

# 딥러닝 기반의 실시간 데이터셋 생성 시스템

장호혁, 탁현준, 이소희, 이영섭  
 인천대학교 임베디드시스템공학과  
 e-mail : {jnhghk, xkruswns, 201301675, YSL}@inu.ac.kr

## A Real-time system for dataset generation based on Deep Learning

Hohyeok Jang, Hyunjun Tak, Sohee Lee, Young-Sup Lee  
 Dept. of Embedded Systems Engineering, Incheon National University

### 요 약

본 논문은 도로에서의 객체탐지를 위한 딥러닝(deep learning) 데이터셋을 자동으로 생성, 분류하는 시스템을 제안한다. 시스템의 작동 과정은 크게 두 가지이다. 먼저 딥러닝을 활용하여 촬영된 영상에 존재하는 객체를 검출한다. 이때, 실시간으로 하는 방법과 레코딩된 영상을 다루는 방법 두가지가 있다. 다음으로 검출된 객체 중 예측 값(score)가 임계치 이상인 객체의 위치와 종류를 파일로 저장한다. 이 시스템은 차량 전방 카메라 위치에 장착된 웹캠을 이용해 영상을 취득하고 임베디드 보드인 TX2 board를 이용해 데이터 셋을 생성한다. 매킨토시의 image labeler app과 비교를 통해 보다 적은 시간비용으로 데이터셋을 생성해 냈음을 확인 하였다.

### 1. 서론<sup>1</sup>

매년 열리는 ‘ILSVRC (ImageNet Large Scale Visual Recognition Challenge)’는 객체 인식/분류 기술 성능을 한 눈에 살펴볼 수 있는 대회이다.

2012년 Alex가 제안한 GPU를 이용한 DCNN (Deep Convolutional Neural Network) 기반의 AlexNet의 등장으로 그전까지 인간의 오차 수준 5 % 미만인 28.2 %에 그쳤던 오차율이 16.7 %로 떨어졌다 [1]. 2015 년 Resnet은 오차율 3.6 % 을 기록하며 객체 인식 분야에서 인간 수준을 넘어서기에 이르렀다. Alexnet이 나타난 이후 딥러닝의 개발, 빅데이터의 활성화, 하드웨어의 발전에 힘입어 처리 가능한 문제의 범위가 급속도로 넓어지면서 여러 가지 분야에 접목되는 중이다 [2].

객체 탐지를 위한 네트워크의 학습은 이미지와 객체 위치, 객체의 종류 포함한 데이터셋을 필요로 한다. 그림 1과 같은 매킨토시에서 제공하는 application인 image labeler는 객체 탐지를 위한 데이터셋을 생성하는데 사용된다. 하지만 이와 같이 사람이 직접 데이터셋을 생성하는 것은 많은 시간이 소요된다.

그렇기 때문에 본 논문에서는 사람이 일일이 라벨링하는 시간 비용을 줄이고자 실시간 데이터셋 생성 시스템을 제안한다. 이 시스템은 텐서플로우 (Tensor-flow)에서 제공하는 SSD (Single Shot multibox Detector) 모델을 적용하여 딥러닝 사용자가 원하는 데이터셋을 생성 및 제공한다.

\* 이 논문은 2018 한이음 ICT 멘토링 프로젝트의 연구비 지원을 받은 결과물입니다.

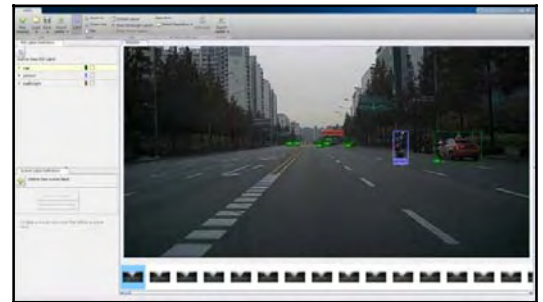


그림 1. Matlab image labeler application GUI

### 2. 본론

#### 2-1 설계 및 구조

본 논문에서 실시간 데이터셋 생성 시스템의 전체적인 흐름도는 그림 2와 같이 구성한다.

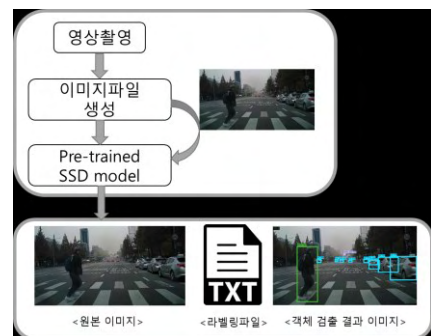


그림 2. 설계 및 구조

블랙박스 역할을 하기 위해 데이터 저장 및 촬영의 용이성을 위해 자동초점 기능이 있는 로지텍社의 웹캠 (HD ProWebcam C920r)을 라즈베리파이 3에 연결하여 영상을 취득한다. 이때, 640 \* 480 사이즈의 영상을 Python 과 Opencv 라이브러리를 이용하여 실시간으로 영상을 캡처한다. 이후 jpg 포맷을 가진 이미지파일로 저장하고, 딥러닝 학습 및 영상 처리를 위하여 엔비디아社의 jetson TX2 보드를 사용하여 레코딩된 영상을 TX2 보드에 입력하여 학습된 모델을 거치는데, SSD의 기본 구조는 그림 3와 같다 [3].

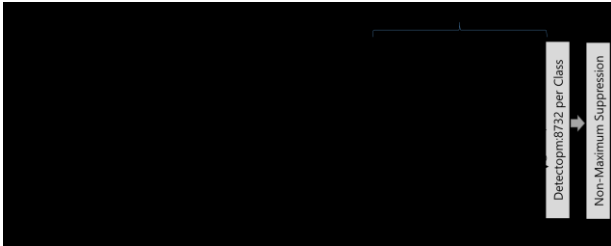


그림 3. SSD 기본 구조

SSD에서는 이미지 특징 추출기로 VGG16을 사용한다. 중간 계층에서는 각각의 classification과 box regression 작업을 같이 수행한다. 최종적으로 중간계층에서 추출한 classification과 box regression 중 가장 신뢰성이 높은 아웃풋에 대해서 최종 결과를 내놓는다 [3].

Jpg 포맷의 원본이미지와 검출 결과 이미지, 그리고 txt 포맷의 라벨을 생성한다. 이때, 텍스트파일에는 검출된 객체의 이름과 박스 위치에 대한 정보가 있다. 박스 위치는 y 좌표의 최솟값, x 좌표의 최솟값, y 좌표의 최댓값, x 좌표의 최댓값으로 설정한다.

네트워크를 거쳐 나온 아웃풋은 원본 이미지와, 검출된 객체, 그 객체에 대한 좌표, 객체가 검출된 이미지로 나뉜다. 이때, 아웃풋에서 검출된 데이터는 학습에 사용될 것이기 때문에 신뢰성이 있어야 한다. 그렇기 때문에 임계값을 설정하였다. 사용자가 설정한 임계값 이상의 정확도를 가진 결과에 대해서만 파일로 저장한다. 그림 4는 임계값을 80%로 설정 했을 때 검출된 객체에 대한 이미지 이다.

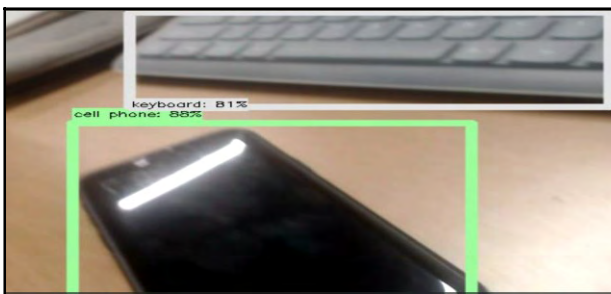


그림 4. 임계값 설정의 예

## 2-2 라벨링 (labeling)

레이블링 또는 라벨링 (labeling)이란 이진화 한 이미지에서 형태를 알아보기 위해 같은 픽셀 값들끼리 그룹화 하여 번호를 매긴 것을 의미한다.

30 fps 인 영상은 1 분이 넘어가면 대개 ‘천’ 단위의 프레임 을 가지고 있다. 방대한 양의 이미지파일을 일일이 수작업을 수행하는 것은 많은 노력이 필요하다. 때문에 본 연구에서는 방대한 양의 이미지 파일의 라벨링 작업의 자동화를 주 목적으로 하고 있다.

## 2-3 GUI

오브젝트 분류는 그림 과 같이 미리 학습된 모델에서 객체의 이름과 마크형식으로 나타나도록 진행한다. 텐서플로우에서 제공하는 SSD 모델에는 90 가지의 객체가 학습 되어있다. 그중 사용자가 원하는 객체에 대해서 설정하면 네트워크를 거쳐서 검출된 객체에 대한 정보를 우측 화면에 출력한다.

파일에서 기본 영상을 불러 온 뒤 학습된 영상으로 출력 된 것을 그림 5와 같이 GUI 상에서 확인할 수 있다.

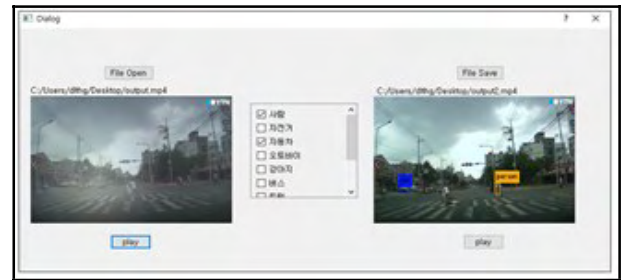


그림 5. GUI 결과

## 3. 실험 결과

표 1은 실시간 데이터셋 생성 시스템과 Matlab의 image labeler application을 사용하였을 때 걸리는 시간을 보여준다. 이미지 100 장을 기준으로 하였으며, image labeler application의 경우 사용자의 숙련도에 따라 차이가 있을 수 있다. 3 가지의 라벨을 가지고 라벨링을 하였을 때 약 61 분이 소요되었다. 도로주행 영상을 프레임 단위로 쪼개 이미지로 만든다면 1000 장 이상의 이미지에 대해서 라벨링 작업을 실시하게 된다. 1000 장을 모두 사람이 하게 된다면 10 시간 이상이 걸릴 것으로 예상된다.

표 1. 100 장 기준 라벨링 시간 비교

	소요 시간 (s)
Image labeler app	3681
Proposed system (TX2)	132
Proposed system (GTX1060)	7.28

그림 6의 (a), (b)는 각각 TX2 보드와, GTX1060에서 Proposed system을 적용하여 소요된 시간이다.

Time to first image는 이미지를 받아들이기 위해 준비하는 시간이고, Elapsed time은 첫 이미지부터 마지막 이미지까지 처리하는데 걸린 총 시간이다. 프로그램이 소요하는 시간은 두 시간을 합친 시간으로 한다.

```
Time to first image:39.0070838928
Elapsed time: 93.379275 seconds.
```

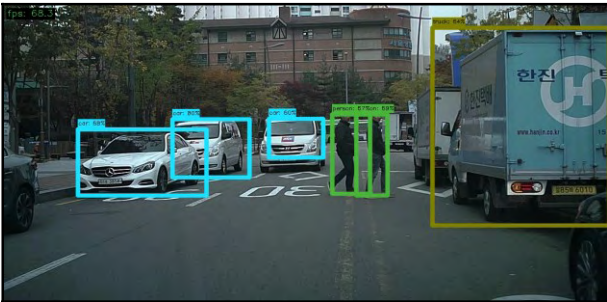
(a) TX2

```
Time to first image:2.97419810295
Elapsed time: 4.311409 seconds.
```

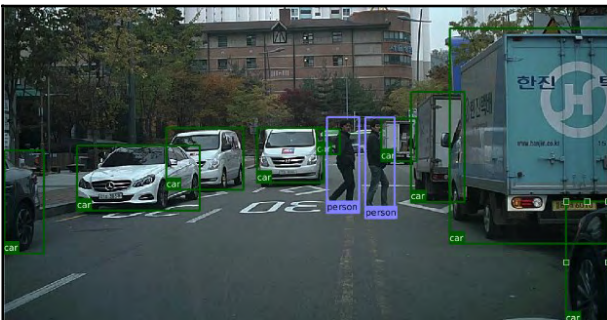
(b) GTX1060

그림6. 시스템 적용 시 소요 시간 비교

속도면에서는 GTX1060 > TX2 > 사람 순이었고, TX2 보드에서는 약 1 초에 1 장 미만을 처리하는 것을 알 수 있었다. 객체 검출 결과와 image labeler application을 사용했을 때 결과 비교는 그림 7 과 같다. 그림 7의 (a) 와 (b) 는 같은 이미지에 대해서 각각 객체 탐지와 Matlab 의 image labeler application 을 사용한 결과이다. 탐지된 객체 수는 6 개였고, 사람은 10 개의 객체를 라벨링 할 수 있었다. 아래 실험 결과와 같이 Proposed system 은 우측 두 대의 트럭중 앞에있는 차량만 인식을 하였고, 사람들 뒤쪽으로 보이는 버스, 차량에 대해서도 검출을 하지 못하였다.



(a) Proposed system



(b) Image labeler

그림7. 라벨링 결과 비교

#### 4. 결론 및 향후 계획

딥러닝을 이용한 실시간 데이터셋 생성 시스템은 대체로 정확하고 실시간의 데이터셋 생성 결과를 보여주었다. 하지만, 실험결과를 보면 사람보다 정확하다고 할 수 없었다.

우리의 연구는 도로 위의 객체들을 예로 들었지만, 도로주행뿐만 아니라 다른 분야에서도 활용한다해도 데이터셋을 생성하는데 시간 비용을 크게 절약할 수 있을 것이라 예상한다. 향후 양질의 데이터셋을 추가하여 현재 약점인 작은 객체에 대해서도 강인하게 동작할 수 있도록 보완할 계획이다.

#### 참고문헌

- [1] 유병인, 황원준, 한승주, 이선민, 김정배, & 한재준. (2015). 인간 수준에 근접한 딥러닝 기반 영상 인식의 동향. 정보과학회지, 33 (9), 32-41.
- [2] 문성은, 장수범, 이정혁, & 이종석. (2016). 기계학습 및 딥러닝 기술동향. 한국통신학회지 (정보와통신), 33 (10), 49-56.
- [3] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C. Y., & Berg, A. C. (2016). Ssd: Single shot multibox detector. In European conference on computer vision, 9905, 21-37.