 객체 라벨링의 정밀도가 높으며 데이터의 학습이 잘 되었다는 것을 알 수 있다. 각 클래스의 정밀도는 Table 4와 같다.

Table 3 Confusion matrix model

		Actual answer	
		True	False
Classification result	True	True positive	False positive
	False	False negative	True negative

Table 4 Average precision of each class

Name	AP(%)
Car	98.15
Ban	95.65
Bus	100
Taxi	97.31
Truck	99.35

3.4 룩업테이블(LUT)을 이용한 알고리즘 처리 속도 개선

교통 감시용 카메라에서 획득한 데이터를 YOLO v3를 이용하여 차량을 검출하기 위해서는 Fig. 4의 데이터 전처리 과정이 필요하다.

여기서, YUV 데이터를 RGB로 변화하는 (1)의 과정 그리고, RGB 데이터를 0.0~1.0으로 정규화하는 (2)의 과정은 많은 계산량이 필요하다. 본 논문에서는 계산량 감소를 위해서 첫 번째, 룩업테이블(LUT)을 이용하여 YUV 데이터를 RGB로 변환하여 연산량을 줄였으며 변환 공식[10]은 식 (4)를 적용하였다.

$$\begin{aligned} R &= 1.164(Y-16)+1.596(V-128) \\ G &= 1.164(Y-16)-0.813(V-128)- \\ &\quad 0.391(U-128) \\ B &= 1.164(Y-16)+2.018(U-128) \end{aligned} \tag{4}$$

두 번째, RGB 데이터의 각 픽셀 값을 255로 나누는 정규화 연산을 LUT로 미리 계산하였다.

4. 실험결과

4.1 실험 방법

실험 데이터는 학습 데이터를 획득한 교차로 12개소에 설치된 교통 감시 카메라로부터 신규 20개의 동영상을 녹화하여 제작하였다. 제작된 동영상은 카메라별로 약 1시간씩 녹화되었으며 20개 동영상의 전체 크기는 15.1 Gb이다. 또한, 실험에 사용된 컴퓨터는 Intel i7-7700 @4.20 GHz CPU, 64.0 GB 메모리 그리고, NVIDIA GTX 1080 Ti GPU로 구성하였다.

실험 방법은 녹화된 테스트 동영상으로부터 차량을 계수하는 프로그램을 제작하여 실제 통과한 차량과 비교하는 방법으로 진행하였다. 제작한 프로그램은 동영상 데이터로부터 Fig. 4에서 설명하는 전처리 알고리즘 처리 및 YOLO v3의 Darknet-53에서 학습된 웨이트(Weights) 파일을 이용하여 차량을 검출하는 기능을 포함하고 있다. 또한, 연속된 프레임에서 동일한 차량을 판단하기 위해 현재 프레임에서 검출된 차량의 중심점과 이전 프레임에서 검출된 차량의 중심점 사이의 x축 및 y축의 거리가 가장 가까운 차량을 동일한 차량으로 판단하는 추적 알고리즘을 구현하였다. Photo. 4는 제작한 프로그램을 이용하여 차량 검출 및 추적과 관련된 모습을 보여주고 있다.

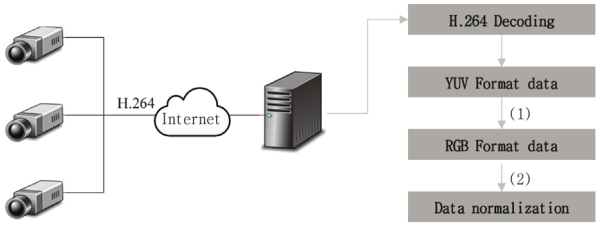

Fig. 4 Data processing flow in YOLO v3

4.2 실험 결과

YOLO v3의 Darknet-53 네트워크 모델을 이용하여 먼 거리에 있는 작은 크기의 차량 검출 성능을 확인하였다. Photo. 5에서 보듯이, YOLO v2의 Darknet-19 네트워크 모델을 적용한 Photo. 2와 비교하여 먼 거리에 있는 작은 크기의 차량에 대해서 검출 성능이 향상되었음을 확인할 수 있었다.

교차로를 통과하는 차량의 대수는 사전에 육안으로 검색하여 계수한 값을 기준으로 Photo. 4에서 설명한 프로그램의 계수 값을 비교하여 평가하였다. 평가 방법은 도로교통공단에서 제시하는 평가 기준을 준용하였으며 Table 5와 같이 오전, 오후, 야간으로 나누어 분류하였다.

실험 결과 YOLO v3의 Darknet-53은 YOLO v2의 Darknet-19에 비해 정확도가 평균 4.6 % 향상되었으며 전처리 과정에서 LUT 알고리즘을 적용함으로써 하나의 프레임에서 5 ms의 처리 속도가 감소되었다. Table 6은 YOLO v2와 v3의 차량 계수 결과를 보여주고 있다.

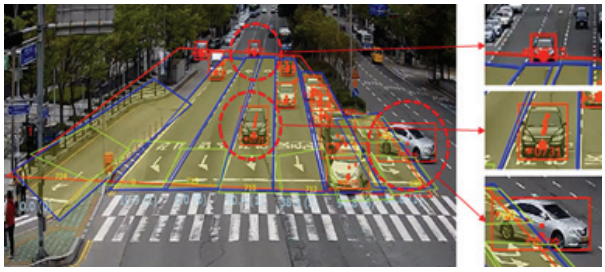

Photo. 4 Vehicle detection and tracking program

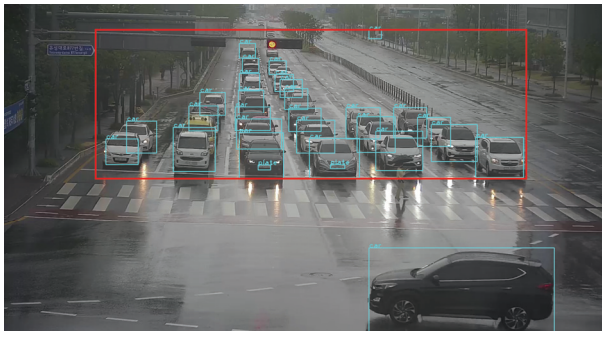

Photo. 5 Improved detection performance of small vehicles over long distance in YOLO v3

Table 5 Vehicle counting evaluation criteria

	Evaluation time	Sample count	Data collection time (min)
Day	09:00-12:00	≥120	≥60
Afternoon	13:00-18:00	≥120	≥60
Night	19:00-22:00	≥120	≥60

Table 6 Vehicle counting evaluation result

	Number of vehicles	Detected vehicle count			
		YOLO v2		YOLO v3	
		Detction count	Detection rate	Detection count	Detection rate
Day	2,196	1,987	90.5	2,088	95.1
Afternoon	1,884	1,739	92.3	1,826	96.9
Night	1,642	1,287	78.4	1,361	82.9

5. 결 론

본 논문은 딥 러닝을 이용하여 교차로를 통행하는 차량의 정확한 계수 및 대기열 계산을 위해서 먼 거리의 작은 차량에 대한 정확도 높은 검출 방법을 제시하고 있다. 기존의 YOLO v2의 Darknet-19 네트워크와 비교하여 YOLO v3의 Darknet-53 네트워크에서 차량 검출 정확도가 4.6 % 향상되는 것을 확인하였다.

그러나 컴퓨터비전 기술을 이용하여 신뢰도 높은 차량 계수 데이터를 안정적으로 제공하기 위해서는 눈, 비, 안개 등 다양한 자연환경과 특히, 야간에도 높은 정확도를 가지는 차량 검출 기술이 지속적으로 연구되어야 할 것으로 판단된다.

References

1) B. G. Han, J. T. Lee, K. T. Lim and Y. Chung, “Real-time License Plate Detection in High Resolution Videos Using Fastest Available Cascade Classifier and Core Patterns,” ETRI Journal, Vol.37, No.2, pp.251-261, 2015.
2) K. Kim, P. Kim, K. Lim, Y. Chung, Y. Song, S. Lee and D. Choi, “Vehicle Color Recognition via Representative Color Region Extraction and Convolutional Neural Network,” Proceedings of the 10th International Conference on Ubiquitous and Future Networks (ICUFN), pp.89-94, 2018.
3) L. Wen, D. Du, Z. Cai, Z. Lei, M. Chang, H. Qi, J. Lim, M. Yang and S. Lyu, “A New Benchmark and Protocal for Multi-object Detection and Tracking,” https://arxiv.org/abs/1511.04136, 2015.
4) K. Kim, P. Kim, Y. Chung and D. Choi, “Multi-Scale Detector for Accurate Vehicle Detection in Traffic Surveillance Data,” IEEE Access, Vol.7, pp.78311-78319, 2019.
5) T. Lee, K. Kim, K. Yun, K. Kim and D. Choi, “A Method of Counting Vehicle and Pedestrian Using Deep Learning Based on CCTV,” Journal of the Korean Institute of Intelligent Systems, Vol.28, No.3, pp.219-224, 2018.
6) J. Redmon, S. Divvala, R. Girshick and A. Farhadi, “You Only Look Once: Unified, Real-Time Object Detection,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.779-788, 2016.
7) J. Redmon and A. Farhadi, “YOLO9000: Better, Faster, Stronger,” Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp.6517-6525, 2017.
8) C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, “Going Deeper with Convolutions,” https://arxiv.org/abs/1409.4842, 2014.
9) J. Redmon and A. Farhadi, “YOLO v3: An Incremental Improvement,” https://arxiv.org/abs/1804.02767, 2018.
10) K. Jack, Video Demystifice-A Handbook for the Digital Engineer, 4th Edn., Newnes, pp.18-19, 2005.