

Level 2. How to customize AI engines



<http://mindslab.ai>

1. 도로상의 객체 인식이란?

1. 학습 방법

- 창문 검출 (Window Detection)
- 번호판 검출 (Plate Detection)



도로상의 객체 인식이란?

What is AI Vehicle Recognition (AVR)?

도로상의 객체 인식이란

도로상의 객체 인식은 도로 위에 있는 CCTV 카메라를 통해 들어온 이미지에 대해 차량의 창문과 번호판의 위치, 그리고 보행자나 운전자의 얼굴을 검출하는 기능을 수행한다.

- 창문 검출 (Window Detection) : 도로 위 차량의 창문의 위치를 검출하는 기능
- 번호판 검출 (Plate Detection) : 도로 위 차량의 번호판의 위치를 검출하는 기능
- 얼굴 검출 (Face Detection) : 사람의 얼굴의 위치를 검출하는 기능

1. 도로상의 객체 인식이란?

1. 학습 방법

- 창문 검출 (Window Detection)
- 번호판 검출 (Plate Detection)



학습 데이터

- 2개의 촬영 환경 (도로 위 CCTV 환경, 톨게이트 카메라 환경)
- 대략 1,000건의 도로 위 차량이 찍힌 이미지
- 이미지 파일(jpg), 이미지 파일에 매칭되는 레이블링 파일(txt)
- 레이블링 파일에는 창문 위치 정보가 포함된다.



학습 모델

YOLO v3

- YOLO(You Only Look Once)는 Object Detection 분야에서 가장 유명한 딥러닝 모델 중 하나
- single convolutional network를 통해 multiple bounding box에 대한 class probability를 계산하는 방식



+



학습 command

- train: 학습 모드
- cfg/coco.data: 데이터 설정 파일 경로
- cfg/yolov3.cfg: 네트워크 설정 파일 경로
- darknet53.conv.74: 사전 학습 모델 경로
- gpus: 학습에 사용될 GPU 번호

```
./darknet detector \
train \
cfg/coco.data \
cfg/yolov3.cfg \
darknet53.conv.74 \
-gpus 0,1,2,3
```

Inference command

- test: Inference 모드
- cfg/coco.data: 데이터 설정 파일 경로
- cfg/yolov3.cfg: 네트워크 설정 파일 경로
- yolov3.weights: 학습한 모델 경로
- data/test1.jpg: 입력 이미지 경로

```
./darknet detector \  
test \  
cfg/coco.data \  
cfg/yolov3.cfg \  
yolov3.weights \  
data/test1.jpg
```


1. 도로상의 객체 인식이란?

1. 학습 방법

- 창문 검출 (Window Detection)
- 번호판 검출 (Plate Detection)



학습 방법

번호판 검출 (Plate Detection)

학습 데이터

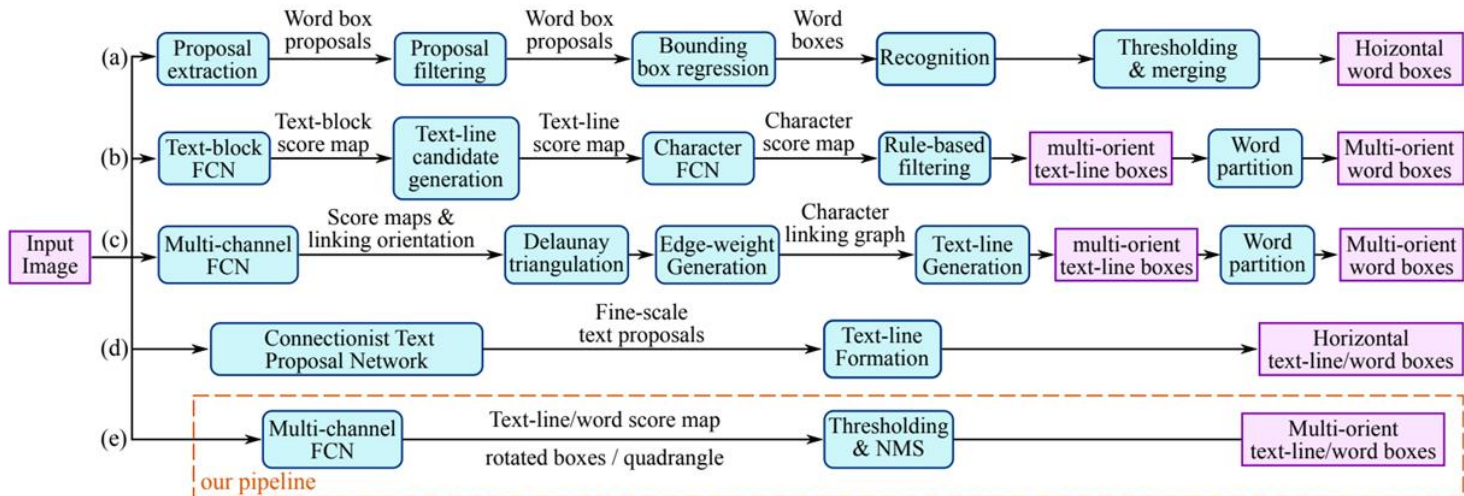
- 2개의 촬영 환경 (도로 위 CCTV 환경, 톨게이트 카메라 환경)
- 대략 5,000건의 도로 위 차량이 찍힌 이미지
- 이미지 파일(jpg), 이미지 파일에 매칭되는 레이블링 파일(txt)
- 레이블링 파일에는 번호판 위치 정보가 포함된다.



학습 모델

EAST (ResNet)

- EAST(An Efficient and Accurate Scene Text Detector)는 Scene Text Detection 분야의 딥러닝 모델이며 ResNet 네트워크로 설계되었음.



Source tree

- checkpoint: EAST 학습 모델
- data_util: data generator 관련 유틸
- eval: inference 기능
- multigpu_train: training 기능
- nets: resnet 모델 네트워크 관련
- training_samples: 학습 데이터셋 샘플

```
— checkpoint_plate_0901
  |— checkpoint
  |— model.ckpt-42421.data-00000-of-00001
  |— model.ckpt-42421.index
  |— model.ckpt-42421.meta
— data_util.py
— eval.py
— __init__.py
— LICENSE
— locality_aware_nms.py
— model.py
— multigpu_train.py
— nets
  |— __init__.py
  |— resnet_utils.py
  |— resnet_v1.py
— readme.md
— requirements.txt
— run_demo_server.py
— training_samples
  |— img_1.jpg
  |— img_1.txt
  |— img_2.jpg
  |— img_2.txt
```

학습 command

- gpu_list: 학습에 사용할 GPU 번호
- input_size: 입력 이미지 크기
- batch_size_per_gpu: 한 GPU 당 batch size
- training_data_path: 학습 데이터셋 경로
- geometry: 검출 박스 종류 지정
- learning_rate: 학습률 (폭)
- pretrained_model_path: 이어서 학습할 사전 모델 경로

```
python multigpu_train.py
--gpu_list=0
--input_size=512
--batch_size_per_gpu=14 --
checkpoint_path=/tmp/east_icdar2015_resnet_v1_50_rbox/
--text_scale=512 --
training_data_path=/data/ocr/icdar2015/
--geometry=RBOX
--learning_rate=0.0001
--num_readers=24
--
pretrained_model_path=/tmp/resnet_v1_50.
ckpt
```

Inference command

- test_data_path: 테스트 할 데이터셋 경로
- gpu_list: 테스트에 사용될 GPU 번호
- checkpoint_path: 학습한 모델의 경로
- output_dir: 인식 결과 이미지가 저장될 폴더 경로

```
python eval.py --  
test_data_path=/tmp/images/ --  
gpu_list=0 --  
checkpoint_path=/tmp/east_icdar201  
5_resnet_v1_50_rbox/  
  
--output_dir=/tmp/
```

관련 자료



- <https://pjreddie.com/darknet/yolo/>
- <https://pysource.com/2019/06/27/yolo-object-detection-using-opencv-with-python/>
- <https://github.com/argman/EAST>
- <https://www.pyimagesearch.com/2018/08/20/opencv-text-detection-east-text-detector/>



감사합니다.

MINDs Lab. AI Platform Company