

# OCR 엔진 기반 분류기 애드온 결합을 통한 이미지 내부 텍스트 인식 성능 향상

## Scene Text Recognition Performance Improvement through an Add-on of an OCR based Classifier

채 호 열\*, 석 호 식\*\*★

Ho-Yeol Chae\*, Ho-Sik Seok\*\*★

### Abstract

An autonomous agent for real world should be able to recognize text in scenes. With the advancement of deep learning, various DNN models have been utilized for transformation, feature extraction, and predictions. However, the existing state-of-the art STR (Scene Text Recognition) engines do not achieve the performance required for real world applications. In this paper, we introduce a performance-improvement method through an add-on composed of an OCR (Optical Character Recognition) engine and a classifier for STR engines. On instances from IC13 and IC15 datasets which a STR engine failed to recognize, our method recognizes 10.92% of unrecognized characters.

### 요 약

일상 환경에서 동작하는 자율 에이전트를 구현하기 위해서는 이미지나 객체에 존재하는 텍스트를 인식하는 기능이 필수적이다. 주어진 이미지에 입력 변환, 특성 인식, 워드 예측을 적용하여 인식된 텍스트에 존재하는 워드를 출력하는 과정에 다양한 딥러닝 모델이 활용되고 있으며, 딥뉴럴넷의 놀라운 객체 인식 능력으로 인식 성능이 매우 향상되었지만 실제 환경에 적용하기에는 아직 부족한 점이 많다. 본 논문에서는 인식 성능 향상을 위하여 텍스트 존재 영역 감지, 텍스트 인식, 워드 예측의 파이프라인에 OCR 엔진과 분류기로 구성된 애드온을 추가하여 기존 파이프라인이 인식하지 못한 텍스트의 인식을 시도하는 접근법을 제안한다. IC13, IC15의 데이터 셋에 제안 방법을 적용한 결과, 문자 단위에서 기존 파이프라인이 인식하는데 실패한 문자의 최대 10.92%를 인식함을 확인하였다.

*Key words* : Scene text recognition (STR), Optical character recognition (OCR), Text detection, Deep learning, Machine learning

---

\* Dpet. of Computer and Communications Engineering, Kangwon National University

\*\* Dept. of Computer Science and Engineering, Kangwon National University

★ Corresponding author

E-mail : hsseok@kangwon.ac.kr, Tel : +82-33-250-6381

※ Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2018R1D1A1B07047156).

Manuscript received Nov. 26, 2020; revised Dec. 23, 2020; accepted Dec. 28, 2020.

This is an Open-Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

## I. 서론

일상 환경에서 자율적으로 동작하는 에이전트를 구현하려는 경우, 이미지나 객체에 존재하는 텍스트를 인식하는 기능(STR, Scene Text Recognition)이 요구된다. 그러나 일반적인 OCR(Optical Character Recognition) 엔진의 동작 환경과 틀리게 이미지나 객체에 존재하는 텍스트의 인식을 시도할 경우 배경의 급격한 변화, 폰트 및 글자 크기의 다양성 등으로 인하여 텍스트 인식의 성능을 높이기 어렵다 [1, 2].

기계학습을 활용한 초창기 STR 연구에서는 텍스트의 구조적인 특징을 활용하거나 다양한 규칙을 정의하여 텍스트를 인식하고자 시도하였다[3, 4]. 그러나 객체 인식 태스크에 대한 딥러닝의 놀라운 성능이 보고되면서 DNN(Deep Neural Network) 모델을 활용하여 STR 태스크를 해결하려는 연구가 활발히 진행되고 있다([2] 참조).

DNN 모델에 기반한 STR 엔진들은 놀라운 텍스트 인식 성능을 보이고 있지만[5], 객체 인식과 비교하였을 때 STR 문제에서 보고된 성능은 실제 응용에 적용하기에 아직 부족하다. 이와 같은 문제를 해결하기 위하여 본 논문에서는 STR 분야에서 가장 최고의 성능을 발휘하는 모듈들 위에 OCR 엔진과 분류기(Classifier)를 추가한 후 DNN 모델에 기반한 STR 엔진이 인식하지 못한 텍스트의 인식을 시도하는 접근법을 소개한다. 제안 방법에서는 이미지에 존재하는 텍스트의 위치 탐색 및 탐색된 이미지 인식 과정에 기존 STR엔진을 적용한 후, 인식에 실패한 텍스트 인식을 위하여 필기체 데이터로 훈련된 OCR 엔진을 적용하여 문자를 분류하는 접근법을 활용한다. STR 엔진 성능 평가에 활용되는 데이터 셋에 대하여 제안 방법의 성능을 확인한 결과, 기존 STR 엔진이 인식하지 못한 문자 중 10.92%의 문자를 인식함을 확인하였다.

본 논문의 구성은 다음과 같다. STR의 일반적인 접근법을 이해할 수 있도록 II장에서는 STR에 대한 주요 연구를 소개한다. 제안 방법은 III장에서 설명되며, IV장에서 구체적인 실험 결과를 보고한다. V장에서는 제안 방법의 의의 및 추후 연구 방향을 설명한다.

## II. 관련연구

이미지나 객체에 존재하는 텍스트를 인식하기 위한 다양한 방법들이 시도되었다[1, 2]. DNN이 소개되기 전에는 텍스트에 존재하는 구조를 활용하여 인식하려는 시도가 활발히 수행되었다. 이 경우 텍스트와 텍스트가 아닌 구역을 구분하기 위하여 다양한 특성을 추출해야 한다. 이미지로부터 MDF (Mean difference feature), SD(Standard deviation), HoG (Histogram of oriented Gradient)를 이용하여 특성을 선정한 후 분류기 훈련에 적용하는 방법이 가능하며[6], 화소 강도(Pixel intensity)를 SVM(Support Vector Machine)과 CAMSHIFT(Continuously Adaptive Mean Shift Algorithm)에 적용하는 것이 가능함도 보고되었다[7]. 그 외에도 컬러 밀집도 차이와 기울기 정보(Gradient map)를 활용한 접근법 역시 시도되었다[8].

SWT(Stroke Width Transform)는 획의 폭이 거의 일정하다는 텍스트의 특징에 주목한 방법으로 픽셀 당 획의 폭에 해당하는 영역을 계산한다. 이 과정에서는 먼저 가장자리를 인식한 후 각 픽셀의 경사 방향을 고려하여 픽셀을 찾게 된다[9]. 이미지에 존재하는 텍스트를 인식하는 과정에서 SWT와 함께 널리 사용되는 컴퍼넌트가 MSER(Maximally Stable Extremal Region)이다[10]. MSER은 영역과 극치(Extreme)의 인텐서티(Intensity)를 고려하여 선정되며 극치의 경계로부터 도출된 표현에서 텍스트 인식을 시도할 수 있다[10, 11]. SWT와 MSER은 이미지나 객체에 존재하는 텍스트를 인식하는 과정에 기본적인 컴퍼넌트로 널리 사용되었다[12, 13].

딥러닝의 등장과 함께 이미지나 객체에 존재하는 텍스트 인식 과정에도 DNN이 널리 사용되기 시작하였다. CNN(Convolutional Neural Network)의 놀라운 객체 인식 능력으로 텍스트 인식에 CNN이 적용되고 있는데, [14]에서는 CRF (Conditional Random Field)와 CNN을 통합한 구조를 제안하였다. [15]에서는 워드 어노테이션(Word annotation)과 응답 맵(Response map)을 결합하여 문자의 중심부를 찾은 후 문자 모델의 학습을 시도하는 DNN 모델을 소개하였다.

이미지에서 텍스트를 인식하는 과정은 여러 단계를 거쳐 결과가 획득되는 일종의 파이프라인으로 취급될 수 있는데, 이미지 인식의 프레임워크를 (1)

변환(Transformation) (2) 특성 획득(Feature extraction) (3) 시퀀스 모델링(Sequence modelling) (4) 예측으로 구분할 수 있다[5]. 각 단계별로 다양한 모델이 사용되는데, 특성 획득을 위해 SWT나 MSER외에도 STN(Spatial Transformer Network)를 활용할 수 있다[16]. STN은 입력 특성 맵의 공간 변환을 위하여 Localisation network, grid generator, sampler를 결합한 형태로 CNN과 결합하여 특성맵의 변환에 사용될 수 있다. 특성 획득 과정에는 VGG[17], ResNet[18], RCNN[19] 등 잘 알려진 DNN 모델을 적용하고 있다[20, 21]. 시퀀스 모델링 및 예측 단계에서는 BiLSTM(Bidirectional LSTM)과 CTC(Connectionist temporal classification), Attn(Attention-based Sequence Prediction) 등이 활용되고 있다 [22, 23].

이미지나 객체에 존재하는 텍스트를 인식하는 엔진의 성능 비교를 위하여 공통된 데이터 셋이 사용되는 경우가 많다. MJSynth[24]와 SynthText[25]은 데이터 합성 단계에서 레이블이 부착되기 때문에 인식 엔진의 훈련 과정에서 흔히 사용된다. 평가를 위해서는 IIT5K-Words[26], SVT(Street View Text) [27], ICDAR2003[28], ICDAR2013[29], ICDAR2015[30], OSTD(The Oriented Scene Text Database)[31]와 같은 데이터 셋들이 흔히 활용된다.

### III. 제안 방법

#### 1. 문제 정의 및 데이터

본 논문에서는 STR 파이프라인이 인식하지 못한 문자들을 인식하기 위하여 OCR엔진과 분류기로 구성된 애드온을 적용하고자 한다. Deep-text-benchmark[5]에서 제일 좋은 성능을 보여주었던 TPS-ResNet-BiLSTM-Attn을 기본 STR 파이프라인으로 선정하고, [5]에서 보고된 실험 결과에서 다른 데이터 셋 대비 가장 인식 성능이 좋았던 IC13[29]과 가장 인식 성능이 나빴던 IC15[30]에 대하여, 해당 STR 파이프라인이 인식하지 못한 문자 인식을 시도하였다. 이를 위해 해당 STR 파이프라인에 OCR 엔진과 분류기 층을 추가하였으며, 필기체 데이터(EMNIST [32]) 및 MJSynth[14] 데이터를 사용하여 추가된 OCR엔진과 분류기를 훈련하였다.

IC13과 IC15 데이터 셋에 대한 STR 파이프라인의 인식 성능은 표 1과 같다. 표 1에서 제시한 인식

Table 1. Recognition performance by TPS-ResNet-BiLSTM-Attn.

표 1. TPS-ResNet-BiLSTM-Attn 인식 성능

Dataset	Image	Accuracy	Number of unrecognized words	Number of unrecognized characters
IC13	1015	92.3%	134	358
IC15	2077	71.8%	782	1125

실패 단어 중 대부분의 경우는 단어를 구성하는 문자의 일부를 인식하지 못한 경우이다. 따라서 본 논문에서는 STR 파이프라인이 인식하지 못한 문자를 인식하여 인식 성능을 높이고자 한다.

우리가 제안하는 애드온은 OCR엔진과 분류기로 구성되어 있다. 애드온의 인식 대상은 표 1에서 “Unrecognized characters”로 표시된 문자들로, 인식에 실패한 개별 문자로 구분한 데이터 셋인 Test Bad case Character set(TeBc)을 구성하였다. OCR 엔진의 훈련을 위하여 EMNIST[32] 데이터 셋과 분류기의 학습을 위한 MJSynth[24]를 사용한다. EMNIST 데이터는 ByClass, Balanced 2가지 케이스를 사용하였으며, MJ 데이터로부터 기존 파이프라인에서 명확히 인식되는 경우를 구분하여 문자 형태로 변환한 Train Good case Character(TrGc)으로 추출하였다.

기존 STR 모듈에서 인식이 제대로 되지 않는 데이터들의 경우 흔들림, 저해상도, 필기체, 투명도, 노이즈 등과 같은 다양한 사유로 텍스트의 특징들이 이미지에 부족한 것으로 보였다. 이에 대응하기 위한 방법으로 스켈레톤화(Skeletonize)[33] 기법을 적용하여 텍스트의 뼈대를 강조하여 손상되거나 변질이 심한 경우에 대한 성능 개선 정도도 동시에 확인하였다. 분류기에 사용될 TrGc, TeBc 데이터

Table 2. Dataset specification.

표 2. 데이터셋 정보

Dataset	Total images	Usage
TeBc	1483	Test Classifier
TrGc	821,022	Train Classifier
EMNIST_ByClass	131,600	Train OCR engine
EMNIST_Balanced	814,255	Train OCR engine
Skeletonize_TeBc	1483	Test Classifier
Skeletonize_TrGc	821,022	Train Classifier

셋에 대해 스켈레톤화 작업을 수행해 Skeletonize\_TrGc, Skeletonize\_TeBc 데이터 셋을 함께 확보하였다(표 2). 스켈레톤화 과정에서 Skimage에서 제공하는 Skeletonize 라이브러리를 사용하였으며, 총 3가지의 방법(Skeletonize, Thin, Lee94) 중 가장 변화가 확실한 Thin method를 적용하였다.

2. 제안 방법

인식 실패 문자 인식률을 향상시키기 위하여 추가되는 애드온은 OCR 엔진과 분류기의 두 가지의 핵심 구조로 구성된다. 필기체 데이터인 EMNIST 데이터 셋을 이용하여 다양한 변형 형태에 공통된 패턴의 학습을 의도하였으며, TrGc 데이터를 분류기 학습에 이용함으로써 이미지에 표현된 텍스트에 공통된 패턴의 학습을 의도하였다. 기존 STR 파이프라인이 인식하지 못한 문자 인식을 위하여 STR 파이프라인의 출력단에 제안된 애드온을 추가한다.<sup>1)</sup>

가. OCR 네트워크 학습

OCR 엔진은 VGG16에 기반하였으며, 마지막 단에서 vector를 출력하기 위해 dense 레이어를 제외하였다. EMNIST 데이터 셋의 EMNIST\_byclass 62 label(0-9, A-Z, a-z), EMNIST\_balanced 47 label(0-9, A-Z, a, b, d, e, f, g, h, n, q, r, t), 총 2가지 데이터 셋 별로 OCR 네트워크의 학습을 시도하여 성능을 비교하였다. 아래 표 3에는 학습에 사용된 모델의 인자 정보가 나열되어 있으며, 표 4에서 OCR엔진의 성능을 보고하였다.

62개 레이블의 데이터와 47개 레이블의 데이터에 대하여 비교한 결과 47개 레이블에 대하여 훈련된 OCR 엔진이 validation set에 대하여 94%, test set에 대하여 88%의 accuracy 성능으로 전체 레이블에 대해 고르게 좋은 성능을 보여 애드온의 OCR엔진으로 선정하였다.

나. 분류기 학습

VGG16 기반 OCR엔진에 입력된 정보는 문자 식별에 관점에서 유효한 특징을 포함하는 벡터로 변환된 후 분류기의 입력으로 사용된다. 분류기로는 Decision Tree를 사용했다.

1) 이 경우 STR 파이프라인의 출력을 애드온을 통과시킬 것인지 결정하는 모듈이 존재해야 하나, 본 논문에서는 STR 파이프라인이 인식하지 못한 문자의 인식이 가능한지 확인하고자 했기 때문에 애드온의 인식 성능만 확인하였다.

Table 3. OCR DNN parameter specification.

표 3. OCR 딥뉴럴네트워크의 인자정보

Layer	Parameters
Input	Transform image c=3, 224×224
Conv1_1	C=32, 64, 3×3, ReLU
Conv1_2	C=64, 64, 3×3, ReLU
Maxpool1	k : 2×2
Conv2_1	C=64, 128, 3×3, ReLU
Conv2_2	C=128, 128, 3×3, ReLU
Maxpool2	k : 2×2
Conv3_1	C=128, 256, 3×3, ReLU
Conv3_2	C=256, 256, 3×3, ReLU
Conv3_2	C=256, 256, 3×3, ReLU
Maxpool3	k : 2×2
Conv4_1	C=256, 512, 3×3, ReLU
Conv4_2	C=512, 512, 3×3, ReLU
Conv4_2	C=512, 512, 3×3, ReLU
Maxpool4	k : 2×2
Conv5_1	C=512, 512, 3×3, ReLU
Conv5_2	C=512, 512, 3×3, ReLU
Conv5_2	C=512, 512, 3×3, ReLU
Maxpool3	k : 2×2
Linear(Vector)	512

Table 4. Comparison of OCR engines.

표 4. OCR 엔진 성능 비교

Model	Dataset	Train	Valid	Test	Valid Acc.	Test Acc.
OCR 62	EMNIST_ByClass	697,931	69823	116,323	76.01	75.24
OCR 47	EMNIST_Balanced	112,799	5,687	18,846	94.02	87.93

Algorithm 1. Training of the AddOn module.

알고리즘 1. 애드온 모듈 학습

```

Input:
EMNIST Balanced,
TrGc - MJSynth single character images
1: Train OCR on EMNIST Balanced
2: For an image  $c_n$  from TrGc
3:   Get vector  $v_n$  OCR ( $c_n$ )
4:   Append  $v_n$  to vector set  $V$ .
    $V = \{v_1, v_2, v_3 \dots v_n\}$ 
5: Train Classifier on  $V$ .
6: Return: Trained OCR and classifier
    
```

알고리즘 1을 통해 확보한 OCR엔진과 분류기로 구성된 애드온을 기존 STR 파이프라인에 추가한 후 기존 STR 파이프라인이 인식하지 못했던 문자들에 대한 인식 성능을 확인하였다.

#### IV. 실험 결과 및 토의

기존 STR 파이프라인이 인식하지 못한 사례들을 살펴보면 그림 1과 같은 경우가 존재한다. 필기체를 인식하지 못하거나, 저해상도로 인해 글자를 인식하지 못한 경우, 수직글자로 인해 알파벳이 오독되거나 인식하지 못하는 경우, 스타일이 가미된 글자의 모호하거나 겹쳐져 개별 문자를 인식하지 못하는 경우 등이 존재했다.

다양한 이유로 기존 STR 파이프라인이 인식에 실패한 문자에 대하여 애드온을 적용한 결과, 최대 10.92%의 문자를 추가로 인식할 수 있음을 확인하였다. 이에 더하여 스킐레톤화를 적용한 경우 인식 성능이 향상됨을 확인하였다.

Table 5. Performance improvement by our Add-on.

표 5. 애드온으로 인한 성능 향상

Model	Dataset	Number of unrecognized characters	Ratio of characters recognize by our Add-on
OCR+DT	TeBc	1483	<b>8.70%</b>
OCR+SK_DT	SK_TeBc	1483	<b>10.92%</b>

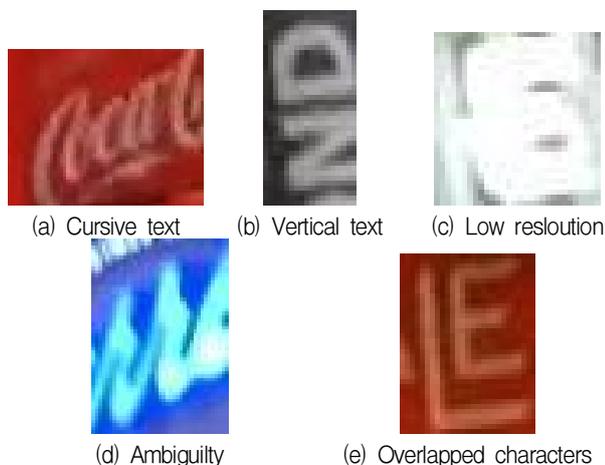


Fig. 1. Examples of unrecognized characters.  
 그림 1. 인식 실패 사례 (a) 필기체 (b) 수직글자  
 (c) 저해상도 (d) 모호함 (e) 겹침

#### V. 결론

본 논문에서는 이미지에 존재하는 텍스트의 성능 향상을 위하여 기존 STR 파이프라인에 OCR엔진과 분류기로 구성된 애드온을 추가하는 방법을 제안하였다. 새로 추가되는 애드온은 기존 STR 파이프라인이 성공적으로 인식한 이미지 텍스트와 다양한 변화를 갖는 필기체 데이터를 이용하여 학습되었기 때문에, 텍스트의 위치가 감지될 경우 기존 STR 파이프라인이 출력하는 벡터를 OCR 엔진과 분류기가 처리하여 기존 STR 파이프라인이 인식하지 못했던 텍스트의 일부(문자 단위로 최대 10.92%)를 인식하는데 성공하였다.

본 논문의 가장 큰 성과는 OCR 기반 분류기 애드온을 추가함으로써 기존 STR 파이프라인이 인식하지 못한 텍스트를 인식하는 것이 가능함을 보인 것이다. 그러나 기본적으로 기존 STR 파이프라인이 인식하지 못한 데이터는 테스트 데이터에 속하는 것이기 때문에, 기존 STR 파이프라인이 인식하지 못한 데이터에 대하여 OCR 기반 분류기 애드온을 적용하는 시점을 결정해야 한다.

추후 연구에서는 이와 같은 문제점을 해결하기 위하여 OCR 기반 분류기 애드온의 인식 결과와 기존 STR 파이프라인의 인식 결과를 결합하는 방법을 연구하고자 한다. 또한 입력 이미지의 변환, 특성 획득, 텍스트 위치 추정 과정에서 기존 STR 파이프라인의 성능을 확인한 후 텍스트 위치 추정, 변환, 특성 획득 모듈의 오류를 감안하여 텍스트를 인식하는 오류 강인 텍스트 인식 방법을 연구할 것이다.

#### References

[1] Y. Zhu, C. Yao, and X. Bai, "Scene Text Detection and Recognition: Recent Advances and Future Trends," *Front Compu .Sci*, vol.10, pp. 19-36, 2016. DOI: 10.1007%2Fs11704-015-4488-0  
 [2] Z. Raisi, M. A. Naiel, P. Fieguth, S. Wardell, and J. Zelek, "Text Detection and Recognition in the Wild: a Review," arXiv:2006.04305, 2020.  
 [3] X. Chen and A. L. Yuille, "Detecting and Reading Text in Natural Scenes," in *Proc. CVPR*

2004, 2004.

[4] J.-J. Lee, P.-H. Lee, S.-W. Lee, A. Yuille, and C. Koch, "AdaBoost for Text Detection in Natural Scene," in *Proc. ICDAR 2011*, pp.429-434, 2011. DOI: 10.1109/ICDAR.2011.93

[5] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What Is Wrong With Scene Text Recognition Model Comparisons? Dataset and Model Analysis," in *Proc. ICCV 2019*, pp.4715-4723, 2019.

[6] S. M. Hanif and L. Prevost, "Text Detection and Localization in Complex Scene Images using Constrained AdaBoost Algorithm," in *Proc. Int. Conf. on Doc. Anal. and Recognit.* pp.1-5, 2009. DOI: 10.1109/ICDAR.2009.172

[7] K. I. Kim, K. Jung, and J. H. Kim, "Texture-based approach for text detection in images using support vector machines and continuously adaptive mean shift algorithm," *IEEE Trans Pattern Ana. Mach Intell.*, vol.25, no.12, pp.1631-1639, 2003.

DOI: 10.1109/TPAMI.2003.1251157

[8] C. Yi and Y. Tian, "Text String Detection From Natural Scenes by Structure-Based Partition and Grouping," *IEEE Trans. Image Process*, vol. 20, no.9, pp.2594-2605, 2011.

DOI: 10.1109/TIP.2011.2126586

[9] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," In *Proc. of CVPR*, 2010.

DOI: 10.1109/CVPR.2010.5540041

[10] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. ACCV 2010*, pp.779-783, 2010.

DOI: 10.1007/978-3-642-19318-7\_60

[11] J. Matas, C. M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis Comput*, vol.22, pp.761-767, 2004. DOI: 10.1016/j.imavis.2004.02.006

[12] X.-C. Yin, X. Yin, K. Huang, and H.-W. Hao, "Robust Text Detection in Natural Scene Images," *IEEE Trans Pattern Ana. Mach Intell*, vol.36, no.5, pp.970-983, 2014.

DOI: 10.1109/TPAMI.2013.182

[13] A. Tabassum and S. A. Dhondse, "Text Detection Using MSER and Stroke Width Transform," in *Proc. CSNT 2015*, 2015.

DOI: 10.1109/CSNT.2015.154

[14] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep Structured Output Learning for Unconstrained Text Recognition," arXiv:1412.5903, 2015.

[15] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "WordSup: Exploiting Word Annotations for Character based Text Detection," in *Proc ICCV 2017*, pp.4940-4949, 2017.

[16] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," in *Proc. NIPS 2015*, pp.2017-2025, 2015.

[17] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in *Proc. ICLR*, 2015.

[18] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proc. CVPR 2016*, pp.770-778, 2016.

[19] C.-Y. Lee and S. Osindero, "Recursive Recurrent Nets with Attention Modeling for OCR in the Wild," in *Proc CVPR 2016*, pp.2231-2239, 2016.

[20] Y. Baek, B. Lee, D. Han, S. Yun, and H. Lee, "Character Region Awareness for Text Detection," in *Proc. CVPR 2019*, 2019.

[21] Y. Liu, S. Zhang, L. Jin, L. Xie, Y. Wu, and Z. Wang, "Omnidirectional Scene Text Detection with Sequential-free Box Discretization," in *Proc. IJCAI 2019*, 2019.

[22] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing Attention: Towards Accurate Text Recognition in Natural Images," in *Proc. ICCV 2017*, pp.5076-5084, 2017.

[23] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin, "GTC: Guided Training of CTC towards Efficient and Accurate Scene Text Recognition," in *Proc. AAAI-20*, pp.11005-11012, 2020.

[24] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. "Synthetic Data and Artificial Neural Networks for Natural Scene Text Recognition,"

in *Workshop on Deep Learning NIPS*, 2014.

[25] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic Data for Text Localisation in Natural Images," in *Proc. CVPR 2016*, 2016.

[26] A. Mishra, K. Alahari, and C. V. Jawahar, "Scene Text Recognition using Higher Order Language Priors," in *Proc. BMVC 2012*, 2012.

[27] K. Wang, B. Babenko, and S. Belongie, "End-to-end Scene Text Recognition," in *Proc. ICCV 2011*, pp.1457-1464, 2011.

DOI: 10.1109/ICCV.2011.6126402

[28] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "ICDAR 2003 Robust Reading Competitions," In *Proc. ICDAR 2003*, pp.682-687, 2003.

[29] D. Karatzas, et al. "ICDAR 2013 Robust Reading Competition," in *Proc. ICDAR 2013*, pp.1484-1493, 2013.

[30] D. Karatzas et al., "ICDAR 2015 Competition on Robust Reading," in *Proc. ICDAR 2015*, 2015.

[31] C. Yi and Y. Tian, "Text String Detection from Natural Scenes by Structure-based Partition and Grouping," *IEEE Trans Image Process*, vol. 20. no.9, pp.2594-2605. 2011.

DOI: 10.1109/TIP.2011.2126586

[32] Cohen, G., Afshar, S., Tapson, J., & van Schaik, A. "EMNIST: an extension of MNIST to handwritten letters.," arXiv:1702.05373, 2017.

[33] T. Y. Zhang, and C. Y. Suen, "A Fast Parallel Algorithm for Thining Digital patterns," *Commun ACM*, vol.27, no.3, pp.236-239, 1984.

DOI: 10.1145/357994.358023

## BIOGRAPHY

### Ho-Yeol Chae (Member)



2019 : BS degree in Computer and Communications Engineering, Kangwon National University  
2019~present : Master course in Computer and Communications Engineering, Kangwon National University

### Ho-sik Seok (Member)



1999 : BS degree in Computer Engineering, Seoul National University.

2001 : MS degree in Electrical Engineering and Computer Science, Seoul National University.

2012 : PhD degree in Electrical Engineering and Computer Science, Seoul National University.

2001~2004 : Instructor, Korea Military Academy.

2016~2020.2 : Assistant professor, Dept. of Computer and Communications Engineering, Kangwon National University

2020.3~present : Assistant professor, Dept. of Computer Science and Engineering, Kangwon National University