KIPS Trans. Softw. and Data Eng.
Vol.11, No.3 pp.115~124
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2022.11.3.115

YOLO, EAST: Comparison of Scene Text Detection Performance, Using a Neural Network Model

Park Chan Yong† · Lim Young Min†† · Jeong Seung Dae††† · Cho Young Heuk†††† · Lee Byeong Chul††††† · Lee Gyu Hyun††††††† · Kim Jin Wook†††††††

ABSTRACT

In this paper, YOLO and EAST models are tested to analyze their performance in text area detecting for real-world and normal text images. The earl ier YOLO models which include YOLOv3 have been known to underperform in detecting text areas for given images, but the recently released YOLOv4 and YOLOv5 achieved promising performances to detect text area included in various images. Experimental results show that both of YOLO v4 and v5 models are expected to be widely used for text detection in the filed of scene text recognition in the future.

Keywords : Scene Text Detection, YOLO, EAST, Neural Network

YOLO, EAST: 신경망 모델을 이용한 문자열 위치 검출 성능 비교

박 찬 용† · 임 영 민†† · 정 승 대††† · 조 영 혁†††† · 이 병 철††††† · 이 규 현††††††† · 김 진 욱†††††††

요 약

본 논문에서는 최근 다양한 분야에서 많이 활용되고 있는 YOLO와 EAST 신경망을 이미지 속 문자열 탐지문제에 적용해보고 이들의 성능을 비교분석 해 보았다. YOLO 신경망은 일반적으로 이미지 속 문자영역 탐지에 낮은 성능을 보인다고 알려졌으나, 실험결과 YOLOv3는 문자열 탐지에 비교적 약점을 보이지만 최근 출시된 YOLOv4와 YOLOv5의 경우 다양한 형태의 이미지 속에 있는 한글과 영문 문자열 탐지에 뛰어난 성능을 보여줌을 확인하였다. 따라서, 이들 YOLO 신경망 기반 문자열 탐지방법이 향후 문자 인식 분야에서 많이 활용될 것으로 전망한다.

키워드 : 문자열 탐지, YOLO, EAST, 신경망

1. 서 론

복잡하고 다양한 배경속에 포함된 문자열을 탐지하는 기술은 자율주행 자동차나 로봇에 적용하는 시각 기술의 일환으로 발전되어 왔다. Fig. 1에서 보듯이 다양한 배경 속에 포함된 문자탐지(Scene Text Detection)는 실시간 번역, 이미지 검출, 영상 파싱, 위치정보추출 그리고 시각장애인을 위한 네비게이션 등 수많은 응용분야 때문에 컴퓨터비전 연구에서 크게 주목을 받고 있다[1].

하지만 다양한 배경 속에 포함된 문자 인식은 인식할 문자열이 해당 배경에서 어디에 위치해 있는지 탐지가 어렵고, 카메라의 성능과 초점, 떨림, 조명 등 다양한 원인에 따라 이미지의 품질에 많은 영향을 끼치며, 텍스트의 배치가 평행하지 않고 심지어 회전될 수도 있으며 다양한 글꼴들로 이루어질 수 있기에 기존 OCR과는 다른 차원의 난이도를 가진다[2].

보편적으로 이미지에서 문자인식을 하는 과정은 문자열 탐지와 문자인식의 2단계로 구분이 되는데, 문자인식뿐만 아니라 선행 단계인 문자열 탐지 또한 어려운 난이도를 가지기에 다양한 연구들이 있어왔다. 전통적으로 문자열 탐지방법들은 문자나 단어 후보를 생성하거나 필터링 및 그룹화 하는 것처럼 다수의 단계를 구성하고 처리를 하게 된다. 하지만, 이러한 접근 방법들은 각각의 단계별로 수많은 매개변수를 최적

※ 이 논문은 2019~2021년도 중소벤처기업부의 창업성장 기술개발사업 지원에 의해 이루어짐[S2833775].
※ 이 논문은 2021년 한국정보처리학회 춘계학술발표대회에서 '이미지 속 문자열 탐지에 대한 YOLO와 EAST 신경망의 성능 비교"의 제목으로 발표된 논문을 확장한 것임.
† 준 회 원 : (주)투아트 과장
†† 비 회 원 : (주)투아트 과장
††† 종신회원 : (주)투아트 개발이사
†††† 비 회 원 : (주)투아트 부사장
††††† 비 회 원 : (재)경상북도경제진흥원 일자리산업실 실장
†††††† 비 회 원 : 경북대학교 컴퓨터공학부 학사과정
††††††† 비 회 원 : 경북대학교 컴퓨터공학부 초빙교수
Manuscript Received : June 29, 2021
First Revision : August 23, 2021
Accepted : August 31, 2021
* Corresponding Author : Kim Jin Wook(deepkaki@knu.ac.kr)

※ 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.

Fig. 1. Examples of Scene Text Detection [3]

화하고 휴리스틱 이론을 적용하였으나 결국 전반적으로 성능 평가가 좋지 못하다[4].

최근에는 SSD(Single Shot multibox Detector), Faster R-CNN 그리고 FCN(Fully Convolutional Network)과 같은 객체를 탐지하고 분할하는 방법을 적용한 딥러닝 기반 문자열 탐지방법들이 제시되고 있다. 이러한 방법은 다양한 배경 속에서 문자 단위가 아닌 단어를 기준으로 경계박스를 찾기 위해 신경망 모델을 학습하여 주목할 만한 결과들을 보여주고 있다.

M. Liao[5]는 TextBox라는 문자탐지를 수행하는 모델을 제안하였는데 이는 일반적인 물체를 감지하는 SSD에 문자열 탐지에 적합한 가로, 세로 비율을 가지는 필터를 설계하고 단일 네트워크(single network)로 구성되어 빠르게 문자열을 탐지하며 정확도도 높은 방법이다. 이 후, Liao[6]는 임의 방향 문자열 탐지를 위해 이전에 제안한 TextBox를 개선한 TextBoxes++를 제안하였으며 저해상도 이미지에 대한 학습으로 인해 SVT 데이터 세트에서도 좋은 성능을 보여준다.

F. Jiang[7]는 ParseNet을 이용한 문자열 분할(text segmentation)과 Fast-RCNN 구조를 이용한 문자 탐지(character detection)를 각각 수행하고 그 결과를 통합하여 문자열을 탐지하는 방법을 제안하고, Y. Jiang[8]은 임의 방향의 텍스트 탐지를 위해 Rotational Region CNN (R2CNN)이라고 불리는 Faster-RCNN 기반 아키텍처를 제안하여 가로로 배열된 문자열 외에 다양한 방향으로 배열된 문자열 추출도 가능함을 보인다.

이 외에도, P. Lyu[3]도 문자열 영역을 나타내는 꼭지점들을 추출하고 모은 뒤, 텍스트 분할 맵(text segmentation map) 정보를 FCN을 통해 결합하여 문자열을 찾아내는 방법을 제시하고 있으며, X. Zhou[9]의 EAST도 문자영역을 표시하는 RBOX방식을 FCN에 접목하여 다양한 방향성을 가진 문자열 탐지에 대해 좋은 성능을 보여준다. H. Hu[10]은 문자(character) 단위로 학습된 FCN을 사용하여 문자열 영역을 추출하는 방법을 제안하는데, 이를 위해 단어단위(word level) 텍스트 라벨링이 된 학습 데이터에서 문자단위(character level)로 영역을 분할하는 방법을 제시하고 다양한 방향성의 문자열 탐지에도 좋은 성능을 보인다. S. Long[11]은 보다 다양한 형태의 문자열 추출을 위한 시도로 FCN을 사용하여 TextSnake라고 하는 문자열 표현 방식을 제안하고 임의의 방향 문자열 뿐만 아니라 곡선 형태로 배열된 문자열도 추출하는 특성을 보여준다.

일반적으로 문자열 탐지와 문자열 인식은 상호 보완적이고 밀접하게 연결되어 있기에 이 두 작업을 하나로 합쳐서 진행하는게 최근의 추세[12,13]이며 NLP에서 주목할 만한 성과를 보이는 어텐션 메커니즘(Attention Mechanism)을 도입하는 사례[12,14]도 있다. P. Lyu[13]는 MaskTextSpotter라고 하는 신경망 구조를 통해 문자열 탐지와 문자열 인식을 동시에 수행하는 모델을 제시하고 있으며, T. He[12]는 문자열 탐지와 문자열 인식을 한꺼번에 수행하는 모델을 제시하고 문자인식에 어텐션 메커니즘을 사용하고, P. He[14]는 inception module을 사용하여 multi-scale featured와 어텐션 맵(attention map)을 사용하여 문자열 영역을 찾아는 방법을 제안한다.

본 논문에서는 현재 전 세계에 서비스 중인 시각장애인을 위한 시각보조 알림 서비스 설리번플러스의 성능을 향상시킬 목적으로 YOLO (You Only Look Once) 신경망과 EAST (An Efficient and Accurate Scene Text Detector) 신경망을 활용하여 복잡한 배경이나 문서와 같이 일반적인 문자열을 탐지하고 이들 모델의 문자열 탐지 성능에 대한 데이터를 제시한다.

최신 YOLO 신경망을 문자 탐지에 특화하여 적용한 사례가 드물며 특히 한글 문자 탐지에 대해서 집중적으로 테스트한 사례는 전무하기에 본 논문의 실험 결과는 향후 유사한 문제에 YOLO와 EAST를 적용하고자 할 때 참고할 수 있을 것이다.

다음 장에서는 시각장애인의 시각보조 앱에 대해 설명하고 이어서 YOLO와 EAST 신경망의 주요 특징 그리고 이들 각각의 신경망 모델에 실제 데이터를 적용한 실험 결과에 대해 설명하고 결론을 도출한다.

2. 시각장애인을 위한 시각보조 음성안내 앱

설리번플러스는 Fig. 2에서 보여주듯이 시각이 불편한 사람들 위해 스마트폰에 탑재된 카메라를 통하여 촬영된 이미지를 인식한 후 음성으로 인식된 정보를 알려주는 시각장애인용 시각보조 알림 서비스(앱)이다. 2021년 6월 1일 기준 전세계 192개국에서 월 15만명 이상의 사용자들이 200만건 이상의 사진을 업로딩하면서 AI를 통해 시각 보조 서비스를 이용하고 있다. 설리번플러스의 주요기능으로는 이미지캡셔닝(Image Captioning) 기술을 이용한 이미지 인식, 이미지 속에 포함된 문자를 탐지하여 알려주는 문자인식(OCR) 기술 그리고 사람의 얼굴을 탐지하여 나이,성별 등 얼굴 정보를 알려주는 얼굴인식(Face Recognition) 기술이 있다. 실제 사용자들이 가장 자주 사용하는 서비스(기능)를 보면, Fig. 3과 같이 문자인식 기능을 사용하는 빈도수가 가장 높은 것을 볼 수 있는데 그만큼 사람들이 문자를 통해 가장 많은 정보를 취득하기 때문이다.

시각장애인들의 경우 스마트폰에 탑재된 카메라를 활용하여 대상을 인식하고자 할 때 카메라 화면 안에 인식하고자 하

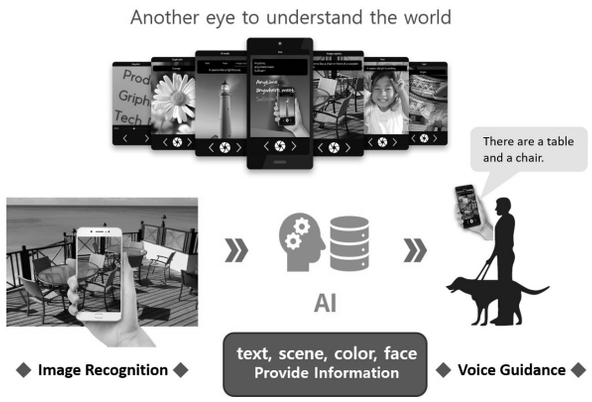


Fig. 2. Introduction of SullivanPlus

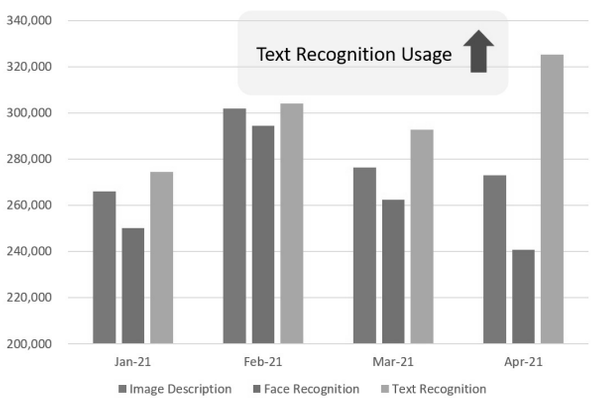


Fig. 3. SullivanPlus's Function Usage Frequency Comparison

는 대상을 정확하게 포함시키는 것이 가장 어려운 문제가 된다. 시력이 정상인 사람들은 쉽게 카메라 화면에 인식하고자 하는 대상을 포함시킬 수 있지만, 앞을 볼 수 없거나 흐릿하게 보게 되는 시각 장애인들의 경우는 카메라 화면에 대상을 포함시키는 것부터 어렵다. 이러한 문제로 인해 대부분의 시각 장애인을 위한 시각 보조 서비스들의 경우, 카메라 화면 안에 대상을 포함시킬 수 있도록 소리와 진동 알림으로 사용자에게 알려주는 기능을 탑재하고 있다. 이 때문에 설리번 플러스의 문자 인식 기능의 경우도, 이미지 속 문자를 인식하는 단계뿐만 아니라 문자열을 탐지하는 단계 모두 중요한 성능 지표로 두고 성능을 향상시키기 위해 노력해오고 있다.

특히 시각 장애인들에게 있어 스마트폰 카메라를 이용하여 문자를 탐지해야 하는 상황은 빠른 피드백을 제공할 필요가 있어 스마트폰 엔진의 성능이 중요할 뿐만 아니라 문자열 탐지 기능의 경우는 특히 모바일 단말기에 탑재되어 처리할 수 있도록 가벼운 모델이어야 한다.

설리번플러스 출시 후 서비스 초반에는 클라우드 기반 AI 기능을 이용하지 않고 스마트폰 디바이스 내에서 동작하는 구글의 문자탐지 API를 이용하여 문자 탐지하는 기능을 추가하였으나, 해당 기능은 라틴 계열 문자만을 탐지하고 한글은 탐지하지 못하는 단점으로 인해 아쉬움이 있었다. 이에 가볍고 성능이 뛰어난 문자탐지 엔진 개발을 위해 신경망 기반의 다양한 접근법들을 고려하던 중, YOLO 신경망이 상당히 빠른 처리 속도와 비교적 양호한 객체 탐지 성능을 보여서 이에 대한 다양한 실험을 진행하게 되었다.

3. EAST와 YOLO 신경망을 이용한 문자열 탐지

3.1 EAST와 YOLO 신경망의 특징

일반적으로 CNN을 이용하여 문자열 탐지를 하는 경우라도, 파이프라인(pipeline) 안에 다수의 단계(stage)와 상호작용하는 요소가 많아지면 성능에 크게 악영향을 미친다. 예를 들어 Semantic Segmentation을 위해서 이미지의 각 픽셀이 어떤 클래스에 속하는지 분류를 해야 하는데 이 때 모든 픽셀을 분류 모델에 투입하여 구분하는 방식은 매우 비효율적이다. 이를 보완하기 위해 FCN(Fully Convolutional Network)은 기존 CNN에서 마지막 Fully Connected layer를 거치기 전 위치 정보가 보존된 feature map을 활용해 Segmentation map으로 복원하는 모델이다. EAST는 이 FCN (Fully Convolutional Network) 모델을 응용해 빠르고 정확한 2단계의 간단한 문자열 탐지 파이프라인으로 구성하여 단어가 포함된 기울어진 문자 영역을 예측할 수 있는 모델이다. 기존의 문자 탐지 모델들은 3~5차례 컨볼루션 블록을 거치게 한 것과 달리 하나의 컨볼루션 블록으로 줄여 연산 시간을 대폭 단축하였는데도 불구하고 Fig. 4에서 보여주듯이 빠른 속도와 높은 정확도의 성능을 보여주어 문자탐지 분야에서 주목을 받고 있다[9].

YOLO는 이미지를 S × S개의 그리드로 분할하고 각각의 각각의 그리드에 해당하는 객체의 신뢰도를 계산한 뒤 그리드를 합치는 동시에 경계상자의 위치를 조정하면서 객체 탐지 정확도를 높이는데 탐지성능이 뛰어나면서 빠른 처리속도를 자랑한다[15]. YOLO 이전의 R-CNN은 이미지를 여러장으로 분할하고, CNN모델을 이용해 이미지를 분석했다. 그렇기 때문에 이미지 한 장에서 객체탐지(Object Detection)를 해도 실제로는 여러 장의 이미지를 분석하는 것과 같았다. 이에 반해 YOLO는 이러한 과정 없이 이미지 전체를 한번만 보

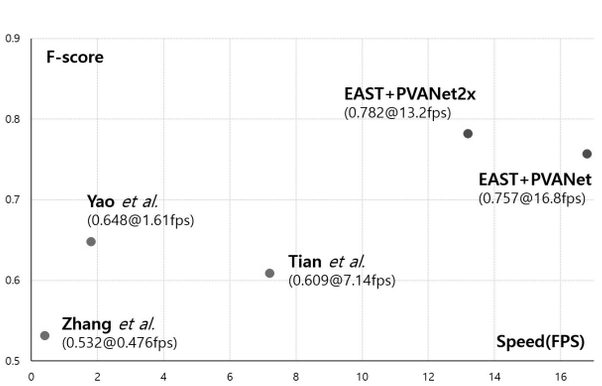


Fig. 4. Performance Versus Speed on ICDAR 2015 Text Localization Challenge [9]

는 특성을 가지고 있으며 이런 특성으로 인해 실시간으로 객체를 탐지 할 수 있는 빠른 처리 속도를 자랑한다.

YOLO가 유명해진 이유는 높은 성능은 아니더라도 준수한 성능으로 실시간으로 객체탐지가 가능하기 때문인데 Fig. 5의 벤치마크 테스트를 보면 기존의 Faster R-CNN보다 6배 빠른 성능을 보여준다[16].

YOLO의 경우 v3 이후 각기 다른 개발자들에 의해 v4와 v5가 개발되어 공개됐는데, Fig. 6의 벤치마크에서 보여주듯 YOLOv4는 EfficientDet와 비교하여 비슷한 인식 성능으로 속도는 2배정도 빠르며 YOLOv3와 비교하여 AP와 FPS는 각각 10%와 12%가 향상된 성능을 보여주고 있다[17].

이에 반해 YOLOv5는 정식 논문은 나오지 않고 깃허브(https://github.com/ultralytics/yolov5)[18]를 통해 오픈소스와 관련 데이터들이 테크 리포터 형식으로 공개되어 있기에 본 논문에서는 깃허브에 공개된 자료들을 사용했다. Table 1에서 보여주듯 YOLOv5s, YOLOv5m, YOLOv5l, YOLOv5x의 4가지 모델로 구성되어 있고 이들은 신경망의 크기가 각각 small, medium, large, xlarge이며 백본(backbone)이나 헤드(head)는 모두 동일하지만, depth_multiple(model depth multiple)과 width_multiple (layer channel multiple)이 다르며 large 모델이 1.0으로 기준이 되는 크기이다. 각 모델별

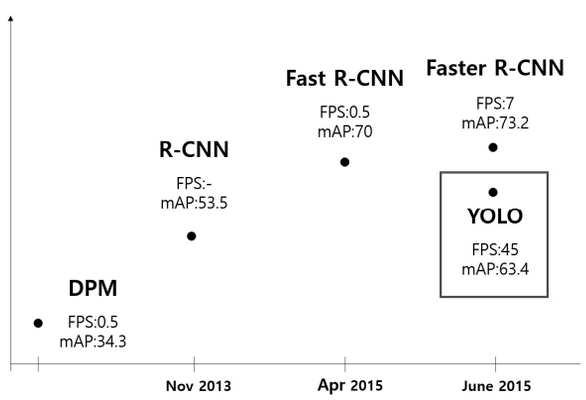


Fig. 5. Performance Comparison of YOLO and R-CNN [15]

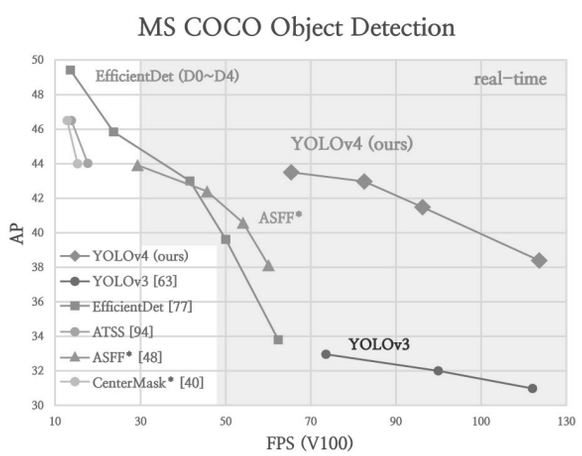


Fig. 6. Comparison of YOLOv4 and Other Object Detectors [17]

Table 1. Pretrained Checkpoints of YOLOv5 Models [18]

Model	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
size (pixels)	640	640	640	640
mAP^{val} 0.5:0.95	36.7	44.5	48.2	50.4
mAP^{test} 0.5:0.95	36.7	44.5	48.2	50.4
mAP^{val} 0.5	55.4	63.1	66.9	68.8
Speed $v100_{(ms)}$	2.0ms	2.7ms	3.8ms	6.1ms
params (M)	7.3M	21.4M	47.0M	87.7M
FLOPS 640(B)	17.0	51.3	115.4	218.8

성능을 Fig. 7에서 보여주고 있는데, 이미지 당 처리속도 대비 성능(AP)은 YOLOv5l 모델이 가장 좋은 것을 알 수 있다.

본 논문에서는 최신 YOLOv4와 YOLOv5 모델들이 문자열 탐지에도 충분히 효과적으로 사용될 수 있을 것으로 기대하면서 이미지 속 문자열 탐지에 적용하고 그 결과를 EAST 신경망 모델과 비교하여 제시한다. 이미 EAST와 YOLO 이들 신경망은 처리 속도 면에서는 충분히 빠르다는 것이 각종 논문에서 제시[9,17,19,20]되고 있기에 본 실험에서는 이들 신경망의 문자열 탐지 성능 면에 중점을 두고 실험을 진행한다. 특히 한글에 대해 이들 다섯 가지 모델(EAST, YOLOv4 Tiny, YOLOv4, YOLOv5s, YOLOv5x)의 성능비교에 대한 자료가 전무한 상황이기에 본 연구는 의의를 가진다.

3.2 신경망의 바운딩 박스 구조

입력된 이미지에 포함된 문자열에 신경망으로 객체 위치를 식별(Boxing)하는 방법으로 AABB(Axis-Aligned Bounding Box) 구조와 RBOX(Rotated Bounding Box) 구조가 있는데, Fig. 8에 보듯이 AABB 구조는 임의의 사각형 내의 각 포인트에서 네변까지의 거리를 나타내는 것으로 정렬된 사각형 구조이고 RBOX 구조는 이러한 AABB 구조의 정보에 추가하여 기울어진 각도 수치도 포함하여 회전된 문자 영역을 좀 더 정확하게 탐지할 수 있는 방법이다. 본 논문에서는 두 구조 모두 활용하여 연구를 진행하였는데 EAST 모델은 RBOX 구조, YOLO 모델은 AABB 구조로 학습 데이터를 구성하여 진행한다. 이 두 자료구조를 보면 EAST와 YOLO 두 신경망의 특징이 잘 드러난다. YOLO의 AABB 구조는 비교적 간단한

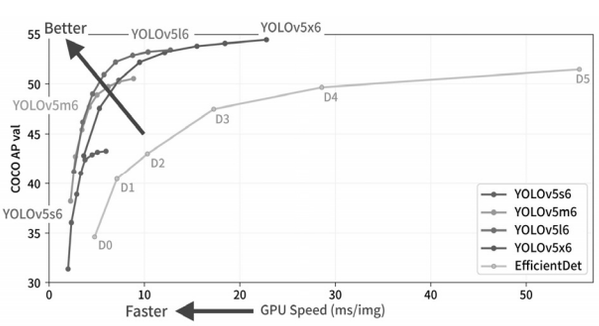


Fig. 7. Comparison of YOLOv5 and Other Object Detectors [18]

Fig. 8. Boxing Examples of RBOX and AABB

구조로 신경망의 처리 속도를 높이기 위한 측면이 있으며, EAST의 RBOX구조는 문자영역을 좀 더 정밀하게 추출하는 장점이 있다. EAST는 본래 문자영역 탐지를 목적으로 고안된 알고리즘이기에 이런 구조를 가지게 된 것으로 보인다. AABB구조와 RBOX구조 이 둘은 분명 속도와 성능 이 두가지 측면에서 각각의 장점이 존재하는 것이다.

4. 실험 및 결과

4.1 실험 환경 및 데이터셋

본 실험에 활용한 신경망 모델별 입력 이미지 사이즈는 EAST 512×512, YOLOv4 608×608 그리고 YOLOv5 640×640 픽셀이며, RGB 32비트이다. 또한 PC 환경은 운영체제는 Ubuntu 18.04 LTS, GPU는 GeForce RTX 2080 TI 11GB 환경에서 진행한다. 설리번플러스 서비스는 스마트폰에 탑재된 카메라를 통해 촬영된 영상속의 문자열을 탐지하는 기술이기에, 모델 학습 데이터를 구성 시 이미지의 다양한 변형 및 왜곡에 대한 특성을 감안하여 학습 데이터를 구성하였으며 특히 데이터의 다양성을 위해 한국지능정보사회진흥원의 AI Hub(aibub.or.kr)에서 제공하는 데이터 세트를 활용하여 학습데이터(Training Data)와 평가데이터(Validation Data)를 구성하고 문자영역 탐지 실험을 진행하였다.

본 논문에서는 두 가지 데이터 세트를 구성하여 모델을 구성하였는데 첫 번째 모델은 일반적인 문서에 있는 문자영역을 감지하는 형태이고 두 번째는 실생활에서 볼 수 있는 책, 간판, 상표 등과 같이 다양한 배경과 조합된 문자를 감지하는 형태이다. 문자 데이터 세트는 한국어-영어 번역 말뭉치의 문어체 뉴스 데이터 세트를 활용하여 한글의 다양한 폰트별 이미지 생성기를 개발한 후 학습데이터와 평가데이터를 생성하였다. 생활 속 이미지의 문자 데이터 세트는 한국어 문자 이미지 중 Text in the Wild로 제공하는 전체 10만장(표지판·이정표 1.7만장, 상표 3.7만장, 간판 3.0만장, 기타 1.6만장) 이미지 중 일부를 추출하여 학습 데이터와 평가 데이터로 활용하여 각각의 모델 성능을 비교하였다.

Fig. 9. Examples of Training Data Set

텍스트 문서의 경우 학습데이터를 20,000개(한글과 영문 각각 10,000개) 그리고 평가데이터 2,000개(한글과 영문 각각 1,000개)를 구성하였으며. 생활 속 이미지의 경우 학습데이터 100,216개(한글과 영문 포함) 그리고 평가데이터 10,000개(한글과 영문 포함)로 구성하였다.

4.2 실험결과 성능 평가 기준

신경망이 출력하는 각 바운딩 박스의 신뢰도(confidence) 값에 대한 임계치(threshold) 이상 되는 영역에 대해 IOU를 적용하여 성능을 평가하는데, Fig. 10처럼 전체 정답 박스 영역과 인식한 박스 영역에서 서로 겹치는 영역의 비율인 IOU로 나타낸다. 본 실험에서는 IOU 수치가 50% 이상이면 True, 미만이면 False로 판단한다. Fig. 10의 (A)에서 파란 점선은 이미지에서 실제 문자열 영역(ground truth)을 지정한 부분이고, 붉은 실선은 신경망이 문자열 영역으로 판단(predict)한 영역을 나타내며 (B)에는 실제 IOU를 계산하는 예를 보이고 있다.

각각의 신경망 모델에 대한 분류 성능 평가지표로 대표적인 지표인 Precision, Recall, F1-Score을 사용하는데, 모델이 True로 예측한 것 중 실제 True인 비율을 나타내는 Precision, 실제 True인 것 중 모델이 True로 예측한 비율을 나타내는 Recall 그리고 마지막으로 Precision과 Recall의 조화 평균을 나타내는 것이 F1-Score이다.

Fig. 11에서 보면 (a)는 데이터에서 정확하게 추출해야 하

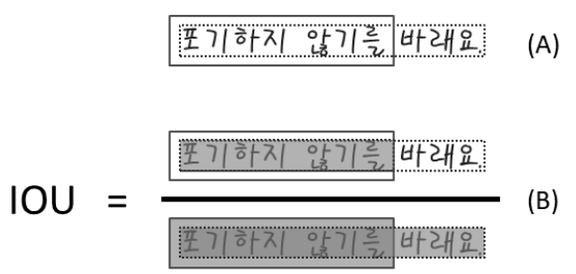


Fig. 10. Intersection Over Union (IOU). Red is **Predicted** Bounding Box and Blue is **Ground Truth** Bounding Box

(A) Five Blue Dotted Boxes are Ground Truth Bounding Boxes

(B) Three Red Boxes are Detected Correctly by Neural Network

(C) Two Red Boxes are Detected Correctly and
One Box is Detected Incorrect by Neural Network

(D) Two Red Boxes are Detected Correctly and
One Box is Detected Incorrect by Neural Network

Fig. 11. Precision and Recall Explanation

는 5개의 영역을 보여준다. (b)에서는 3개의 영역을 신경망이 문자영역으로 정확하게 탐지한 예를 보여주고 있으며, (c)와 (d)는 2개의 영역에 대해서 정확하게 탐지하고 다른 한 영역은 잘못 탐지한 예를 보여준다.

Table 2에는 성능평가지표의 정의를 나타내고 있으며 Fig. 11에 대해 성능평가지표를 적용해 보면 Fig. 11(b)의 경우 Precision은 2/3, Recall은 2/5가 되고 (c)와 (d)의 경우는 Precision은 3/3, recall은 2/5가 된다.

Table 2. Evaluation Criteria Description

Terms Deifinitions
• **True Positives [TP]** - Text area correctly identified as text area
• **True Negatives [TN]** - Non text area correctly identified as non text area
• **False Positives [FP]** - Non text area incorrectly identified as text area
• **False Negatives [FN]** - Text area incorrectly identified as non text area
Criteria Definitions
• **Precision** - true positives per predicted positive - *Precision = TP / (TP+ FP)*
• **Recall** - true positives per real positive - *Recall = TP / (TP + FN)*
• **F1 Score** - The harmonic mean of the precision and the recall - *F1 Score = TP / (TP + 0.5(FP +FN))*

4.3 실험결과

각각의 모델에 대한 트레이닝 횟수별 성능지표를 Table 3~5에서 보여주고 있으며, 세 모델 모두 Epoch 약 5회에서 최상의 성능을 보여주고 있어 이때의 신경망을 이용하여 모델별 -성능을 비교하였다.

단순한 배경에 문자로만 구성된 일반 문서 형태를 대상으로 한글과 영문 둘 다 학습한 상태의 모델에서 영문과 한글 문자 영역 탐지 성능을 Table 6과 Table 7에서 보여주고 있다. 그리고, 생활 이미지의 문자열 추출에 대한 실험 결과는 Table 8에 정리하였다.

실험결과 한글과 영문 모두 학습한 상태에서 일반 텍스트 문서의 문자열 영역 탐지는 YOLOv4 Tiny를 제외한 EAST와 YOLOv4, YOLOv5s, YOLOv5x 네 모델이 모두 우수한 성능을 나타낸다.

Table 3. EAST's Performance by Number of Training

Epoch	Precision	Recall	F1-score
1	0.3608	0.5065	0.38
2	0.6643	0.9274	0.7741
3	0.9976	0.9991	0.9984
4	0.99	1	0.99
5	0.9994	1	1

Table 4. YOLOv4's Performance by Number of Training

Epoch	Precision	Recall	F1-score
1	0.83	0.99	0.90
2	0.87	1	0.93
3	1	1	1
4	0.99	1	0.99
5	0.99	1	1

Table 5. YOLOv5x's Performance by Number of Training

Epoch	Precision	Recall	F1-score
1	0.8511	0.9996	0.9193
2	0.892	0.9997	0.9997
3	0.931	0.9997	0.9997
4	0.9525	0.9998	0.9998
5	0.986	1	0.9929

Table 6. English Text Detection Ratio After Training Eng. + Kor.

Model	Precision	Recall	F1-score
EAST	1	0.9978	0.9989
YOLOv4 Tiny	0.60	0.80	0.68
YOLOv5s	0.992	1	0.9978
YOLOv4	0.99	1	1
YOLOv5x	0.916	1	0.9561

Table 7. Korean Text Detection Ratio After Training Eng. + Kor.

Model	Precision	Recall	F1-score
EAST	0.9994	1	0.9997
YOLOv4 Tiny	0.61	0.80	0.69
YOLOv5s	0.997	1	0.9984
YOLOv4	1	1	1
YOLOv5x	0.986	1	0.9929

Table 8. Ratio of Scene Text Detection After Training Eng. + Kor.

Model	Precision	Recall	F1-score
EAST	0.42	0.35	0.38
YOLOv4 Tiny	0.61	0.36	0.46
YOLOv5s	0.613	0.686	0.620
YOLOv4	0.64	0.68	0.66
YOLOv5x	0.613	0.588	0.6002

Table 9. Ratio of Scene Text Detection with Refined Data

Model	Precision	Recall	F1-score
EAST	0.60	0.53	0.57
YOLOv4 Tiny	0.65	0.40	0.49
YOLOv5s	0.763	0.836	0.782
YOLOv4	0.80	0.84	0.82
YOLOv5x	0.763	0.734	0.770

Table 8은 생활속 이미지를 대상으로 한 문자열 탐지 실험 결과로 결과는 상당히 낮게 나왔는데, 그 이유는 AI Hub의 학습데이터의 정확도가 상당히 낮아서 발생한 문제였다. AI Hub의 학습데이터의 한 이미지에는 다수의 문자열 영역들이 라벨링 데이터로 만들어져 있는데, 전체 라벨링 데이터 중 학습 결과에 영향을 줄 수 있는 심하게 정확하지 않은 라벨링 데이터들이 약 30% 이상이나 존재하여 낮은 인식 결과를 보인 것으로 판단하였다.

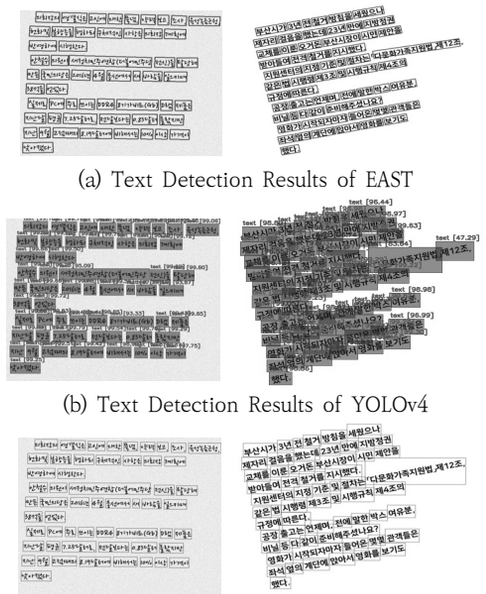
따라서, 본 실험에서는 명확하게 잘못 라벨링 된 데이터들을 제거한 뒤 신경망을 다시 학습하여 그 결과를 비교해 보았다. Table 9에서 개선된 학습 데이터를 이용하여 라벨링한 결과를 보인다.

또한 Figs. 12, 13에서는 EAST와 YOLOv4, YOLOv5 세 모델을 통해 문서와 생활 속 이미지를 대상으로 실제 문자열을 탐지한 결과를 보여준다. YOLOv4는 다수의 바운딩 박스가 도출되어 이들 바운딩 박스들이 서로 중첩하는 영역에 대해 푸른색으로 표시하였다. 그림에서 보면 이들 중첩영역 부분에 대해서만 문자인식 단계에서 고려해도 되지만 정확도를 더 향상시키기 위해서는 중첩영역들을 합집합(union)한 영역에 대해서 문자인식을 적용해 볼 수도 있다. 결과를 보면 YOLOv4와 YOLOv5의 경우, 추출된 바운딩 박스가 문자들을 충분히 포함하고 문자 인식에 사용할 수 있을 수준으로 생성됨을 알 수 있다.

4.4 결과분석

실험결과를 보면 전체적으로 클라우드 환경의 신경망 모델이 더 많은 레이어와 큰 사이즈의 이미지를 입력이미지로 활용하여 학습하기 때문에 상대적으로 무거우나 성능 면에서는 우수한 것을 알 수 있다. 하지만 모바일 환경의 신경망 모델 또한 성능지표로는 조금 떨어지지만 문서가 아닌 간판, 책표지 등과 같이 실생활 환경의 문자를 빠르게 인식할 필요가 있는 환경에서는 충분히 활용할 가치가 있는 것으로 확인된다.

다만, AI Hub 학습데이터의 정밀도가 아직 많이 낮다는 점은 크게 아쉬운 부분이다. AI Hub가 만들어지고 지금까지 양적 성장 위주로 데이터를 구축해온 결과이기에 나타난 현상이라 판단된다.

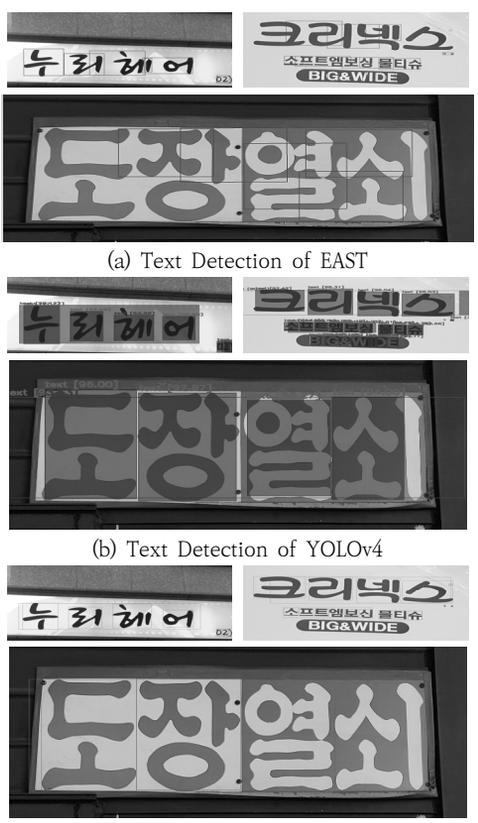


(a) Text Detection Results of EAST

(b) Text Detection Results of YOLOv4

(c) Text Detection Results of YOLOv5

Fig. 12. Results of Text Detection in Documents



(a) Text Detection of EAST

(b) Text Detection of YOLOv4

(c) Text Detection of YOLOv5

Fig. 13. Results of Scene Text Detection

실제 AI Hub에서 제공하는 라벨링 데이터를 Fig. 14와 15에서 나타내었다. Fig. 14는 정확하게 라벨링 된 예를 보여주는데, 각 글자 단위로 바운딩 박스가 만들어져있고 단어 단위로도 바운딩 박스가 정확하게 만들어져 있다. 하지만, Fig. 15(a)를 보면

오렌지색 화살표가 바운딩 박스가 너무 넓게 만들어진 데이터 영역들을 가리키고 있으며, 15(b)는 바운딩 박스를 만들어야 할 문자영역에 대해서 만들어지지 않은 부분과 너무 넓게 만들어진 바운딩 박스 영역을 보여주고 있다. 이렇게 실제 문자 영역보다 너무 크게 바운딩 박스가 만들어진 라벨링 데이터들은 학습결과에 부정적인 영향을 끼치게 된다.

본 실험에서는 육안으로 명확하게 잘못 라벨링 된 데이터들만 우선 제거하는 방식으로 정제하고 실험을 하였으며, 그 결과 데이터를 정제하기 전 실험결과 보다 평균 15% 내외의 성능 향상을 보였다(Table 8과 9 참고). 이는 무시하기 힘든 수치의 증가이며 데이터 라벨링 품질을 전반적으로 더 개선하면 보다 나은 학습 결과를 보일 것으로 판단된다.

EAST 모델의 경우 영문만 학습한 상태에서도 한글문자 영역 탐지율이 높게 나오지만 YOLO 모델의 경우는 한글과 영문 모두 학습한 경우 전체적으로 더 나은 문자 탐지 성능을 보이고 있다. 이와 같은 결과는 EAST와 YOLO 신경망의 특성에 따른 결과이기에 이에 대한 후속 연구도 필요하다.

실험결과를 통해 두 신경망 모델 모두 문자열 학습 시, 각각의 문자열에 대한 기하학적인 특성을 학습하고 이러한 특

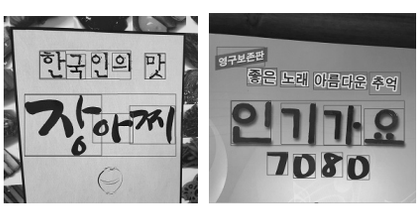
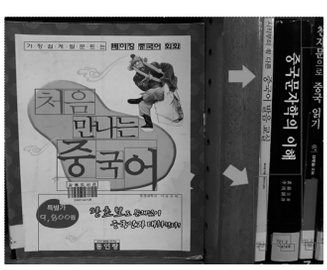
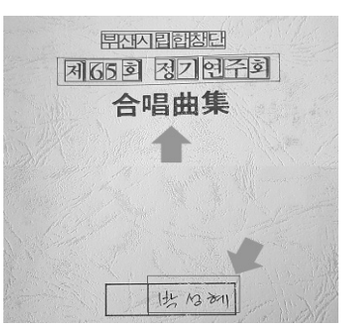


Fig. 14. Example of Accurate Data Labelling



(a) Orange Arrows Indicate Bound Boxes That Does Not Fit Text Area



(b) Top Orange Arrow Indicates Text Area That Should be Box Bounded and Bottom Arrow Indicates Bound Box That Does Not Fit Text Area

Fig. 15. Example of Inaccurate Data Labelling

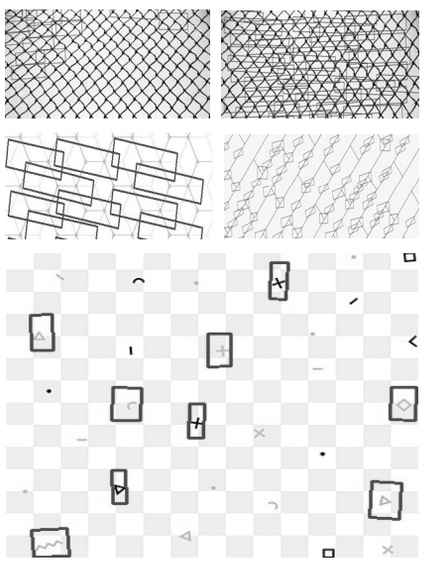


Fig. 16. Examples of False Detection

Fig. 17. Demo Video of EAST and YOLO Neural Networks

성을 통해 문자열을 탐지하는 것으로 유추해 볼 수 있는데 이로 인해 종종 단순 패턴 조합을 포함한 문자열이 아닌 영역에 대해서도 문자 영역으로 오감지하는 경우가 있다. 이러한 사례는 Fig. 16에서 보여주고 있다.

그림에서 보듯이 반복적인 패턴으로 구성된 이미지나 도형을 문자열로 판단하여 단어경계박스를 그리는 것을 알 수 있다.

이런 오인식이 나오는 이유는 이미지에서 문자영역 부분이 엣지(edge)가 발생하는 영역이 많으며 신경망이 이런 엣지 영역을 문자가 있는 영역으로 학습하기 때문으로 추정된다. 문자가 없는데 문자영역으로 오인식한 이미지들은 대체로 기하학적 모양의 배열이 문자열의 배열에서 나타나는 형태와 유사한 엣지가 검출되는 것으로 보이며 그로 인해 발생하게 되는 결과로 보인다.

EAST와 YOLO 모두 이러한 오탐지 사례들을 보이고 있으며 이 부분은 추후 신경망의 각 노드가 활성화 될 때의 데이터 값들을 시각화해 보면 보다 분명하게 이해할 수 있을 것이다. Fig. 17은 본 논문에서 구현한 EAST와 YOLO 신경망들의 성능을 시각화한 것으로 다음 링크[1] 에서 해당 동영상을 확인할 수 있다.

1) www.youtube.com/watch?v=ZpRNfWzuexQ

5. 결 론

본 논문에서는 일반적으로 문자열 탐지 시, 많이 활용되는 EAST 신경망과 최근 가벼우면서 객체 감지에 있어 뛰어난 성능으로 각광을 받고 있는 YOLO 신경망 두 모델을 선택하여 다양한 이미지 속에서 문자열을 탐지하는 실험을 진행하고 그 성능을 비교해 보았다.

실험을 진행하기 전에는 일반적으로 문자열 탐지 분야에서 많이 사용되고 있는 EAST 신경망 모델이 더 나은 성능을 보일 것으로 예상했으나, 실험결과 일반적인 단순한 문자열로 구성된 문서 형태에 대해서는 두 모델이 대등한 성능을 보였으며 생활 속 이미지 문자열은 YOLO가 더 나은 성능을 보여주었다. 가벼우면서 뛰어난 객체 감지 성능 때문에 다양한 분야에서 활용되는 YOLO는 v4와 v5 최신 모델에서 이처럼 객체 뿐만 아니라 문자 영역 탐지에서도 충분히 강력한 성능을 보여주기에 앞으로 문자인식 기술 분야에도 적극적으로 활용될 것으로 예상된다.

이번 연구는 한국형 AI데이터 확보를 위해 구축된 AI Hub의 다양한 데이터들을 이용하여 각 신경망들의 성능을 평가하였다. 그동안 AI기술을 연구함에 있어 표준으로 사용할 만한 국내 데이터들이 전무했던 상황이었으니 과학기술정보통신부에서 구축한 한국형 인공지능 데이터를 활용하여 실험을 한 사례로서도 큰 의의를 가진다고 할 수 있다. 하지만, 학습 데이터의 정밀도가 낮다는 문제는 향후 개선되어야 할 과제로 보인다.

후속 연구에서는 AI Hub 데이터들의 품질을 보다 더 정제하면서 데이터 품질에 따른 신경망 성능 향상에 대한 연구와 ICDAR, MSRA-TD500, CTW-1500 그리고 TotalText 데이터 세트 등 일반적으로 널리 활용되는 데이터 세트에 대해서도 추가적인 비교 실험을 진행하고 관련 코드와 데이터들도 깃허브를 통해 공개할 계획이다. 또한 YOLO를 이용하여 추출된 문자영역의 문자들을 인식하는 신경망까지 구성하여 완전한 문자인식 엔진도 구현하고자 한다. 최근 단어나 문장단위로 신경망을 학습하여 문자인식을 수행하는 방법들이 제안되고 또한 우수한 성능을 보여주고 있기에, 문자영역을 감지하는 YOLO 신경망 모델과 결합한 문자인식 엔진의 성능도 기대해 볼 수 있다.

References

[1] Y. M. Baek, B. D. Lee, D. Y. Han, S. D. Yun, and H. S. Lee, "Character region awareness for text detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp.9365-9374, 2019.

[2] T. Wang, T. Zhu, L. Jin, C. Luo, X. Chen, Y. Wu, and M. Cai, "Decoupled attention network for text recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol.34, No.7, pp.12216-12224, 2019.

[3] P. Lyu, C. Yao, W. Wu, S. Yan, and X. Bai, "Multi-oriented scene text detection via corner localization and region segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.7553-7563, 2018.

[4] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," in *European Conference on Computer Vision*, Springer, Cham, pp.56-72, 2016.

[5] M. Liao, B. Shi, X. Bai, X. Wang, and W. Liu, "Textboxes: A fast text detector with a single deep neural network," in *Thirty-first AAAI Conference on Artificial Intelligence*, 2017.

[6] M. Liao, B. Shi, and X. Bai, "Textboxes++: A single-shot oriented scene text detector," *IEEE Transactions on Image Processing*, Vol.27, No.8, pp.3676-3690, 2018.

[7] F. Jiang, Z. Hao, and X. Liu, "Deep scene text detection with connected component proposals," *arXiv preprint arXiv:1708.05133*, 2017.

[8] Y. Jiang, et al., "R2cnn: rotational region cnn for orientation robust scene text detection," arXiv preprint arXiv:1706.09579, 2017.

[9] X. Zhou, C. Yao, H. Wen, Y. Wang, S. Zhou, W. He, and J. Liang, "East: an efficient and accurate scene text detector," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.5551-5560, 2017.

[10] H. Hu, C. Zhang, Y. Luo, Y. Wang, J. Han, and E. Ding, "Wordsup: Exploiting word annotations for character based text detection," in *Proceedings of the IEEE International Conference on Computer Vision*, pp.4940-4949, 2017.

[11] S. Long, J. Ruan, W. Zhang, X. He, W. Wu, and C. Yao, "Textsnake: A flexible representation for detecting text of arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp.20-36, 2018.

[12] T. He, Z. Tian, W. Huang, C. Shen, Y. Qiao, and C. Sun, "An end-to-end textspotter with explicit alignment and attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.5020-5029, 2018.

[13] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes," in *Proceedings of the European Conference on Computer Vision (ECCV)*, pp.67-83, 2018.

[14] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single shot text detector with regional attention," in *Proceedings of the IEEE International Conference on Computer Vision*, pp.3047-3055, 2017.

[15] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.779-788, 2016.

[16] S. Qin and R. Manduchi, "Cascaded segmentation-detection networks for word-level text spotting," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, Vol.1, pp.1275-1282, 2017.
[17] A. Bochkovskiy, C. Y. Wang, and H. Y. M.. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
[18] G. Jocher, K. Nishimura, T. Mineeva, R. Vilariño, GitHub repository [Internet], https://github.com/ultralytics/yolov5
[19] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
[20] X. Wang, S. Zheng, C. Zhang, R. Li, and L. Gui, "R-YOLO: A real-time text detector for natural scenes with arbitrary rotation," *Sensors*, Vol.21, No.3, pp.888, 2021.

박 찬 용
https://orcid.org/0000-0002-7814-0260
e-mail : bw_yong13@tuat.kr
2014년 계명대학교 게임모바일콘텐츠(학사)
2021년 경북대학교 컴퓨터학부(석사)
2016년 ~ 현 재 ㈜투아트 과장
관심분야 : Cloud Computing & Deep Learning

임 영 민
https://orcid.org/0000-0003-1312-4697
e-mail : youngmin@tuat.kr
2013년 계명대학교 게임모바일콘텐츠(학사)
2015년 경북대학교 컴퓨터학부(석사)
2016년 ~ 현 재 ㈜투아트 과장
관심분야 : Cloud Compuiting & Front-end Design

정 승 대
https://orcid.org/0000-0003-1303-5766
e-mail : sdjeong@tuat.kr
2007년 경북대학교 컴퓨터공학과(석사)
2010년 경북대학교 컴퓨터학부(박사수료)
2019년 ~ 현 재 ㈜투아트 개발이사
관심분야 : Computer Vision & Virtual Reality

조 영 혁
https://orcid.org/0000-0003-1383-2482
e-mail : Philip@tuat.kr
2019년 ~ 현 재 ㈜투아트 부사장
관심분야 : Energy-aware Computing & Cloud Computing

이 병 철
https://orcid.org/0000-0003-1854-2214
e-mail : bclee@live.co.kr
1993년 경북대학교 전자공학과(석사)
2021년 대구가톨릭대학교 신소재화학공학과(박사수료)
2020년 ~ 현 재 (재)경상북도경제진흥원 일자리산업실 실장
관심분야 : AI-based image processing, hologram, superhydrophobic

이 규 현
https://orcid.org/0000-0001-7981-9243
e-mail : prodzpod@protonmail.com
2016년 ~ 현 재 경북대학교 컴퓨터공학부 학사과정
관심분야 : Cloud Computing & Server Consolidation

김 진 욱
https://orcid.org/0000-0002-6000-3073
e-mail : deepkaki@knu.ac.kr
1994년 경북대학교 컴퓨터공학과(학사)
1996년 경북대학교 컴퓨터공학과(석사)
2009년 경북대학교 컴퓨터공학과(박사)
2018년 ~ 2020년 대구창조경제혁신센터 실장
2020년 ~ 현 재 경북대학교 컴퓨터학부 초빙교수
관심분야 : Machine Learning, HCI