

Object Detection과 Segmentation을 위한 다양한 패키지들

✓ Keras와 Tensorflow 기반의 다양한 오픈소스 패키지들



쉬운 활용과 Customization 가능한 장점
Object Detection 알고리즘 별로 서로 다른 패키지들을 적용해야 하는 단점.

✓ OpenCV 의 DNN 모듈



- 간편하게 Object Detection Inference 가능
- 학습이 불가능하며 주로 CPU 위주로 GPU 활용이 어려움.

✓ Tensorflow Object Detection API



- 가장 많은 Detection 알고리즘 적용 가능
- 다루기가 어렵고 특히 학습을 위한 절차가 너무 복잡

Coco dataset 등으로 Pretrained된 model을 사용. (Custom dataset 이 아니면 train (X))

파이썬 딥러닝 컴퓨터 비전

2:59 / 15:43

MacBook Pro

Keras와 Tensorflow 기반의 다양한 오픈소스 패키지들

YOLO



Keras-yolo3

- 개인 개발자가 만든 패키지. 심플하면서도 좋은 성능 보유
- Tensorflow 1.13, Keras 2.2 필요
- Git소스: <https://github.com/qqqweeee/keras-yolo3>

tf 1.13, tf 1.15로 각각 Conda 사용환경이 다름

Retinanet



Keras-Retina^{net}

- 정교한 검출 성능으로 많은 사용자에게 각광 받고 있음
- Tensorflow 1.15, Keras 2.3 필요
- Git소스: <https://github.com/fizyr/keras-retinanet>

Mask R-CNN



Mask R-CNN

- Segmentation에서 매우 많이 활용되고 있음. 다양한 기능과 편리한 사용성으로 여러 프로젝트에 적용됨.
- Tensorflow 1.13, Keras 2.2 필요
- Git소스: https://github.com/matterport/Mask_RCNN

파이썬 딥러닝 컴퓨터 비전

5:50 / 15:43

MacBook Pro

OpenCV의 Deep Learning 지원 – DNN 패키지

OpenCV는 Computer Vision 처리를 위해 널리 사용되는 범용 Library. 오픈 소스 기반의 실시간 이미지 프로세싱에 중점을 둔 라이브러리이며 C/C++ 기반의 Library이지만 Python, Java등과의 Interface도 제공

- Image Classification, Object Detection, Segmentation 등의 분야에 Machine learning 기능을 포팅하기 시작
- Deep Learning 기반의 Computer Vision 역시 별도 DNN 모듈로 구현되어 OpenCV에 포팅됨.
- DNN 모듈은 OpenCV 3.1에서는 추가 모듈 형태로 지원되었고, OpenCV 3.3 버전부터 기본 모듈에 포함.

DNN

OpenCV

파이썬 딥러닝 컴퓨터 비전

▶ 8:48 / 15:43

MacBook Pro

OpenCV dnn 모듈에서 지원하는
딥러닝 프레임워크

- 카페(Caffe) ✓
- 텐서플로(TensorFlow) ↗ 별도의 Deep Learning 프레임워크가 필요하지 않으며 타 프레임워크의 모델을 그대로 로드하여 Inference 가능.
- 토치(Torch) ✓
- 다크넷(Darknet) ↗ 그러나 모델 학습이 불가능하며, GPU 활용이 어려움. 주로 CPU 위주

Tensorflow object detection API 지원 모델

타 오픈 소스 패키지 대비 가장 많은 Object Detection 알고리즘을 지원. MS-COCO 뿐만 아니라 Open-Image 기반의 Pretrained 모델도 지원
Faster RCNN, SSD, FRCN, MASK-RCNN 지원. Backbone도 Mobilenet, Inception, Resnet 등 다양하게 지원
현재 2020년 2월 기준, YOLO 지원하지 않음. 그리고 Retinanet에 대한 지원도 약함.

pretrained MS-COCO로 학습된 모델

detection model	Model name	Speed (ms)	COCO mAP[^1]	Outputs
ssd_mobilenet_v1_coco	faster_rcnn_resnet50_coco	30	21	Boxes
ssd_mobilenet_v1_0.75_depth_coco *	faster_rcnn_resnet50_lowproposals_coco	26	18	Boxes
ssd_mobilenet_v1_quantized_coco *	rfcn_resnet101_coco	29	18	Boxes
ssd_mobilenet_v1_0.75_depth_quantized_coco *	faster_rcnn_resnet101_lowproposals_coco	29	16	Boxes
ssd_mobilenet_v1_ppn_coco *	faster_rcnn_inception_resnet_v2 atrous_coco	26	20	Boxes
ssd_mobilenet_v1_fpn_coco *	faster_rcnn_inception_resnet_v2 atrous_lowproposals_coco	56	32	Boxes
ssd_resnet_50_fpn_coco *	faster_rcnn_nas	76	35	Boxes
ssd_mobilenet_v2_coco	faster_rcnn_nas_lowproposals_coco	31	22	Boxes
ssd_mobilenet_v2_quantized_coco	mask_rcnn_inception_resnet_v2 atrous_coco	29	22	Boxes
ssdlite_mobilenet_v2_coco	mask_rcnn_inception_v2_coco	27	22	Boxes
ssd_inception_v2_coco	mask_rcnn_resnet101 atrous_coco	42	24	Boxes
faster_rcnn_inception_v2_coco	mask_rcnn_resnet50 atrous_coco	58	28	Boxes

파이썬 딥러닝 컴퓨터 비전

▶ 12:53 / 15:43

MacBook Pro

Tensorflow의 Mobile Device에서의 Object Detection 집중

Lite

→ 서버사양 ↓, 주로 CPU 기반

→ 주로 엣지 컴퓨팅, 등 가벼운 환경에서 돌아갈 수 있게!

SSD + MobileNet

GPU, CPU에서도 상대적으로 빠름
(tiny YOLO 만큼은 아니지만)

VS

(YOLO) Tiny-YOLO

CPU 환경에서 ↓, 정확도 ↓
But, GPU 환경에서 속도 ↑

파이썬 딥러닝 컴퓨터 비전

▶ 13:03 / 15:43

MacBook Pro

Tensorflow Object Detection의 문제점

Tensorflow Object Detection API



TensorFlow

- 불필요하게 복잡한 학습 과정
- Tensorflow Version 2와 Version 1의 충돌.
- 많은 API가 사용되지만 Document나 Tutorial의 부족
- Research 형태의 모듈로 안정화에 의문

파이썬 딥러닝 컴퓨터 비전

▶ 15:34 / 15:43

MacBook Pro