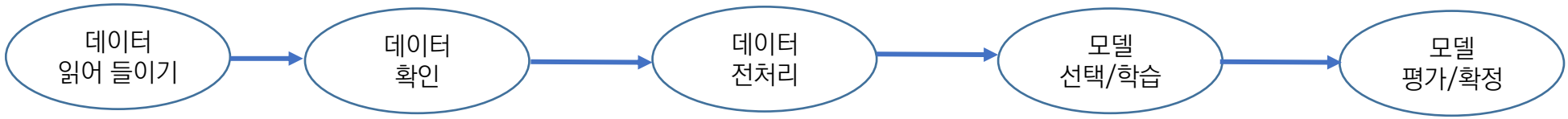
Automated Machine Learning

프로세스 설계서

작성자 : 송창화

1. 프로세스 개요



데이터 읽어 들이기 (Data Read)

1. [신규] 버튼
   (1) 파일 선택, 문서명-> [등록]
   (2) Excel Data -> DB 처리
   (3) 컬럼유형, 최소값,최대값, 평균, 표준편차,분산 제1/2/3 사분위(Q1~3) 누락값수 계산 처리
   (4) Boxplot Image 생성

2. [선택] 버튼
   (1) 데이터 업로드 결과 display

3. [삭제] 버튼
   (1) 해당 문서 삭제 처리

데이터 확인 (Data Confirm)

1. 문서목록 - [선택] 버튼
   (1) 컬럼별 데이터 분석 결과
   - 컬럼유형, 최소값,최대값, 평균, 표준편차,분산 제1/2/3 사분위(Q1~3) 누락값수 Display

2. 컬럼별 데이터 분석 결과 - [선택] 버튼
   (1) 데이터 분포(Boxplot) Image display
   (2) [이미지]를 클릭하면 이미지가 [확대]됨.

데이터 전처리 (Data Preprocessing)

1. [목표변수 적용] 버튼
   (1) 목표변수 선택 Check 처리

2) [데이터 전처리 적용] 버튼
   (1) 중복값 처리
   (2) 결측값 처리
   (3) 이상값 처리

모델 선택/학습 (Model Selection/Training)

1. [분류 모델] 버튼
   (1) 그래디언트 부스팅(Gradient Boost)
   (2) 의사 결정 트리(Decision Tree)
   (3) Random Forest
   (4) SVM(Support Vector Machine)
   (5) 신경망(Neural Network)

2. [군집 모델] 버튼
   (1) K-평균 군집화 (K-means Clustering)

3. 모델별 학습 처리
   (1) Confusion Matrix(혼동/오차 행렬)
   (2) (신경망) 훈련 데이터 대 검증 데이터 손실 그래프
   (3) ROC Curve (수신자 판단 곡선)

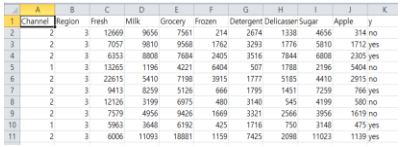
모델 평가/확정 (Model Evaluation/Confirm)

1. [확정] 버튼
   (1) 해당 문서 확정 처리 - 확정일자

2. 모델별 학습 처리 결과
   (1) Confusion Matrix
   (2) (신경망) 훈련 데이터 대 검증 데이터 손실 그래프
   (3) ROC Curve (수신자 판단 곡선)

2. 기능 프로세스

2.1 데이터 읽어 들이기 (Data Read)



1. Excel 파일(CSV 포맷)을 업로드 한다.

2 automl_dataread.py

2. 업로드된 파일을 적용하여 Auto ML 관련 테이블을 생성한다.

(1) 데이터 상세 테이블(TB_AUTOML_TEXT) Insert 처리
- Header 데이터
- 실제 데이터

(2) 데이터 컬럼 상세 테이블(TB_AUTOML_COLUMN_INFO) Insert 처리
- 컬럼유형, 최소값,최대값,평균, 표준편차,분산, 제1/2/3 사분위(Q1~3)
- 누락값수

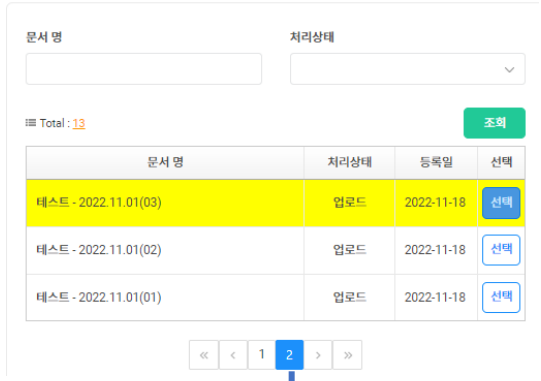
(3) 데이터 마스터 테이블(TB_AUTOML_MASTER) Update 처리
- 처리상태 코드 : 업로드
- 업로드 파일 행 개수, 업로드 파일 행 개수

3 automl_dataconfirm.py

3. 데이터 상세 테이블(TB_AUTOML_TEXT)을 select하여 데이터 컬럼 이미지 테이블을 생성한다.

(1) Boxplot((상자 수염 그림) Image 생성
- Boxplot(상자 수염 그림) – 서버에 저장하기
- IMAGE_GUBUN(이미지 구분) : '02'

2.2 데이터 확인 (Data Confirm)



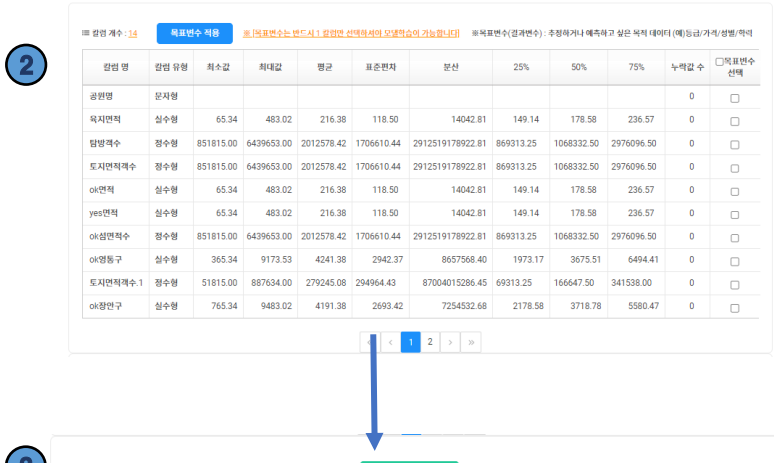
1. 문서목록에서 [선택] 버튼을 클릭한다.

2. ‘컬럼별 데이터 분석 결과’ 목록에서 [선택] 버튼을 클릭한다.
   (1) 3.번 처리
   (2) [데이터 확인] 버튼을 클릭하면 처리상태가 ‘데이터 확인’으로 변경된다.

3. 데이터 분포(Boxplot) Image display
   – [이미지]를 클릭하면 이미지가 [확대]된다.

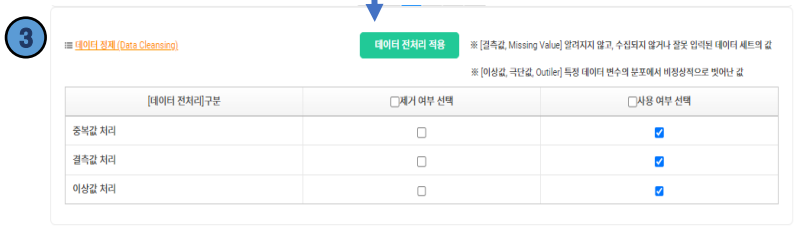
2.3 데이터 전처리 (Data Preprocessing)

1. 문서목록에서 [선택] 버튼을 클릭한다.



2. ‘컬럼별 데이터 분석 결과’ 목록에서 [목표변수 선택] Check 처리한다.

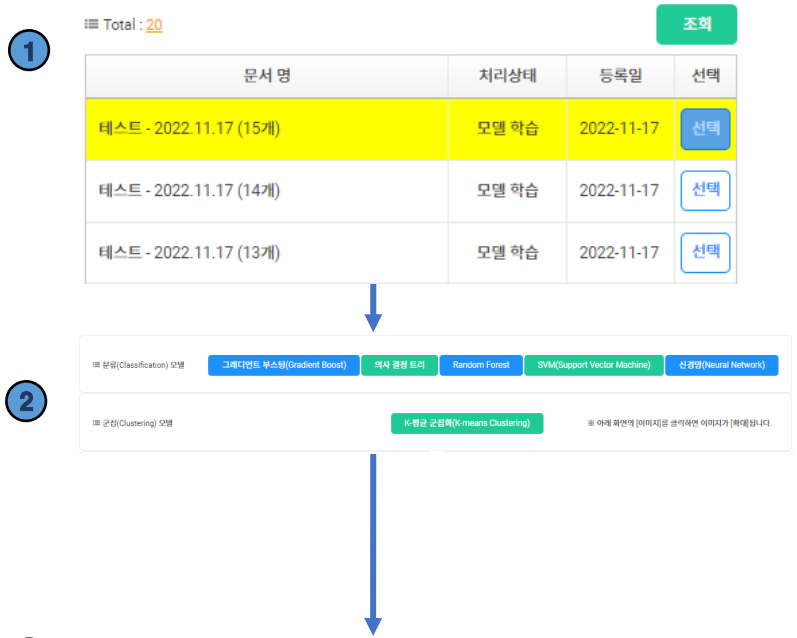
   (1) [목표변수 적용] 버튼을 클릭한다.
   - 처리상태가 ‘데이터 전처리’로 변경된다.



3. [데이터 전처리] 구분별로 제거 여부, 사용 여부를 선택한다.

   (1) 중복값 처리
   (2) 결측값 처리
   (3) 이상값 처리

2.4 모델 선택/학습 (Model Selection/Training)



1. 문서목록에서 [선택] 버튼을 클릭한다.

2. 학습할 [분류 모델] / [군집 모델]을 클릭한다.

(1) 분류 모델
- 그래디언트 부스팅(Gradient Boost)
- 의사 결정 트리(Decision Tree)
- Random Forest
- SVM(Support Vector Machine)
- 신경망(Neural Network)

(2) 군집 모델 : K-평균 군집화(K-means Clustering)

3. [분류 모델] / [군집 모델]을 학습하시겠습니까? -> [예]를 클릭한 경우

(1) 데이터 컬럼 상세 테이블(TB_AUTOML_COLUMN_INFO) select
(2) 데이터 전처리(Data Preprocessing) – 목표변수 활용
- 변수 정의
- 범주형 데이터를 숫자형 데이터로 전환
- 범주형 데이터와 숫자형 데이터 결합
- 모든 특징의 이름 리스트 처리

(3) 데이터 분할 - train_test_split
(4) 알고리즘 선택
(5) 학습, 예측
(6) 학습 결과 테이블(TB_AUTOML_TRAINING_RESULT) 처리
(7) ROC Curve(수신자 판단 곡선) 처리
(8) Confusion Matrix(혼동/오차 행렬) 처리
(9) (신경망) 훈련 데이터 대 검증 데이터 손실 그래프 처리
(10) 처리상태 '모델 학습'으로 변경 처리

4

≔ Total : 5 모델 학습 목록

※정밀도(Precision) : 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율

※ 민감도(Sensitivity) : 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율

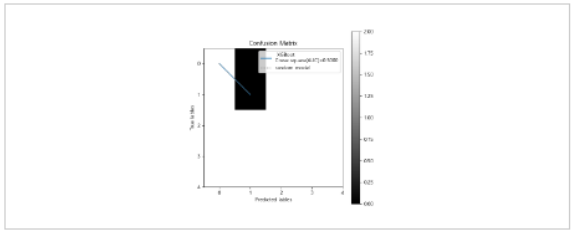
※ F1-score : 정밀도와 민감도의 조화평균 -> 2*(정밀도*민감도)/(정밀도+민감도)

학습 구분	정밀도	민감도	F1-score	학습일자	학습횟수	선택
Gradient Boost	0.5	0.5	0.5	2022-11-17	1	선택
의사결정트리	0.4	0.4	0.4	2022-11-17	1	선택
Random Forest	0.4	0.4	0.4	2022-11-17	1	선택
SVM	0.4	0.4	0.4	2022-11-17	1	선택
신경망	0.25	0.25	0.25	2022-11-17	1	선택

5

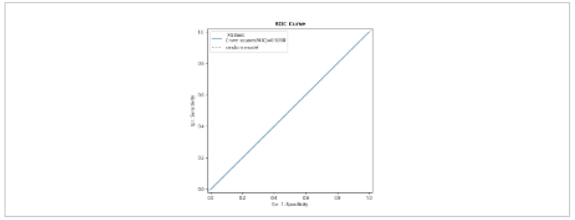
≔ Confusion Matrix (혼동행렬) / (신경망) 훈련 데이터 대 검증 데이터 손실 그래프

※(혼동/오차행렬) 학습을 통한 예측성능을 측정하기 위해 예측 값과 실제 값을 비교하기 위한 표



≔ ROC Curve ※ 특이도(Specificity) : 실제 진단결과가 음성 중에 음성을 음성이라고 맞춘 비율

※ (수신자 판단 곡선) 모델의 효율성을 민감도와 특이도를 이용하여 그래프로 나타낸 것



5. [모델 학습] 처리 후, '모델 학습 목록'에 학습한 데이터가 나타난다.

(1) 학습구분, 정밀도, 민감도, F1-score, 학습일자, 학습횟수
(2) [선택] 버튼을 클릭하면 6. 학습 결과 Image가 나타난다.

6. 학습 결과 만들어 진 Image

(1) Confusion Matrix (혼동 / 오차 행렬)
- 학습을 통한 예측성능을 측정하기 위해 예측 값과 실제 값을 비교하기 위한 표

(2) ROC Curve (수신자 판단 곡선)
- 모델의 효율성을 민감도와 특이도를 이용하여 그래프로 나타낸 것

2.5 모델 평가/확정 (Model Evaluation/Confirm)

1

Total : 22 조회

문서 명	처리상태	등록일	선택
테스트 - 2022.11.17 (15개)	모델 학습	2022-11-17	선택
테스트 - 2022.11.17 (14개)	모델 학습	2022-11-17	선택
테스트 - 2022.11.17 (13개)	모델 학습	2022-11-17	선택

1. 문서목록에서 [선택] 버튼을 클릭한다.

2

Total : 5 모델 학습 목록

※정밀도(Precision) : 모델이 True라고 분류한 것 중에서 실제 True인 것의 비율
※ 민감도(Sensitivity) : 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율
※ F1-score : 정밀도와 민감도의 조화평균 -> 2*(정밀도*민감도)/(정밀도+민감도)

학습 구분	정밀도	민감도	F1-score	학습일자	학습횟수	확정일자	선택	확정
Gradient Boost	0.5	0.5	0.5	2022-11-17	1		선택	확정
의사결정트리	0.4	0.4	0.4	2022-11-17	1		선택	확정
Random Forest	0.4	0.4	0.4	2022-11-17	1		선택	확정
SVM	0.4	0.4	0.4	2022-11-17	1		선택	확정
신경망	0.25	0.25	0.25	2022-11-17	1		선택	확정

2. [확정] 버튼을 클릭한다..

(1) 확정일자
(2) 데이터 마스터 테이블(TB_AUTOML_MASTER) Update 처리
- 처리상태 코드 : 모델 확정

EasyOCR을 적용한 OCR Solution 개발

프로세스 흐름도

작성자 : 송창화

1. 프로세스 개요

학습

1. 대상 : 사용자가 학습대상으로 선택한 글자이미지
2. 주요학습정보 : 글자이미지 파일과 해당 글자 정보
3. 처리개요 : EasyOCR로 통해 이미지 안에 있는 글자를 인식하고 인식된 글자를 다시 보정한 후 해당 이미지 글자를 학습시키므로써 글자 인식률을 높여가도록 한다.

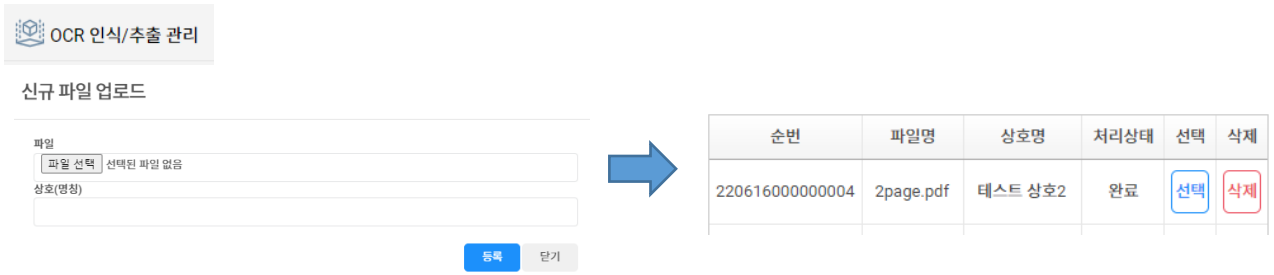
■ *실행방법*
1) Windows 실행 〉 CMD 수행
2) 학습
2-1) CD C:\dev_project\project_ocr\training
2-2) project_*ocr_training.bat (배치실행 파일 : 매일 오전 1시에 자동 실행)*

2. 기능 프로세스

2.1 문서 인식 및 추출, 학습

1

1. 인식할 이미지 또는 PDF 파일을 업로드 한다.



2

project_ocr.py

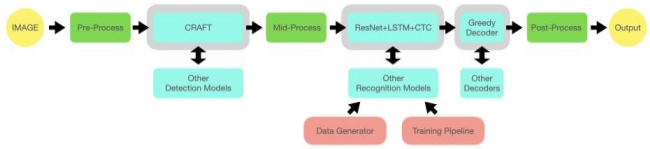
2. 업로드된 파일을 EasyOCR 프로그램을 통해 문자 인식및 추출 한다.
   이때 EasyOCR는 이미 학습된 학습모델을 사용한다.

```
ocr_reader = easyocr.Reader(['en', 'ko'], gpu=True,
                            model_storage_directory=model_storage_dir,
                            user_network_directory=user_network_dir,
                            recog_network=recog_network)
```

model_storage_directory = C:\dev_project\project_ocr\training\model
user_networ_directory = C:\dev_project\projecte_ocr\training\user_network
recog_network = custom (지정한 모델파일이름)

6

EasyOCR Framework (https://github.com/JaidedAI/EasyOCR)



3

project_ocr.py

3. 인식된 글자이미지와 해당 글자를 찾아 박싱처리하며 각각의 글자이미지와 해당 글자를 DB에 저장한다.

TB_OCR_DOC_TEXT

TEXT_IMAGE_PATH	TEXT_NM	SCORE
C:₩dev_project₩aiblue_ocr₩textimagefile₩2205290000000010_01_1.png	신고번호	0.99996
C:₩dev_project₩aiblue_ocr₩textimagefile₩2205290000000010_01_2.png	제 2021-00058호	0.79328
C:₩dev_project₩aiblue_ocr₩textimagefile₩2205290000000010_01_3.png	사 업 장 폐 기 물 배 출 자 신 고증 명 서	0.51762
C:₩dev_project₩aiblue_ocr₩textimagefile₩2205290000000010_01_4.png	'폐기물관리법 시행규칙] 제18조제2항제1호 및	0.35470
C:₩dev_project₩aiblue_ocr₩textimagefile₩2205290000000010_01_5.png	제2호에 해당하는	0.67710
C:₩dev_project₩aiblue_ocr₩textimagefile₩2205290000000010_01_6.png	경우)	0.99315

4

인식/추출 검증

4. 인식및 추출된 이미지와 글자를 확인한 후 잘못 인식된 글자에 대한 글자를 보정하여 저장한다.

5

학습대상 선정

5. 인식및 추출된 글자중 추가 학습이 필요한 대상을 선정한다.

OCR AI 학습

선택	인식 이미지	추출 글자	Score	검증 글자
☑	신고번호	신고번호	0.99997	
☐	제 202100617호	제 202100617호	0.84961	
☑	건설폐기물	건설폐기물	0.62347	

TB_OCR_DOC_TEXT

TEXT_NM	SCORE	TEXT_UPDT_NM	TEXT_TRAINING_YN
신고를	0.58454	NULL	Y
하여음을 증명합니다	0.39516	하였음을 증명합니다.	Y
2027년 08월 05일	0.93767	2021년 08년 05월	Y
서 울 특 별 시 강 남 구	0.47139	NULL	N

6

project_ocr_training.py

6. 사전학습모델을 기반으로 추가학습 대상을 일배치로 학습한다.

6.1. 학습대상 데이터의 글자이미지와 글자 정보를 가져온다.(신규 학습대상및 기존 학습건 모두 가져온다)

SELECT * FROM TB_OCR_DOC_TEXT WHERE (TEXT_TRAINING_YN = 'Y' OR TEXT_TRAINING_SCORE > 0)

```
for idx, doc in enumerate(doc_text_list):
    train_image_path = doc['TEXT_IMAGE_PATH']
    train_image_nm   = doc['TEXT_IMAGE_PATH'].split('\\')[-1].split('.')
    image_ext  = train_image_nm[-1]
    text_updt_nm = ''
    if doc['TEXT_UPDT_NM'] is not None and doc['TEXT_UPDT_NM'] != '':
        text_updt_nm = doc['TEXT_UPDT_NM']
    else:
        text_updt_nm = doc['TEXT_NM']

    shutil.copyfile(train_image_path, step1_file_path + '\\train\\' + text_updt_nm + '_' + str(idx) + '.' + image_ext)
```

6.2. 학습대상 데이터가 너무 작은 경우 필요시 학습을 위한 글자이미지를 생성하여 학습에 이용하도록 한다.

```
SELECT * FROM TB_OCR_TRAIN_TEXT WHERE DEL_YN = 'N'
```

```
make_image_lib   = default_path + '\\TextRecognitionDataGenerator\\trdg\\run.py'
train_text_path  = learn_image_path + '\\train_text.txt'
make_image_train = "python {run} -c {count} -l ko -i {text} --output_dir {output}" \
                   .format(run=make_image_lib,count=data_count['train'], text=train_text_path, output=os.path.join(step1_file_path, 'train'))
```

6.3. 빠른 학습을 위한 학습 데이터셋을 생성한다.

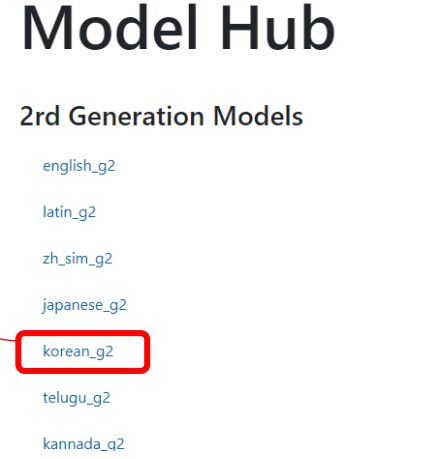
```
convert_image_lib  = default_path + '\\TRDG2DTRB\\convert.py'
make_lmdb_lib      = default_path + '\\deepTextRecognitionBenchmark\\create_lmdb_dataset.py'
convert_image_train = "python {convert} --input_path {input}\\train --output_path {output}\\train" \
                      .format(convert=convert_image_lib, input=step1_file_path, output=step2_file_path)
make_lmdb_train     = "python {lmdb} --inputPath {input}\\train --gtFile {input}\\train\\gt.txt --outputPath {output}\\train" \
                      .format(lmdb=make_lmdb_lib, input=step2_file_path, output=step3_file_path)
```

C:₩dev_project₩project_ocr₩training₩learning_image₩3₩train₩

6.4. 학습데이터셋로 글자인식 학습을 실행한다. (학습시 easyocr에서 제공하는 사전학습모델 사용)

```
model_train_lib    = default_path + '\\deepTextRecognitionBenchmark\\train.py'
pretrained_model   = default_path + '\\EasyOCR\\pre_trained_models\\korean_g2.pth'
exp_name = 'aiblue_ocr'
learning_model = os.path.join('saved_models',exp_name)
number_of_interation = 3000
interval_for_valid = 100
batch_size = 32
model_train  = 'python {train} \
                --exp_name   {exp_name} \
                --train_data {input}\\train \
                --valid_data {input}\\valid \
                --select_data / \
                --batch_ratio 1 \
                --Transformation None \
                --FeatureExtraction VGG \
                --SequenceModeling BiLSTM \
                --Prediction CTC \
                --input_channel 1 \
                --output_channel 256 \
                --hidden_size 256 \
                --batch_max_length {batch_max_length} \
                --saved_model {saved_model} \
                --FT \
                --workers 0 \
                --data_filtering_off \
                --num_iter {num_iter} \
                --valInterval {valInterval} \
                --batch_size {batch_size} '.format(train=model_train_lib, \
                                                   exp_name=exp_name, \
                                                   input=step3_file_path, \
                                                   num_iter=number_of_interation, \
                                                   valInterval=interval_for_valid, \
                                                   batch_size=batch_size, \
                                                   batch_max_length=batch_max_length, \
                                                   saved_model=pretrained_model)
```

https://www.jaided.ai/easyocr/modelhub/



6.5. 학습된 학습모델을 신규 문서 인식 및 추출에 사용할 수 있도록 해당 폴더에 복사한다.

```
model_storage_dir   = default_path + '\\model'
user_network_dir    = default_path + '\\user_network'
recog_network       = 'custom'
learn_image_path    = default_path + '\\learning_image'
shutil.copyfile(learning_model + '\\best_accuracy.pth', model_storage_dir + '\\' + recog_network +'.pth')
```

6.6. 학습 대상인 이미지에 대해 학습된 모델로 다시 인식하고 해당 결과를 DB에 업데이트 한다.

SELECT * FROM TB_OCR_DOC_TEXT WHERE TEXT_TRAINING_YN = 'Y' AND TEXT_TRAINING_NM IS NULL

```
model_storage_dir   = default_path + '\\model'
user_network_dir    = default_path + '\\user_network'
recog_network       = 'custom'
ocr_reader = easyocr.Reader(['en', 'ko'], gpu=True,
                            model_storage_directory=model_storage_dir,
                            user_network_directory=user_network_dir,recog_network=recog_network)
for idx, doc in enumerate(text_training_list):
    src_image = doc['TEXT_IMAGE_PATH']
    result = ocr_reader.readtext(src_image
                                 , detail='wordbeamsearch'
                                 , paragraph=False
                                 , min_size=5)
```

2

project_ocr.py