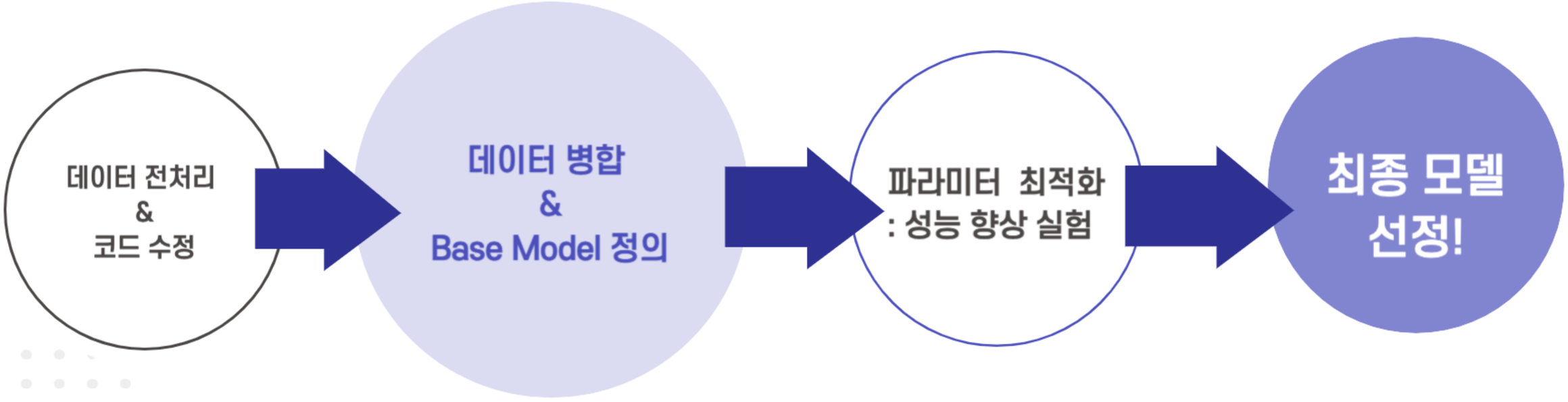
2023 비전 AI 와 비즈니스 - 최종발표

TEXT RECOGNITION WITH OCR

임형빈
배지환
류병하

CONTENTS

01.중간 발표 이후 프로젝트 흐름
: 한글 인식에 특화된 OCR 모델



3가지 서로 다른 데이터 셋 사용!
MENU & WILD text & 손글씨

01

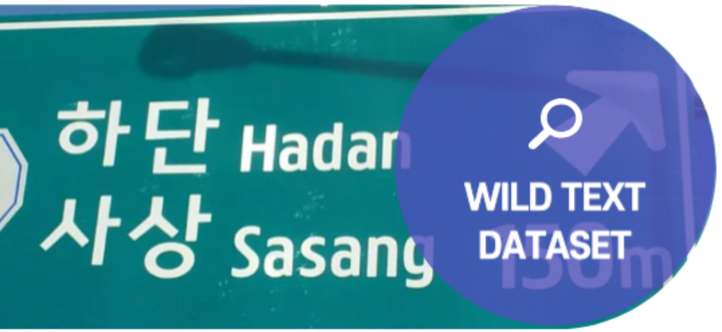
데이터 셋 전처리

DATASETS & Preprocessing

한글 문자 OCR DATASET

다양한 형태의 한글 문자 OCR

성별, 나이 연령대가 다른 125명의 작성자가
작성한 한글 필기체 데이터 1,101,225장
-> Bbox 추출 필요 X



WILD TEXT DATASET

Wild text 데이터셋

생활 전반에서 볼 수 있는 글씨 이미지 데이터
표지판, 식품 영양정보 등 ,10만 여장의 이미지
-> Bbox 별로 사진을 crop 해주는 과정

관광 음식메뉴판 데이터

10만개의 서로 다른 메뉴판 데이터
-> Bbox 별로 사진을 Crop
-> 사진별로 있는 Label을 합침

각기 다른 분야의 데이터를 활용해 성능 향상을 도모!

데이터 전처리

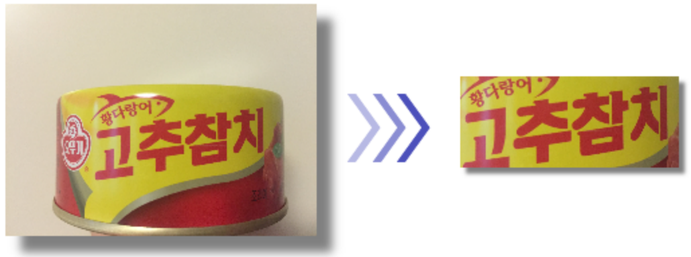
1. Train , val , test split (0.7 / 0.15 / 0.15)

```
ocr_good_files = os.listdir('/data/ocr/Goods/')
len(ocr_good_files) # 37220

random.shuffle(ocr_good_files)

n_train = int(len(ocr_good_files) * 0.7)
n_validation = int(len(ocr_good_files) * 0.15)
n_test = int(len(ocr_good_files) * 0.15)
```

2. 원본 데이터의 라벨을 활용하여, BBOX 형태로 텍스트 부분을 크롭



ex) bbox : [981,134,78,70]

3. 라벨을 LMDB 데이터로 변환 & 모델에 맞는 파일 구성

- deep-text-recognition-benchmark
  - data
    - test
    - train
    - validation
    - gt_test.txt
    - gt_train.txt
    - gt_validation.txt

02

Base Model 정의를 위한 학습 진행

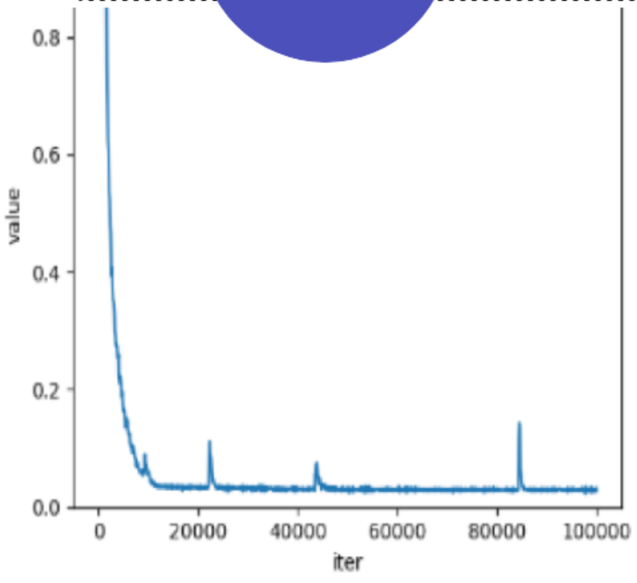
각 데이터 학습을 통해 코드 가동성, 데이터 오류 여부 파악
Feature Extractor 선정을 위한 학습

각 데이터로 학습 진행

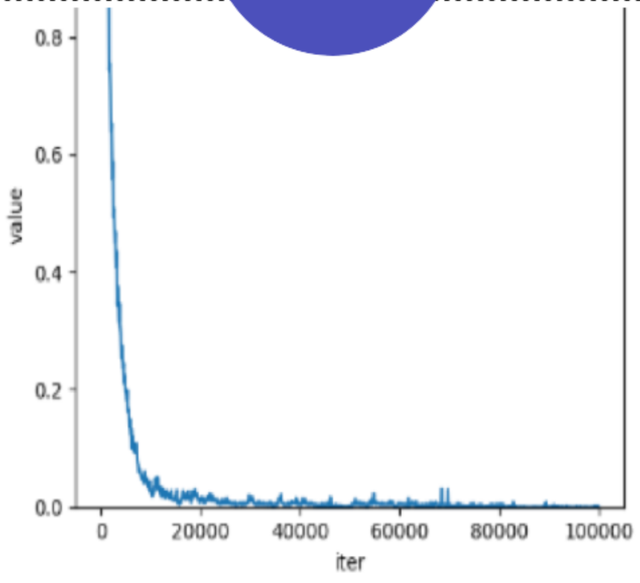
코드 가동성 , 데이터 오류 여부 파악

< Train Loss >

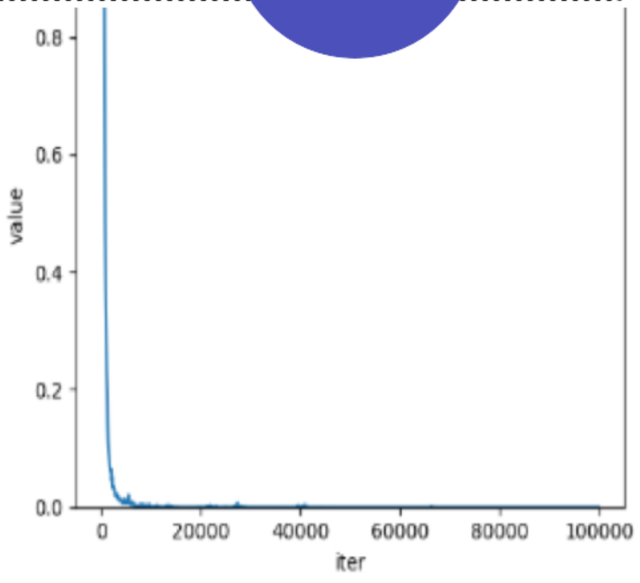
관광음식메뉴판 DATASET



WILD TEXT DATASET



한글 문자 OCR DATASET



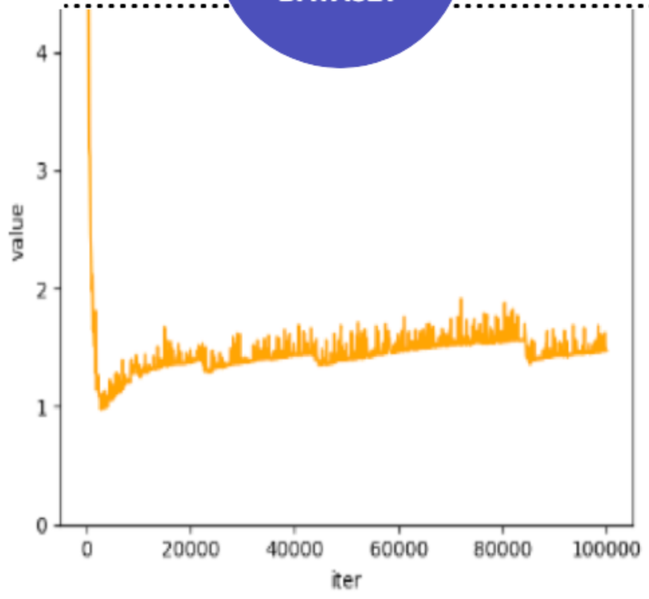
TPS + ResNet + BiLSTM + CTC , epoch 100,000

각 데이터로 학습 진행

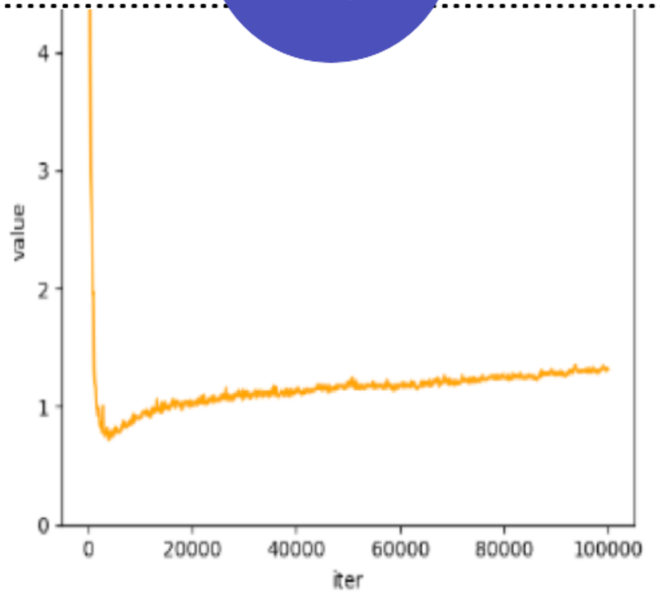
코드 가동성 , 데이터 오류 여부 파악

< Validation Loss >

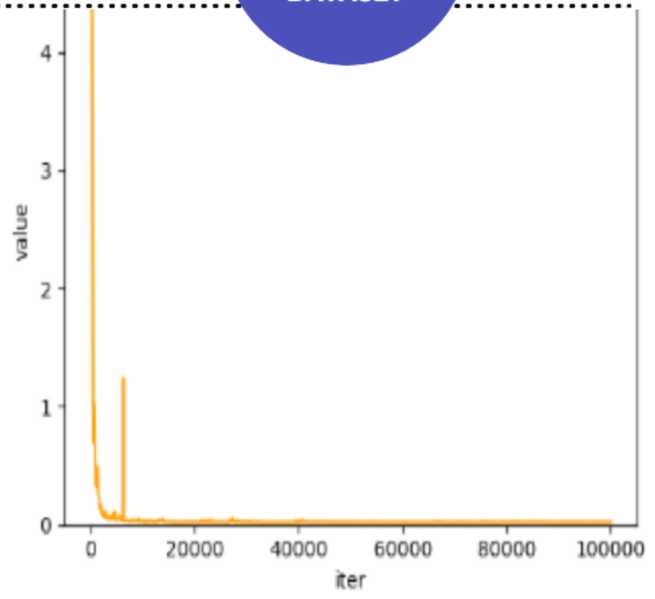
관광음식메뉴판 DATASET



WILD TEXT DATASET



한글 문자 OCR DATASET



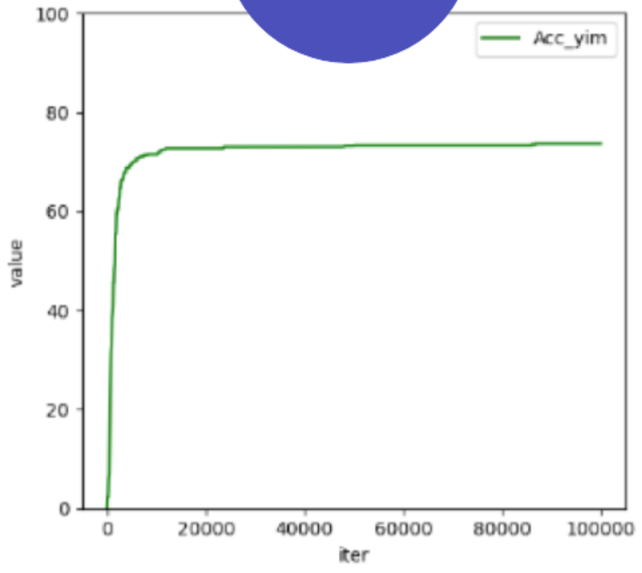
TPS + ResNet + BiLSTM + CTC , epoch 100,000

각 데이터로 학습 진행

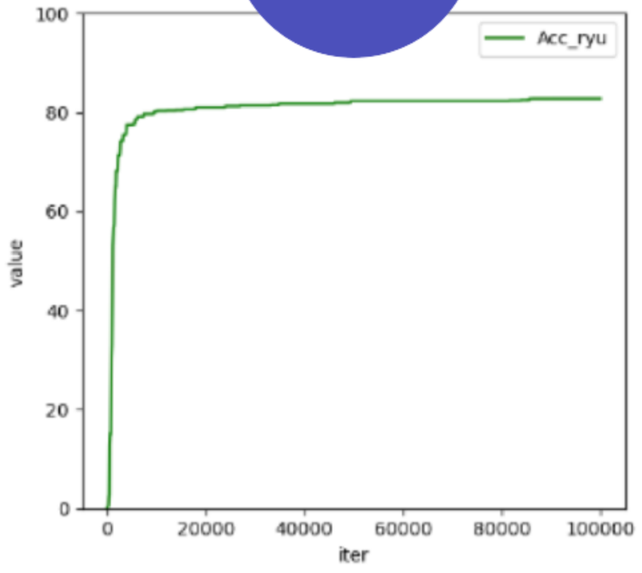
코드 가동성 , 데이터 오류 여부 파악

< Accuracy >

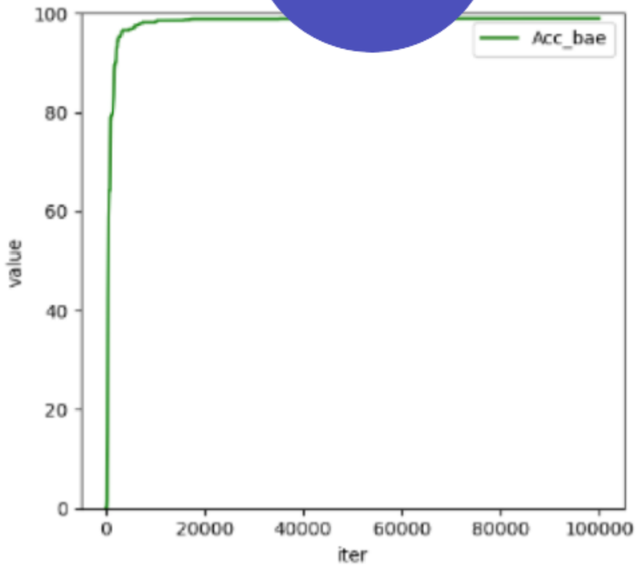
관광음식메뉴판 DATASET



WILD TEXT DATASET



한글 문자 OCR DATASET



TPS + ResNet + BiLSTM + CTC , epoch 100,000

Feature Extractor 선정을 위한 학습

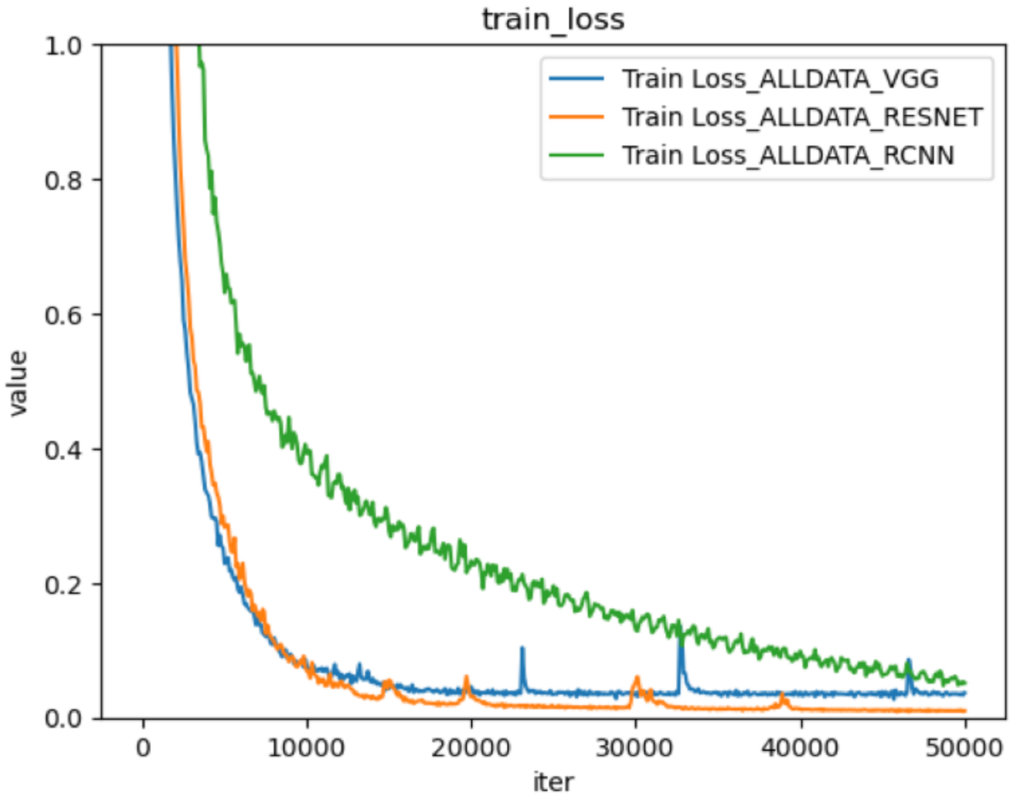
- 각각 10만개의 데이터 활용 : 총 30만
  -> Train : 21만 / Val,Test : 각 4.5만

VGG vs. ResNet vs. RCNN

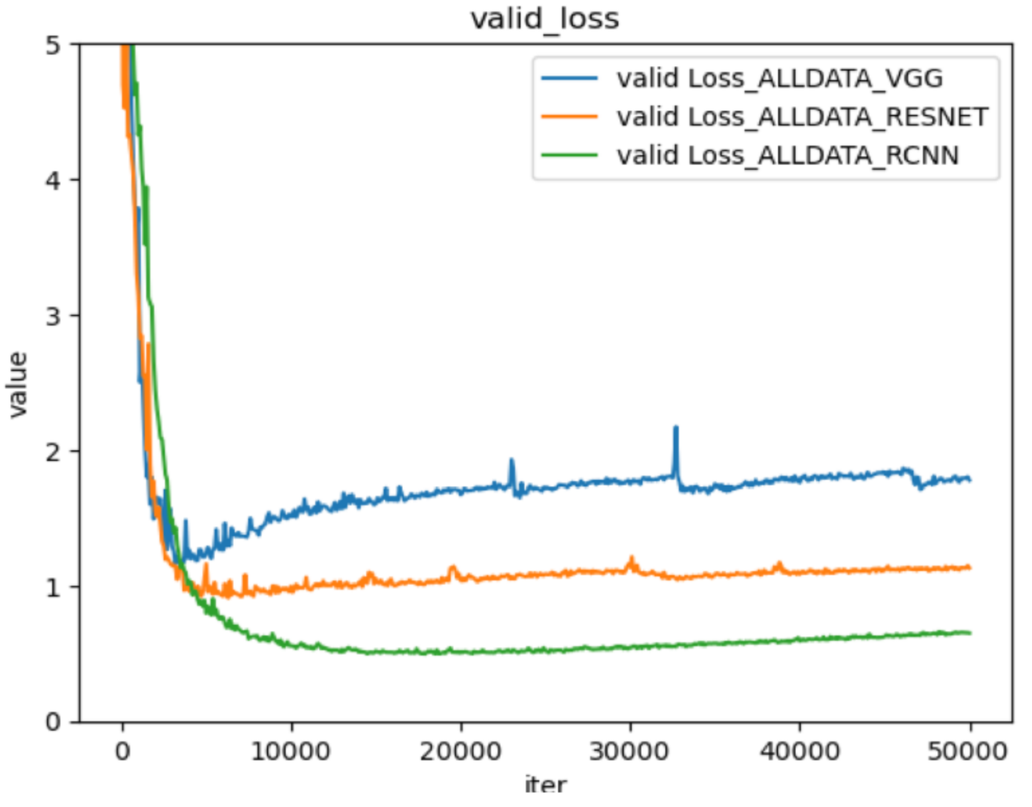
데이터 병합 후 학습

Feature Extractor 선정을 위한 학습

Train Loss

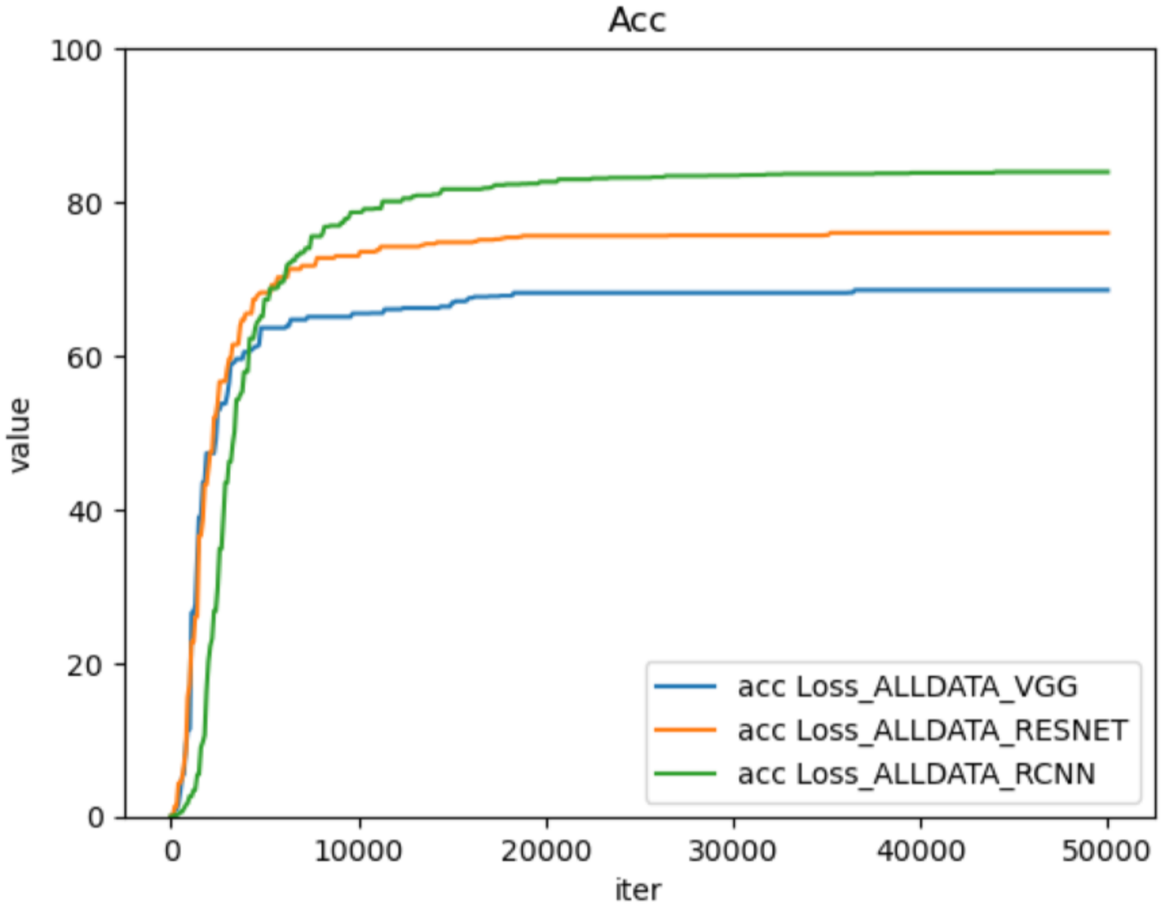


Valid Loss



데이터 병합 후 학습

Feature Extractor 선정을 위한 학습



Accuracy

VGG : 68.549

RESNET : 76.218

RCNN : 83.923 - Test Acc : 86.491

RCNN을 Feature Extractor로 선정!

03

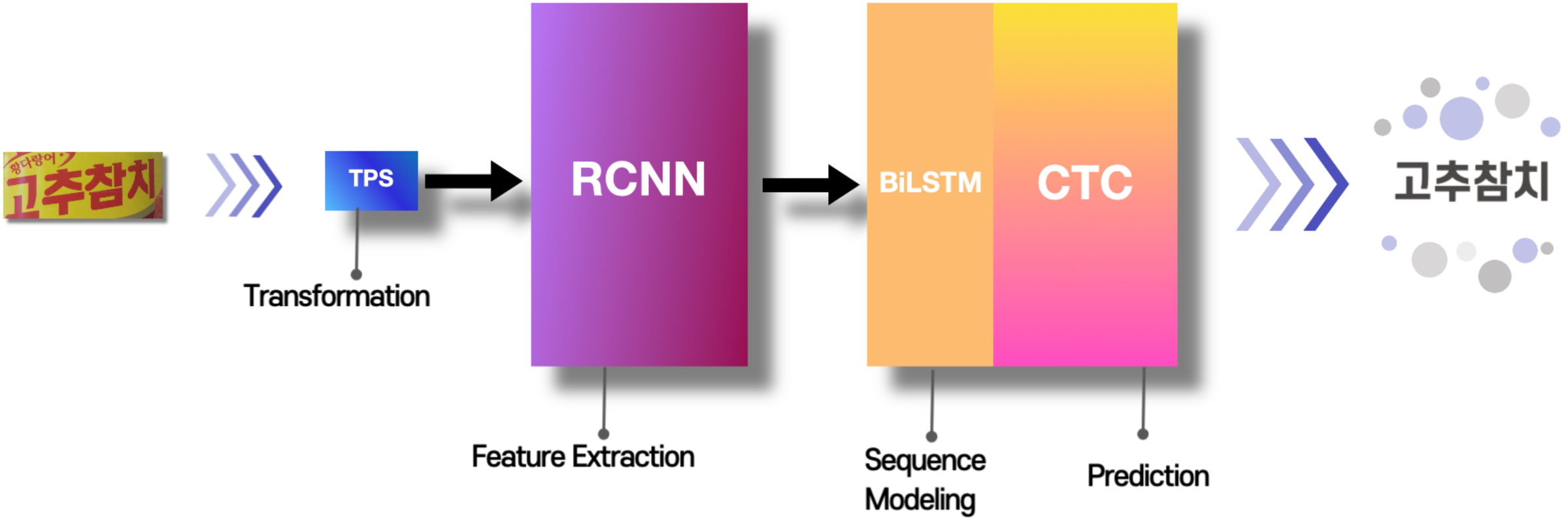
성능 향상 실험

img W, img H 를 이용한 해상도 조정

batch size 조정

optimizer 선정 , Hyper parameter조정

베이스 모델 정의

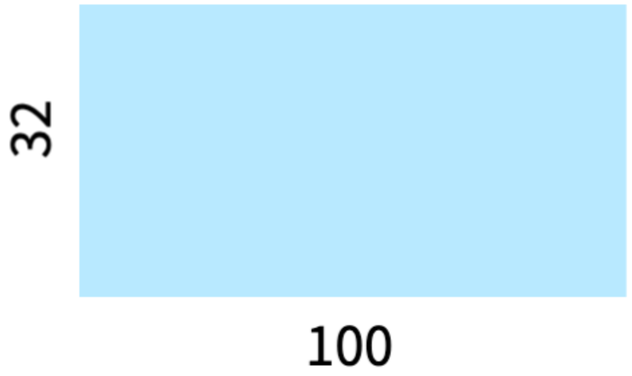


성능 향상을 위한 시도 시작!

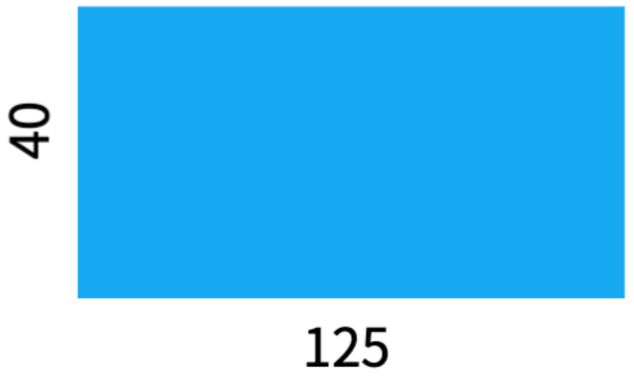
성능 향상 실험

img H, img W - 해상도

Model's Default img H , W



1.25 size scaled up



2 size scaled up

해상도 상승을 통해 input 이미지의 정보량을 up!
Feature Extraction 성능 향상을 도모

성능 향상 실험

batch size

Model's Default batch size

194

Modified batch size

64

Local Minimum 에서 빠져나올 수 없는 경우를 방지하기 위해
batch size 를 작게 조정

성능 향상 실험

Optimizer Hyperparameter

다양한 스타일과 변형을 가지고 있는 데이터 :
그래디언트의 분포가 불균형할 것이라는 예상
상대적으로 강건한 **Adadelta** 선정

Default

Gradient Clip = 5
rho = 0.95

Modified

Gradient Clip = 3
rho = 0.99

Local Minimum 에서 학습이 종료되지 않도록
rho 값과, Gradient clip Threshold 를 조정

성능 향상 실험

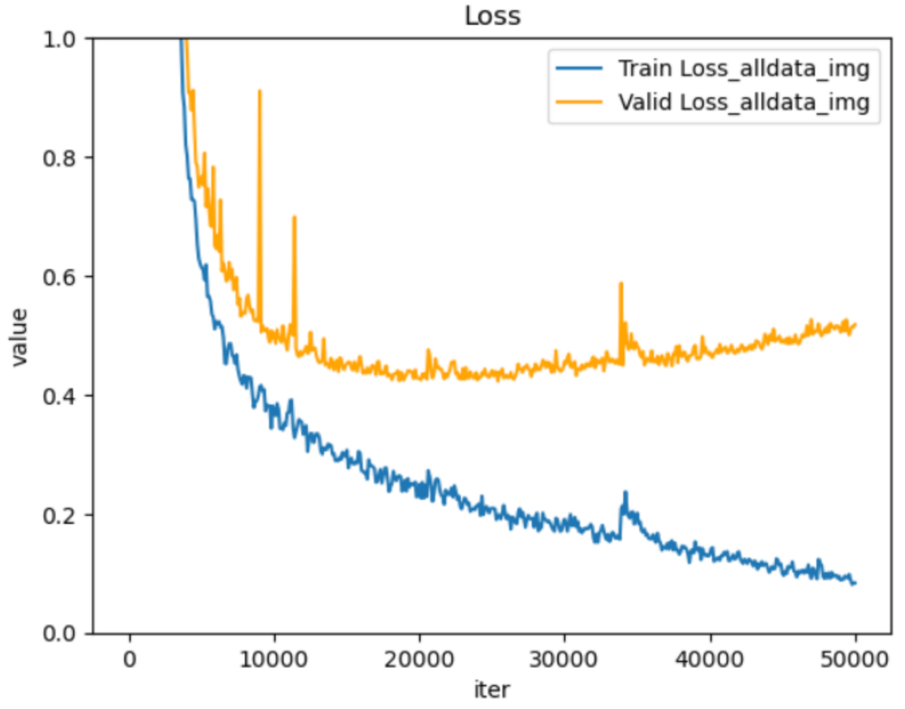
비교실험

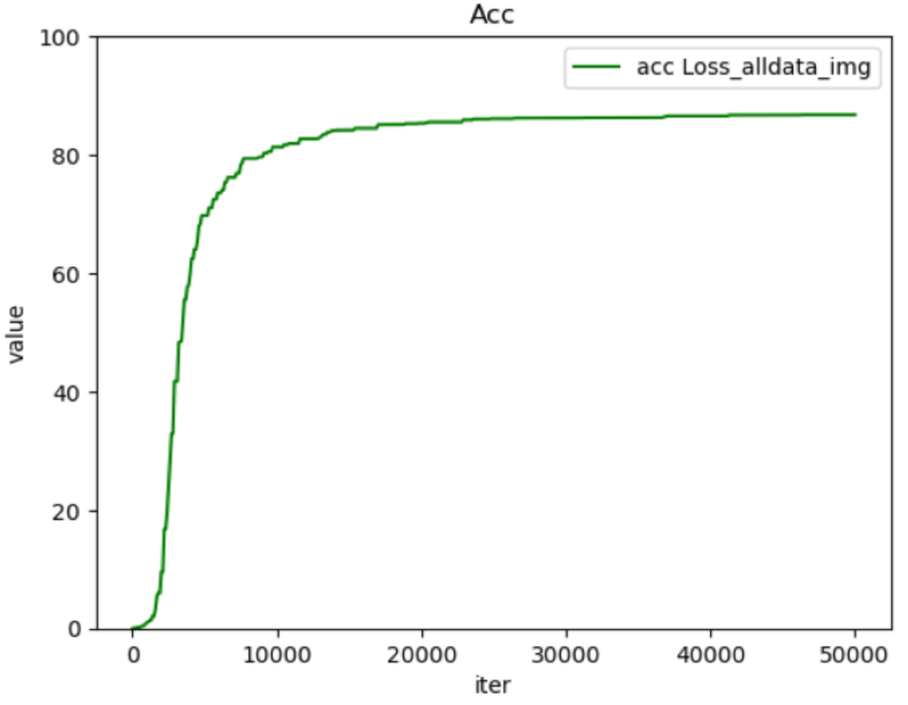
#	Trans-Feat-Seq-Pred	Batch_size	imgHxW	grad_clip	rho	Acc %	Time hour
1	TPS-RCNN-BiLSTM-CTC	192	32x100	5	0.95	86.5	26.4
2	TPS-RCNN-BiLSTM-CTC	64	32x100	5	0.95	86.8	21
3	TPS-RCNN-BiLSTM-CTC	192	32x100	3	0.99	87.5	27.1
4	TPS-RCNN-BiLSTM-CTC	192	40x125	5	0.95	89.1	22.3
5	TPS-RCNN-BiLSTM-CTC	64	64x200	5	0.95	**90.3**	37.4

성능이 비교적 높은 4,5번 중 Time hour 가 적은 4번 모델을 최종 선정

최종 모델 선정

TPS + RCNN + BiLSTM + CTC (40 * 125 scale)





Train loss : 0.07

Valid loss : 0.54

최종 Acc : 89.1

wild_data recognition을 원활하게 할 수 있는 전처리 방법론을 사용하면 성능 향상이 기대됨