물체인식 딥러닝 모델 구성을 위한 파이썬 기반의 Annotation 툴 개발

임송원, *박구만
서울과학기술대학교
technocubic1003@naver.com, *gmpark@seoultech.ac.kr

Development of Python-based Annotation Tool Program for Constructing Object Recognition Deep-Learning Model

Songwon Lim *Gooman Park
Seoul National University of Science and Technology

요 약

본 논문에서는 물체인식 딥러닝 모델 생성에 필요한 라벨링(Labeling)과정에서 사용자가 다양한 기능을 활용하여 효과적인 학습 데이터를 구성할 수 있는 GUI 프로그램을 구현했다. 프로그램의 인터페이스는 파이썬 기반의 GUI 모듈인 Tkinter 를 활용하여, 실시간으로 이미지 데이터를 수집할 수 있는 크롤링(Crawling)기능과 미리 학습된 Retinanet 을 통해 이미지 데이터를 인식함으로써 자동으로 주석(Annotation) 과정을 수행할 수 있는 기능을 구성했다. 또한, 수집한 이미지 데이터를 다양한 효과와 노이즈, 변형 등으로 Augmentation 기능을 추가함으로써, 사용자가 모델을 학습하기 위한 데이터 전처리 단계를 하나의 GUI 프로그램에서 수행할 수 있도록 했다. 또한 사용자가 직접 학습한 모델을 추정 모델(Inference Model)로 변환하여 프로그램에 입력할 수 있도록 설계한다.

1. 서론

최근 컴퓨터 비전 인식 문제는 다양한 딥러닝 모델을 활용하여 여러 분야로 응용되면서, 각 분야에 맞는 개별적인 신경망(Customizing Model)을 구성하는 중요성은 커지고 있다. 따라서 개발자 및 연구자는 기존의 머신러닝 방식의 학습 데이터를 구성하기 위한 주석 과정(Annotation)은 필수적이며, 학습 데이터의 양에 따라 필요한 노동력과 시간의 규모는 커질 수 있다.

이에 본 논문에서는 물체인식 딥러닝 모델을 구성하는데 필요한 라벨링(labeling)과정에서 사용자가 다양한 기능을 활용하여 효과적인 학습 데이터를 구성할 수 있는 GUI 프로그램을 구현했다.

기존의 프로그램에서는 특정 네트워크만 학습이 가능한 하위의 프로그램이거나, 주석 과정에서 사용자가 이미지 데이터를 일일이 박스 형태로 그려야 하는 작업이었다. 그러나 본 프로그램에서는 절대값 좌표 계산뿐 아니라 YOLO 네트워크 형식[1]의 좌표값 계산도 가능하다. 또한, 기존의 학습된 딥러닝 모델을 기반으로 사용자가 버튼 클릭만으로 이미지 데이터를 인식하고, 라벨링 데이터를 구성할 수 있도록 했다. 또한 사용자가 학습한 모델을 추론 모델(Inference Model)로 변환하여 GUI 프로그램에 연동하도록 설계한다.

본 논문의 구성은 다음과 같다. 2 절에서는 전체 GUI 프로그램에 대한 구조도를 살펴보고, 3 절에서는 각 기능의 구현 방법을 살펴본다. 4 절은 실험 및 구현한 프로그램 활용에 대한 토의를 하고, 5 절에서는 결론을 맺는다.

2. GUI 프로그램 구성도

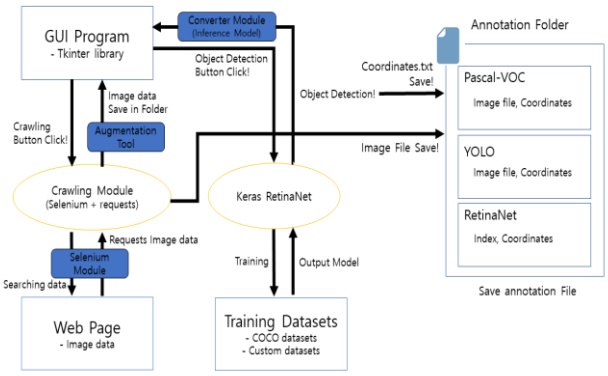

그림 1. GUI 프로그램 구성도

본 프로그램을 구성하는 기능은 위 그림과 같이, 각각의 기능을 모듈화하여, 사용자가 버튼 조작으로 기능을 작동할 수 있도록 했다. 각 모듈에는 셀레니움(Selenium)을 통해, 어떠한 웹페이지 구조에서도 이미지 데이터를 저장할 수 있는 크롤링(Crawling) 기능을 추가했으며, 수집한 데이터의 양을 증가시키기 위한 변형 및 효과 등을 적용할 수 있는 Augmentation 기능도 구성했다. 또한, 미리 학습된 RetinaNet 을 GUI 프로그램에 연동하여 이미지 데이터를 인식할 수 있도록 했다.

3. 프로그램 구현 및 RetinaNet 학습 모델

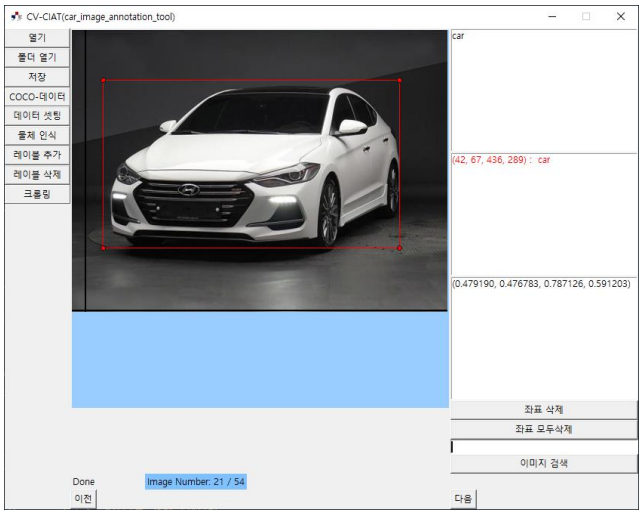

그림 2. 구현한 GUI 툴 프로그램

본 프로그램의 인터페이스는 그림 2 와 같은 형태로 파이썬의 GUI 라이브러리인 Tkinter 를 활용하여 구현했다. 프로그램과 같이 이미지 파일을 불러올 경우, 이미지들은 각각 다른 해상도를 가지기 때문에, 이미지가 잘리거나 확대되어 현시되는 문제점이 있다. 이에 Bicubic 보간법(interpolation)을 활용하여 이미지의 너비나 높이가 한쪽이 더 큰 경우로 나누어 적용했다. Bicubic 보간법은 인접하는 픽셀에 대해 선형 보간하는 방법으로, 이미지가 균형있는 비율로 프로그램에 현시될 수 있다.

그림 3. 중고차 사이트 크롤링 기능 구현

또한, 사용자가 원하는 이미지 데이터를 저장할 수 있도록 크롤링 기능을 프로그램에 추가하였다. 크롤링 기능은 크게 두 가지의 기능으로, 사용자가 키워드를 검색하면, 해당 키워드의 이미지를 구글 사이트에서 수집하는 기능이다. 두 번째는 그림 3 과 같이 다양한 차량 이미지를 수집할 수 있는 기능으로, 여러 중고차 사이트에 접속하여, 사용자가 원하는 차량의 이미지를 저장할 수 있다. 크롤러는 어떠한 웹페이지 구조에도 접근이 용이한 셀레니움과 파이썬의 requests 모듈로 구현했다. 특히, 셀레니움은 브라우저(Browser)를 자동화 형태로 조작할 수 있기 때문에 데이터에 접근하는 속도는 조금 느리더라도, 접근이 어려운 데이터에서 수집에 용이하다.

그림 2 와 같이, 구현한 프로그램에서 이미지 데이터를 마킹한 것은 프로그램의 '물체 인식' 이라는 버튼을 클릭하면서, 툴(Tool)은 해당 물체를 인식하여 박스 형태로 나타낸 것이다. 이것이 가능한 이유는 물체인식 분야에서 뛰어난 RetinaNet 을 통해 COCO 데이터셋[4]을 학습한 모델을 GUI 프로그램에 연동했기 때문이다. 학습한 모델을 프로그램에 연동하기 위해서는 별도의 추론 모델(Inference Model)로 전환할 수 있는 변환 모듈(Converter Module)이 필요하다. 이 변환 모듈은 파이썬에서 제공하는 Resnet 라이브러리에 적합한 형태로 만들어야 한다. 이는 RetinaNet 의 구조를 살펴보면 알 수 있다.

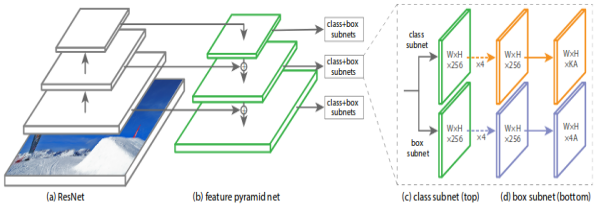

그림 4. RetinaNet 구조도

RetinaNet 은 그림 4 와 같이, 백본망(Backbone Network)으로 레즈넷(Resnet)과 두개의 Task-specific Sub Network 로 구성되어 있다. 백본망의 Resnet 에서는 입력된 전체 이미지에 대한 피처맵(Feature Map)을 계산하고, 두 개의 'Task-Sub' 네트워크에서는 각각 분류와 경계 상자(Bounding Box)를 추정하는 역할을 한다. 이러한 구조적 형태 때문에, 프로그램과 연동하기 위해서는 Resnet 의 구조적 설계를 포함하는 학습 모듈과, 이러한 모듈에 적합한 형태로 데이터가 입력할 수 있도록 하는 학습 데이터 전처리 모듈이 필요하다. 텐서플로(Tensorflow)에서는 이를 케라스 기반의 모듈로서 제공하고 있으며, 케라스 기반의 모듈 형식에 일치하는 추론(Inference) 모듈을 구현하여 프로그램과 네트워크가 연동할 수 있도록 구성했다.

4. 프로그램 활용 및 실험 결과

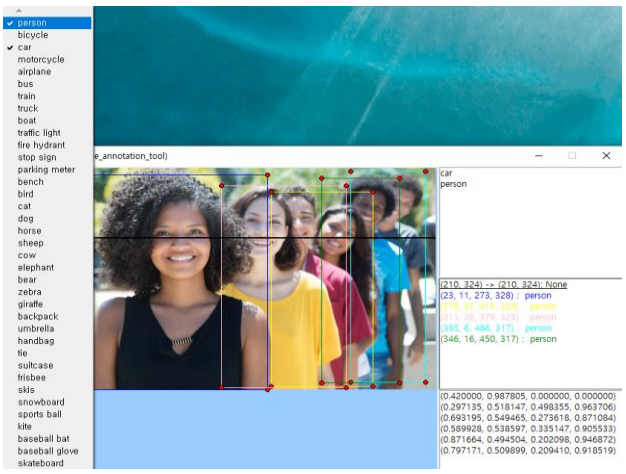

그림 5. COCO-dataset 학습한 RetinaNet 활용

그림 5 는 COCO 데이터셋으로 학습한 RetinaNet 을 케라스 기반으로 구현한 변환 모듈에 입력하고, 이를 GUI 프로그램에 연동하여 이미지 데이터를 인식한 결과이다. 프로그램의 우측에는 이미지 데이터를 인식한 좌표가 기록되는 것을 확인할 수 있다. 좌표에는 두가지 형식으로 구성되어 있고, 첫째는 'int' 형태의 절대값 좌표로 'Pascal VOC' 데이터셋[5]에 적합한 좌표이다. 두번째는 YOLO 모델의 형식으로, 객체를 잡은 박스의 중심점으로부터 너비와 높이의 비율에 따른 계산으로 'float' 형태이다.

사용자가 생성한 모델을 프로그램에 입력함으로써, 학습 데이터를 버튼 클릭만으로 Annotation 과정을 진행할 수 있기 때문에, 좀 더 빠르고 실수가 없는 학습데이터를 생성할 수 있게 됐다.

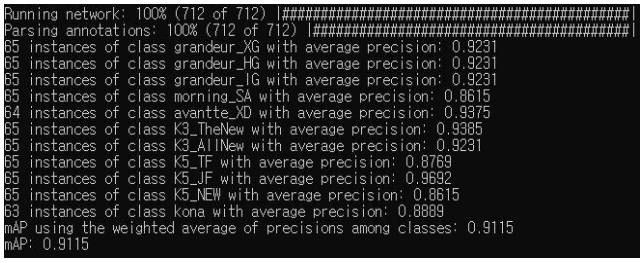

그림 6. 차량 학습 모델에 대한 인식 결과

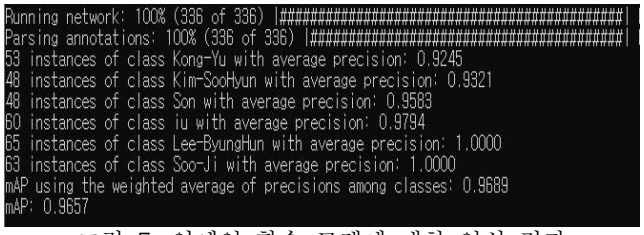

그림 7. 연예인 학습 모델에 대한 인식 결과

본 논문에서는 이러한 활용 방식을 사용하여 두 가지 방식으로 실험했다. 첫 번째 활용 방식은 차량 모델 11 종(그랜저 XG, 그랜저 HG, 그랜저 IG, 모닝, 아반떼 XD, 더 뉴 K3, 올 뉴 K3, K5-JF, K5-JF, 뉴 K5, 코나)에 대해 학습하고, 11 종에 대한 차량 식별 정확도를 알아본다. 두 번째 활용은 총 6 명의 유명 연예인(공유, 김수현, 손흥민, 아이유, 이병헌, 수지)의 이미지를 학습하고, 연예인의 이미지에 대한 정확도를 살펴본다. 각각 활용 방식에 따라 테스트 이미지를 준비하고, 학습한 신경망으로 테스트 이미지에 대한 정확성을 평가할 수 있는 테스트 모듈을 준비했다. 구성한 테스트 모듈에 이미지를 입력하면 학습된 Retinanet 의 정확도를 측정한다.

그림 8. 잘못된 형태의 Bounding Box

개별적인 학습 주제를 가지고 실험한 결과, 테스트 모듈을 통해 이미지를 분류하는 정확도는 매우 높게 나왔다. 차량 11 종에 대한 정확도를 보면 mAP 약 91%를 기록했다. 유명 연예인에 대한 정확도는 그림 7 과 같이, mAP 약 96% 로, 이미지 분류에 대한 높은 정확성을 보였다.

하지만 객체를 포착하는 경계 상자(Bounding Box)에 대해서는 그림 8 과 같이 불안정한 모습이 보였다. 개별적인 데이터에 맞는 전처리 모듈과 학습 모듈을 구현한다고 하더라도, 본 프로그램에서 직접 학습한 모델을 입력하여 annotation 과정에서 사용한다면, 객체의 경계 상자를 잡아내는 것이 위와 같이 불안정하게 물체를 인식할 수 있다. 이를 대비하여, 사용자가 불안전한 박스의 좌표를 마우스 조작을 통해 변경할 수 있도록 마우스 액션 기능을 추가하였다.

5. 결론

본 논문에서는 이미지 학습 데이터를 구성하는데 필요한 다양한 과정을 하나의 GUI 프로그램으로 구현했다. 라벨링 과정에서 사용하는 다양한 프로그램은 많지만, 사용자가 따를 수 있는 제약사항을 파악하여 만든 프로그램이다. 사용자가 데이터를 실시간으로 수집하고, 곧바로 수집한 데이터를 버튼 클릭만으로 annotation 진행할 수 있도록 하였으며, 기존의 학습된 모델을 프로그램에 입력할 수 있도록 추론 모듈을 구현했다. 이에 사용자는 개별적인 모델을 구성하는데 필요한 이미지 데이터를 하나의 프로그램에서 구성할 수 있게 되었다. 또한 사용자가 직접 수집한 데이터를 활용하여 학습한 모델을 프로그램에 재입력함으로써, 추가적인 데이터를 빠르게 수집할 수 있도록 하는 데이터 수집의 준자동화 개념에 의의가 있다.

참 고 문 헌

[1] Joseph Redmon "You Only Look Once: Unified, Real-Time Object Detection", arXiv:1506.02640v5 [cs.CV] 9 May 2016

[2] Tsung-Yi Lin, Priya Goyal "Focal Loss for Dense Object Detection", arXiv:1405.0312v3 [cs.CV] 21 Feb 2015

[3] Zhou Dengwen, "An edge-directed bicubic interpolation algorithm", 10.1109/CISP.2010.5647190 21, 29 November 2010

[4] Tsung-Yi Lin, "Microsoft COCO: Common Objects in Context", arXiv:1405.0312v3 [cs.CV] 21 Feb 2015

[5] Mark Everingham, Luc Van Gool "The Pascal Visual Object Classes (VOC) Challenge", 30 July 2008