

MINI PROJECT No.1

Durable Rule패키지를 사용한 규칙 생성

Contens

- 규칙 설명

- 코드 분석

- 구현 결과

규칙 설명

subject	explain	do
학습 전	DB는	전처리 과정을 거친다.
학습 전	Dataset은	Train, Validation, Testset으로 나눈다.
GT 파일이	json 또는 xml 형태로 제공되면	txt 형태로 변경한다.
txt 형태의 GT 파일은	TopX, BottomY, Width, Height	형태를 가진다.
학습 시	IMG 파일의 크기는	640x640 사이로 조정한다.
학습 시	IMG 파일의 확장자는	PNG로 변경한다.
학습 시	GPU는	최대 4개를 이용한다.
프레임 수 증가를 위해	학습 매개변수를	조정한다.
프레임 수 증가를 위해	IMG 파일의 크기는	416x416 사이즈로 조정한다.
프레임 수 증가를 위해	비교적 가벼운	tiny 모델을 이용한다.
성능 향상을 위해	Data Augmentation을	적용한다.
Data Augmentation은	imgaug	python 라이브러리를 이용한다.
Data Augmentation은	imgaug를 이용하여	Blur, Rotatoin, Shear 등을 적용한다.
학습 후	Testset을 이용하여	성능을 평가한다.
모델 성능은	mAP를 기준으로	평가한다.

코드 분석

durable 패키지 import 및 규칙 생성

```
from durable.lang import *

with ruleset('Company_Rule'):
    # 규칙 생성
    @when_all((m.explain == "DB는") & (m.do == "전처리 과정을 거친다."))
    def ruel1(c):
        c.assert_fact({'subject': c.m.subject, 'explain': 'DB는', 'do': '전처리 과정을 거친다.'})

    @when_all((m.explain == "Dataset은") & (m.do == "Train, Validation, Testset으로 나눈다."))
    def ruel2(c):
        c.assert_fact({'subject': c.m.subject, 'explain': 'Dataset은', 'do': 'Train, Validation, Testset으로 나눈다.'})

    @when_all((m.explain == "json 또는 xml 형태로 제공되면") & (m.do == "txt 형태로 변경한다."))
    def ruel3(c):
        c.assert_fact({'subject': c.m.subject, 'explain': 'json 또는 xml 형태로 제공되면', 'do': 'txt 형태로 변경한다.'})

    @when_all((m.explain == "IMG 파일의 크기는") & (m.do == "640x640 사이로 조정한다."))
    def ruel4(c):
        c.assert_fact({'subject': c.m.subject, 'explain': 'IMG 파일의 크기는', 'do': '640x640 사이로 조정한다.'})
```

코드 분석

기존 규칙으로 새로운 규칙 추가 생성 및 출력

```
# 기존 규칙으로 새로운 규칙 추가

@when_all((m.explain == 'json 또는 xml 형태로 제공되면') & (m.do == "txt 형태로 변경한다."))
def Make_Rule1(c):
    c.assert_fact({'subject': 'txt 형태의 GT 파일은', 'explain': 'TopX, BottomY, Width, Height', 'do': '형태를 가진다. ##### 규칙 추가 #####'})

@when_all((m.explain == 'Data Augmentation을') & (m.do == "적용한다."))
def Make_Rule2(c):
    c.assert_fact({'subject': 'Data Augmentation은', 'explain': 'imgaug', 'do': 'python 라이브러리를 이용한다. ##### 규칙 추가 #####'})

@when_all((m.explain == 'imgaug') & (m.do == "python 라이브러리를 이용한다. ##### 규칙 추가 #####"))
def Make_Rule3(c):
    c.assert_fact({'subject': c.m.subject, 'explain': 'imgaug를 이용하여', 'do': 'Blur, Rotatoin, Shear 등을 적용한다. ##### 규칙 추가 #####'})

@when_all((m.explain == 'Testset을 이용하여') & (m.do == "성능을 평가한다."))
def Make_Rule4(c):
    c.assert_fact({'subject': '모델 성능은', 'explain': 'mAP를 기준으로', 'do': '평가한다. ##### 규칙 추가 #####'})

# 규칙 출력
@when_all(+m.subject)
def output(c):
    print('Fact: {0} {1} {2}'.format(c.m.subject, c.m.explain, c.m.do))
```

코드 분석

규칙 입력

```
assert_fact('Company_Rule', {'subject': '학습 전', 'explain': 'DB는', 'do': '전처리 과정을 거친다.'})
assert_fact('Company_Rule', {'subject': '학습 전', 'explain': 'Dataset은', 'do': 'Train, Validation, Testset으로 나눈다.'})
assert_fact('Company_Rule', {'subject': 'GT 파일이', 'explain': 'json 또는 xml 형태로 제공되면', 'do': 'txt 형태로 변경한다.'})
assert_fact('Company_Rule', {'subject': '학습 시', 'explain': 'IMG 파일의 크기는', 'do': '640x640 사이로 조정한다.'})
assert_fact('Company_Rule', {'subject': '학습 시', 'explain': 'IMG 파일의 확장자는', 'do': 'PNG로 변경한다.'})
assert_fact('Company_Rule', {'subject': '학습 시', 'explain': 'GPU는', 'do': '최대 4개를 이용한다.'})
assert_fact('Company_Rule', {'subject': '프레임 수 증가를 위해', 'explain': '학습 매개변수를', 'do': '조정한다.'})
assert_fact('Company_Rule', {'subject': '프레임 수 증가를 위해', 'explain': 'IMG 파일의 크기는', 'do': '416x416 사이즈로 조정한다.'})
assert_fact('Company_Rule', {'subject': '프레임 수 증가를 위해', 'explain': '비교적 가벼운', 'do': 'tiny 모델을 이용한다.'})
assert_fact('Company_Rule', {'subject': '성능 향상을 위해', 'explain': 'Data Augmentation을', 'do': '적용한다.'})
assert_fact('Company_Rule', {'subject': '학습 후', 'explain': 'Testset을 이용하여', 'do': '성능을 평가한다.'})
```

구현 결과

Fact: 학습 전 DB는 전처리 과정을 거친다.

Fact: 학습 전 Dataset은 Train, Validation, Testset으로 나눈다.

Fact: txt 형태의 GT 파일은 TopX, BottomY, Width, Height 형태를 가진다. ##### 규칙 추가 #####

Fact: GT 파일이 json 또는 xml 형태로 제공되면 txt 형태로 변경한다.

Fact: 학습 시 IMG 파일의 크기는 640x640 사이로 조정한다.

Fact: 학습 시 IMG 파일의 확장자는 PNG로 변경한다.

Fact: 학습 시 GPU는 최대 4개를 이용한다.

Fact: 프레임 수 증가를 위해 학습 매개변수를 조정한다.

Fact: 프레임 수 증가를 위해 IMG 파일의 크기는 416x416 사이즈로 조정한다.

Fact: 프레임 수 증가를 위해 비교적 가벼운 tiny 모델을 이용한다.

Fact: Data Augmentation은 imgaug를 이용하여 Blur, Rotatoin, Shear 등을 적용한다. ##### 규칙 추가 #####

Fact: Data Augmentation은 imgaug python 라이브러리를 이용한다. ##### 규칙 추가 #####

Fact: 성능 향상을 위해 Data Augmentation을 적용한다.

Fact: 모델 성능은 mAP를 기준으로 평가한다. ##### 규칙 추가 #####

Fact: 학습 후 Testset을 이용하여 성능을 평가한다.

깃허브 주소 <https://github.com/ej-1206/Durable-Rule>