

Yolov5와 opencv를 사용한 차량 교통량 및 속도 측정

이민섭^o, 우지영*, 남윤영*

^o순천향대학교대학원 ICT융합학과,

*순천향대학교대학원 ICT융합학과

e-mail: lys1978000@gmail.com^o, jywoo@sch.ac.kr*, ynam@sch.ac.kr*

Measurement of vehicle traffic volume and velocity using Yolov5 and opencv

Minseop Lee^o, Jiyoung Woo*, Yunyoung Nam*

^oDept. of ICT Convergence, SoonChunHyang University,

*Dept. of ICT Convergence, SoonChunHyang University

● 요약 ●

본 논문에서는 Yolov5와 Deepsort를 사용한 Tracking by detection을 구현하여 특정 영역을 통과하는 차량의 수를 집계하고, 각 차량의 추정속도를 계산하는 시스템을 구현한다. 실시간 객체 탐지 기능을 수행하는 Yolov5 모델의 학습에는 Kaggle의 개방 데이터인 '도요타 자동차 이미지'를 사용한다. 이미지 크기 640*640, 배치사이즈 16, Early stopping 플래그를 사용하여 학습했을때, Yolov5의 객체 탐지 성능은 정확도 98%, 정밀도 0.961, mAP 0.72을 보여주었다.

키워드: 객체 탐지(object detection), 객체 추적(object tracking)

I. Introduction

국내에서는 인도, 자전거도로, 국도, 고속화도로, 고속도로 등 국민들의 이동권을 보장하기 위한 도로 인프라가 구축되어 있고, 지속적으로 확대하고 있다. 그중 차마의 통행을 위한 국도, 고속화도로, 고속도로 등은 통행자와 그 주변의 안전을 위해 속도의 상한을 제한하고 있고, 고정식 과속 단속 카메라, 이동식 과속 단속 카메라, 구간 단속 카메라를 통해 과속 차량을 단속하여 벌점 및 범칙금을 부여하고 있다. 특히, 2019년 '도로교통법'과 '특정범죄가중처벌등에관한법률 일부개정법률안' 통칭 '민식이법'이 발의된 이후 아산지역 스텝존 내 과속 단속 카메라의 설치율이 2019년 6%에서 2021년 52%로 상승하였고[1], 이후로도 전국의 스텝존에서 지속적으로 과속 단속 카메라를 설치하고 있다. 과속 단속 카메라의 가격은 대당 4,200만원 가량으로 상당히 고가이므로 이를 설치함에있어 많은양의 예산이 사용되고 있다. 따라서, 고가의 센서가 사용된 과속 단속 카메라 대신 영상처리를 통해 차량의 통행량 및 속도를 계산할수 있도록 이 시스템을 구현했다.

II. Preliminaries & The Proposed Scheme

1. 시스템 적용범위 및 개요도

본 시스템은 고속화도로, 고속도로 대비 상대적으로 차량의 통행속도가 낮은 국도, 그중에서도 스텝존에서 고정식 과속 단속 카메라로서 기능할 수 있는 시스템의 개발을 목적으로 하였다. Fig. 1. 는 시스템 개요도이다.



Fig. 1. System Workflow

2. 객체 탐지

본 시스템에서 객체 탐지는 Yolov5 모델을 사용하여 수행하였다. Kaggle의 개방 데이터인 '도요타 자동차 이미지'를 학습데이터로 사용하였고[2], 학습이미지에서 목표표하는 객체인 차량이 위치하는 좌표를 주석하는 작업은 데이터 정제 기능을 제공하는 사이트인 Roboflow에서 수행하였다[3].

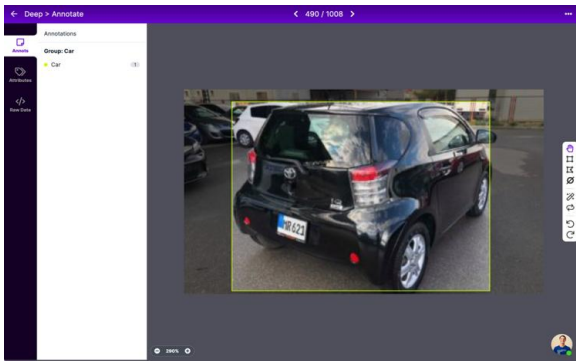


Fig. 2. Image Annotation

객체 탐지 모델의 학습 및 검증은 Google Colab GPU 환경에서 수행되었다. 이미지 크기 640*640, 배치사이즈 16 상태에서 모델의 크기에 차이를 두고 5n(nano), 5s(small), 5x(xlarge) 세 모델의 객체 감지 성능을 측정하였다. 측정 결과는 Table 1. 과 같다.

Table 1. Detection performance

Model	Accuracy	Precision	mAP
5n	0.95	0.91	0.51
5s	0.98	0.96	0.72
5x	0.99	0.96	0.77

3. 객체 추적

객체 탐지에서 탐지된 객체를 추적하는 객체 추적은 칼만필터와 헝가리안 메소드를 사용하는 Deepsort를 적용하여 구현하였다. Deepsort의 적용은 Google Colab GPU 환경에서 수행되었다. Table 1. 의 세 모델에 Deepsort를 적용하였을때의 FPS(Frame Per Second)는 Table 2. 과 같다.

Table 2. Tracking performance

Model	FPS
5n	25
5s	18
5x	9

4. 교통량 및 속도 측정

객체 탐지와 객체 추적에서의 성능을 종합적으로 평가하여, 5s모델을 사용하는것으로 결정하였다. Opencv를 사용하여 객체들의 중심좌표가 임의의 선을 통과하는 횟수를 집계하여 교통량 파악이 가능하게 구현하였고, 객체의 속도는 Fig. 3. 과 같이 계산하였다.

$$\text{Velocity} = \text{Actual Distance} / \text{Duration} * 3.6 * \alpha$$

Fig. 3. Velocity calculate formula

위 식에서 Actual Distance는 설정 구간 사이의 실측 거리를 의미한다. Duration은 설정 구간을 통과하는데 소요된 프레임의 수를

FPS로 나눈 값을 의미한다. α 는 결과로 출력될 영상의 FPS를 객체 추적중의 FPS로 나눈값으로, 기기의 성능에 따라서 Velocity의 값이 다르게 나오는것을 방지하기 위한 값이다.

5. 시스템 구동 화면



Fig. 4. Result Screen of the System

Fig 4. 는 본 시스템의 구동 화면이다. 두 빨간 선분 사이의 구간이 시스템의 설정 구간이다. 좌측 선분을 통과하는 차량 수로 교통량을 파악하고, 좌측과 우측 선분을 통과하는 시점의 프레임 인덱스의 차이를 활용하여 속도를 계산한다.

III. Conclusions

본 시스템은 상대적으로 차량의 속도가 낮은 스쿨존에서 고가의 과속 단속 카메라를 대신하여 영상처리를 통한 차량의 속도 측정 기능을 구현하는데 목적이 있다. 하지만, 공공데이터포털에서 제공하는 국내 차량 이미지의 경우 번호판에 마스킹이 되어있어 과속 차량의 번호판을 OCR기능을 통해 수집하는 기능을 구현하지 못하였다. 향후 데이터를 수집하여 번호판 OCR을 추가하는 등 시스템의 기능을 향상 시키고자 한다.

ACKNOWLEDGEMENT

본 연구는 교육부의 지자체-대학협력기반지역혁신사업(1345341784)의 지원을 받아 수행되었음

REFERENCES

[1] ASANDESK, <http://www.asandesk.co.kr/news/articleView.html?idxno=15018>
 [2] <https://www.kaggle.com/datasets/occulainsights/toyota-car-s-over-20k-labeled-images>
 [3] <https://roboflow.com/>.