CRAFT 구현 코드 정의서

* 최연식 -

Version : 1.0

1. 개요

* 본 코드는 CLOVA AI에서 2019년에 제안한 논문으로(Character Region Awareness for Text Detection(CRAFT)) Scene-Text Detection(STD) 태스크를 위한 코드임.
* STD란, Wild 환경의 이미지 속 글자 위치를 Localization하는 것임.
* CLOVA AI에서 제공하는 Official Code의 경우 내부적인 이유로 학습 부분에 대한 코드를 제공하지 않고 있음. 이에 이를 구현한 CRAFT-Remade라는 코드를 이용하여 학습을 진행하였음.
* Official Code의 경우 ICDAR2013 데이터 셋에 대해 H-mean(F1-score) 95.2%가 나왔으며, CRAFT-Remade 코드의 경우 재현 결과 82%가 나왔음.

1. 코드구성도(주요 파일 별도 설명)

CRAFT-Remade-ys

├── config.py : Hyper Parameter 및 데이터 경로 설정 파일

├── LICENSE

├── main.py : 실질적인 실행 파일

├── ReadMe.md

├── requirements.txt : pip 내 설치된 패키지 버전 안내 파일

├── src

│ ├── craft\_model.py

│ ├── generic\_model.py

│ ├── \_\_pycache\_\_

│ │ ├── craft\_model.cpython-37.pyc

│ │ ├── generic\_model.cpython-37.pyc

│ │ └── vgg16bn.cpython-37.pyc

│ ├── UNET\_ResNet.py

│ ├── utils

│ │ ├── check\_dataset.py

│ │ ├── data\_manipulation.py

│ │ ├── data\_structure\_ic13.py

│ │ ├── data\_structure\_ic15.py

│ │ ├── data\_structure\_ic17.py

│ │ ├── merger.py

│ │ ├── parallel.py

│ │ ├── running\_mean.py

│ │ ├── utils\_old.py

│ │ └── utils.py

│ └── vgg16bn.py

├── train\_synth

│ ├── config.py

│ ├── dataloader.py

│ ├── synthesize.py

│ ├── test.py

│ └── train.py

└── train\_weak\_supervision

├── config.py : Weak Supervision에 대한 추가적인 설정 파일

├── dataloader.py

├── \_\_init\_\_.py

└── trainer.py : Weak Supervision 학습이 진행되는 코드

1. 실행방법

* 학습방법

python main.py weak-supervision --model {{weight\_file\_path}} --iter {{number\_of\_12.5k\_iteration}}

* 추론방법

python main.py synthesize --model {{weight\_file\_path}} --folder {{test\_folder}}

1. 데이터 셋

* SynthText (English) : 이미지에 글자를 인위적으로 합성한 데이터 셋으로 약 80만장의 이미지가 있음. Character-level annotation.

<SynthText Dataset 예시>

* 텍스트, 컨테이너이(가) 표시된 사진

  자동 생성된 설명ICDAR 2013 (English) : 본 태스크에 주로 쓰이는 벤치마크 데이터 셋 중 하나로, 229장의 학습 이미지와 233장의 테스트 이미지로 구성되어있음. Word-level annotation.

<ICDAR 2013 Dataset 예시>

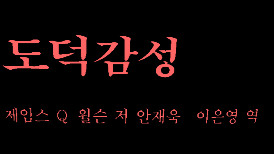
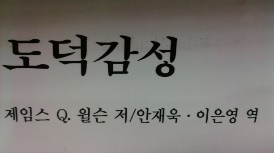
* ICDAR 2017(MLT) (Eng, Kor, … 9 multi-lingual) : 본 태스크에 주로 쓰이는 벤치마크 데이터 셋 중 하나로, 7200장의 학습 이미지와 1800장의 검증 이미지, 9000장의 테스트 셋으로 구성되어있음. Word-level annotation.

<ICDAR 2017(MLT) Dataset 예시>

* 무한정보기술 데이터 셋(Kor+Eng) : 무한정보기술 회사에서 수집 및 라벨링을 한 데이터 셋으로 2914장의 이미지가 있음. 라벨링 품질이 좋지 않음.(간판 전체 영역으로 크게 라벨링이 되어 있음.) Word-level annotation.

<무한정보기술 Dataset 예시>

* KAIST 데이터 셋(Kor+Eng) : KAIST에서 2011년에 수집 및 배포한 데이터 셋으로 간판 글자 위주의 데이터 셋임. 해상도는 640\*480으로 낮지만 객체 스케일이 커서 모자이크 기법으로 학습하면 좋은 성능을 낼 수 있을 것으로 보임. Word-level annotation



<KAIST Dataset 예시>

* 학습 가중치 파일

1. craft\_mlt\_25k.pth : Official code에서 제공하는 파일로, ICDAR13 기준 F1 95.2% 달성
2. remade\_ic13\_17.pkl : 본 코드로 구현한 학습 모델 파일로, F1 82% 달성
3. SynthText.pkl : 합성데이터를 이용하여 50k iter만큼 학습한 모델 파일

* 위 데이터 셋과 가중치 파일들은 1번 서버 /data2/yeonsik/std/ 폴더에서 확인할 수 있음.

1. 결론

* 현재 SynthText + ICDAR 2013 + ICDAR2017 데이터 셋으로 학습을 재현해보았으나 Official Code에서 학습 부분에 대한 정보 제공이 많이 부족하여(전반적인 학습전략, 하이퍼 파라미터 등) 재현에 큰 어려움이 있음. (ICDAR 2013 데이터 셋에 대하여 H-mean 기준, Official : 95.2%, 재현 : 82%)

1. 참고자료

* Official code : <https://github.com/clovaai/CRAFT-pytorch>
* 본 학습재현에 사용된 code : <https://github.com/yeonsikch/CRAFT-Remade-ys>
* 추가적으로 사용 해볼만한 code : <https://github.com/backtime92/CRAFT-Reimplementation>