BIG DATA / AI
경진대회

김태인, 김동근

Competition 1

화물 이동 차량의 영상 분석 : 화물 적재 유/무 판단(classification)

01. 학습 과정

```
train_gen = ImageDataGenerator(rescale=1. / 255,
                               rotation_range=20, width_shift_range=0.2,
                               height_shift_range=0.2, shear_range=0.2, zoom_range=0.2,
                               brightness_range=[0.8, 1.2],
                               validation_split=0.2)
validation_gen = ImageDataGenerator(rescale=1. / 255, validation_split=0.2)
```

```
train_gen.flow_from_directory(train_path,
                              batch_size=BATCH_SIZE,
                              shuffle=True,
                              color_mode='rgb',
                              target_size=(IMG_WIDTH, IMG_HEIGHT),
                              subset='training',
                              class_mode='binary',
                              )
```

ImageDataGenerator를 통해 이미지 데이터의 부족 현상을 해결

01. 학습 과정

```
label = [0] * zro + [1] * one
class_weight = compute_class_weight(class_weight = 'balanced', classes=np.unique(label), y=label)
```

class weight를 통해 이미지 데이터 불균형 문제 해결

01. 학습 과정

```
base_model = ResNet152(weights='imagenet', include_top=False, input_shape=(IMG_WIDTH, IMG_HEIGHT, 3))

model = Sequential()
model.add(base_model)
model.add(GlobalAveragePooling2D())
model.add(Dropout(0.25))
model.add(Dense(1, activation='sigmoid'))
model.summary()
```

```
model.compile(optimizer=tf.keras.optimizers.Adam(2e-5),
              loss='binary_crossentropy',
              metrics=['accuracy'])
```

epoch(5)	val_accuracy(0.2) (%)	time(sec)	Input size	Batch size
MobileNetV2	0.8184	0:10:15.913060	224 x 224	32
EfficientNetB0	0.5042	0:10:22.713614	224 x 224	32
ResNet152	0.8322	0:20:51.688205	224 x 224	32
NasNetLarge	0.9228	1:50:02.982123	331 x 331	16

여러번의 전이 학습을 통한 모델 선정 및 파인 튜닝

01. 학습 과정

Resnet152 epoch	train_loss	train_accuracy	val_loss	val_accuracy
1	0.3017	0.8591	0.7562	0.5410
2	0.1236	0.9510	0.7356	0.5892
3	0.0730	0.9740	0.4577	0.7901
4	0.0497	0.9817	0.4046	0.8350
5	0.0391	0.9845	0.4752	0.8532
6	0.0376	0.9846	0.4236	0.8670
7	0.0281	0.9892	0.5389	0.8641
8	0.0237	0.9911	0.5318	0.8524
9	0.0246	0.9909	0.4345	0.8674

train 셋 훈련 결과

01. 학습 과정

```
checkpoint = ModelCheckpoint(save_file_name, monitor='val_loss', verbose=1, save_best_only=True, mode='auto')

earlystopping = EarlyStopping(monitor='val_loss', patience=5)
```

학습 과정에 checkpoint와 earlystopping 진행

02. 학습 결과

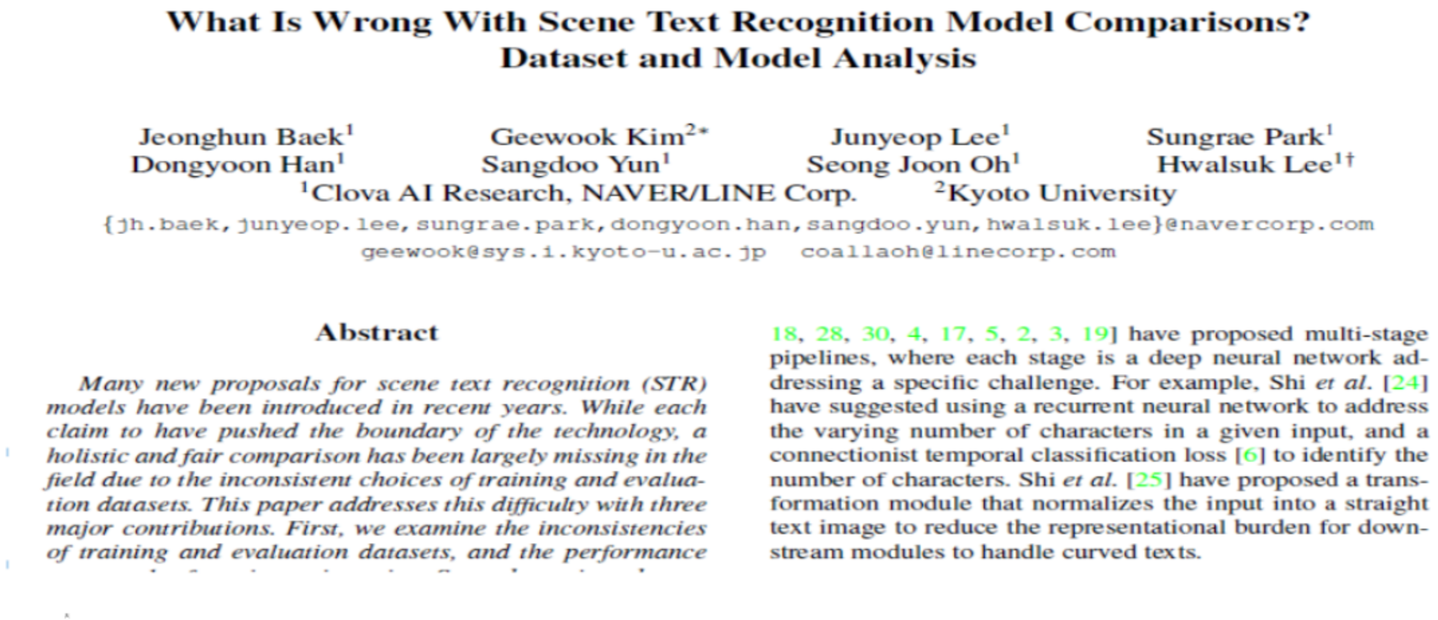
	Image list	GT	Prediction	accuracy
0	0\empty_V1.jpg	0	[0]	TRUE
1	0\empty_V10.jpg	0	[1]	FALSE
2	0\empty_V2.jpg	0	[0]	TRUE
3	0\empty_V3.jpg	0	[0]	TRUE
4	0\empty_V4.jpg	0	[0]	TRUE
5	0\empty_V5.jpg	0	[0]	TRUE
6	0\empty_V6.jpg	0	[1]	FALSE
7	0\empty_V7.jpg	0	[1]	FALSE
8	0\empty_V8.jpg	0	[0]	TRUE
9	0\empty_V9.jpg	0	[0]	TRUE
10	1\loaded_V1.jpg	1	[1]	TRUE
11	1\loaded_V10.jpg	1	[1]	TRUE
12	1\loaded_V2.jpg	1	[1]	TRUE
13	1\loaded_V3.jpg	1	[1]	TRUE
14	1\loaded_V4.jpg	1	[1]	TRUE
15	1\loaded_V5.jpg	1	[1]	TRUE
16	1\loaded_V6.jpg	1	[1]	TRUE
17	1\loaded_V7.jpg	1	[1]	TRUE
18	1\loaded_V8.jpg	1	[1]	TRUE
19	1\loaded_V9.jpg	1	[1]	TRUE
20	Average Accuracy	0.85		
21	Inference Speed(ms)	300.6310701		

최종 모델에 대한 Average accuracy : 85%

Competition 2

스마트 강재 적치장 구축 : 강재 문자 인식(Optical Character Recognition)

01. 학습 과정



What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis

Jeonghun Baek[1] Geewook Kim[2*] Junyeop Lee[1] Sungrae Park[1]
Dongyoon Han[1] Sangdoo Yun[1] Seong Joon Oh[1] Hwalsuk Lee[1†]
[1]Clova AI Research, NAVER/LINE Corp. [2]Kyoto University
{jh.baek, junyeop.lee, sungrae.park, dongyoon.han, sangdoo.yun, hwalsuk.lee}@navercorp.com
geewook@sys.i.kyoto-u.ac.jp coallaoh@linecorp.com

Abstract

Many new proposals for scene text recognition (STR) models have been introduced in recent years. While each claim to have pushed the boundary of the technology, a holistic and fair comparison has been largely missing in the field due to the inconsistent choices of training and evaluation datasets. This paper addresses this difficulty with three major contributions. First, we examine the inconsistencies of training and evaluation datasets, and the performance

18, 28, 30, 4, 17, 5, 2, 3, 19] have proposed multi-stage pipelines, where each stage is a deep neural network addressing a specific challenge. For example, Shi *et al.* [24] have suggested using a recurrent neural network to address the varying number of characters in a given input, and a connectionist temporal classification loss [6] to identify the number of characters. Shi *et al.* [25] have proposed a transformation module that normalizes the input into a straight text image to reduce the representational burden for downstream modules to handle curved texts.

논문 자료 참고하여 Naver ocr pretrained 모델 사용

TPS-ResNet-BiLSTM-Attn-case-sensitive.pth

01. 학습 과정

```
# OCR 폴더로 이동 (ubuntu)
cd ~/deep-text-recognition-benchmark

# window
cd deep-text-recognition-benchmark

# demo.py 실행 (UBUNTU)
CUDA_VISIBLE_DEVICES=0 python3 demo.py --Transformation TPS --FeatureExtraction ResNet --SequenceMod
```

pretrained 모델 추론 과정

01. 학습 과정

```
# create train_lmdb file
python3 create_lmdb_dataset.py --inputPath data/training --gtFile data/train_gt.txt --outputPath data

# create valid_lmdb file
python3 create_lmdb_dataset.py --inputPath data/validation --gtFile data/valid_gt.txt --outputPath d
```

커스텀 데이터셋 제작

1. deep-text-recognition-benchmark 폴더 안에 lmdb 파일을 담을 ‘predata' 폴더 생성
2. predata 폴더 안에 'training', 'validation' 폴더 생성
3. training, validation 폴더 안에 학습 및 검증용 이미지 넣기
4. data 폴더 안에 _gt.txt 파일 생성 후 라벨링하기
5. 이미지와 라벨링 텍스트 파일을 lmdb 형태로 변환

01. 학습 과정

```
---------------------------------------------------------------------------
dataset_root: predata/train
opt.select_data: ['01', '02', '03']
opt.batch_ratio: ['0.482', '0.482', '0.036']
---------------------------------------------------------------------------
dataset_root:    predata/train dataset: 01
sub-directory:       /01         num samples: 3438
num total samples of 01: 3438 x 1.0 (total_data_usage_ratio) = 3438
num samples of 01 per batch: 64 x 0.482 (batch_ratio) = 31
---------------------------------------------------------------------------
dataset_root:    predata/train dataset: 02
sub-directory:       /02         num samples: 3694
num total samples of 02: 3694 x 1.0 (total_data_usage_ratio) = 3694
num samples of 02 per batch: 64 x 0.482 (batch_ratio) = 31
---------------------------------------------------------------------------
dataset_root:    predata/train dataset: 03
sub-directory:       /03         num samples: 270
num total samples of 03: 270 x 1.0 (total_data_usage_ratio) = 270
num samples of 03 per batch: 64 x 0.036 (batch_ratio) = 2
---------------------------------------------------------------------------
Total_batch_size: 31+31+2 = 64
---------------------------------------------------------------------------
```

학습데이터 나누어 학습 진행

01. 학습 과정

```
--------------------------------------------------------------------------------
[500/500] Train loss: 0.04392, Valid loss: 0.08413, Elapsed_time: 184.42395
Current_accuracy : 90.952, Current_norm_ED  : 0.97
Best_accuracy    : 91.111, Best_norm_ED     : 0.97
--------------------------------------------------------------------------------
Ground Truth            | Prediction            | Confidence Score & T/F
--------------------------------------------------------------------------------
9012H15B17              | 9012H15B17            | 0.5084          True
983S11B41               | 983S11B41             | 0.9466 True
3801                    | 3801                  | 0.9990       True
4303                    | 4303                  | 0.9929       True
983S17T54               | 983S17T54             | 0.9740 True
--------------------------------------------------------------------------------
```

학습 best accuracy : 91.11%

01. 학습 과정

```
---------- Options -------------
exp_name: TPS-ResNet-BiLSTM-Attn-Seed1111
train_data: predata/train
valid_data: predata/val
manualSeed: 1111
workers: 0
batch_size: 64
num_iter: 500
valInterval: 50
saved_model: models/TPS-ResNet-BiLSTM-Attn-case-sensitive.pth
FT: True
adam: False
lr: 1
beta1: 0.9
rho: 0.95
eps: 1e-08
grad_clip: 5
baiduCTC: False
select_data: ['01', '02', '03']
batch_ratio: ['0.482', '0.482', '0.036']
total_data_usage_ratio: 1.0
batch_max_length: 25
imgH: 32
imgW: 100
rgb: False
character: 0123456789abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ!"#$%&'()*+,-./:;<=>?@[₩]^_`{|}·
sensitive: True
PAD: False
data_filtering_off: True
Transformation: TPS
FeatureExtraction: ResNet
SequenceModeling: BiLSTM
Prediction: Attn
num_fiducial: 20
input_channel: 1
output_channel: 512
hidden_size: 256
num_gpu: 1
num_class: 96
----------------------------------------
```

parameter 값

02. 학습 결과

	Image list	GT	Prediction	accuracy
0	01/000000C	6601	6601	100.00%
1	01/000006	3603	3603	100.00%
2	01/000006	1003	1003	100.00%
3	01/000007	9102	9102	100.00%
4	01/000009	9901	9901	100.00%
5	01/00001C	2801	2801	100.00%
6	01/000015	7702	7702	100.00%
7	01/00002C	1003	1003	100.00%
8	01/000021	4601	4601	100.00%
9	01/000021	2003	2003	100.00%
10	02/001552	995B17R0	995B17R0	100.00%
11	02/001554	8093T42B	8093T42B	100.00%
12	02/001554	8093D41U	8093D41U	100.00%
13	02/001557	8108F51F8	8108F51F8	100.00%
14	02/001557	8109B37A	8109B37A	100.00%
15	02/00155E	8108F51F8	8108F51F8	100.00%
16	02/00155E	8109B37A	8109B37A	100.00%
17	02/00155E	8057D17L	8057D17L	100.00%
18	02/001559	C1051BAA	C1051BAA	100.00%
19	02/001559	C1051BAA	C1051B8A	80.00%
20	03/000623	8109S17H	8109S17H	100.00%
21	03/000623	FB875209-	FB875209-	100.00%
22	03/000624	8094E31J1	8094E31J1	100.00%
23	03/000624	FC359707-	FC359707-	100.00%
24	03/000624	995E21F64	995E21F64	100.00%
25	03/000624	FC360027-	FC360027-	100.00%
26	03/000624	8109E51H	8109E51H	100.00%
27	03/000624	FC360144-	FC360144-	100.00%
28	03/000624	81B9A322	8109A322	90.00%
29	03/000624	FB871275-	FB871275-	100.00%
30	Average A	99		
31	Inference	0.138873		

최종 모델에 대한 Average accuracy : 99%

감사합니다.