

# Object Detection과 OCR을 활용한 다중 책 제목 검출 모델

김성연\*<sup>1</sup>, 박진우\*\*<sup>2</sup>, 김상민\*\*\*<sup>3</sup>, 나웅\*<sup>2</sup>, 장유진\*\*\*\*<sup>4</sup>

## Multi-book Label Detection Model using Object Detection and OCR

Seongyeon Kim\*<sup>1</sup>, Jinwoo Park\*\*<sup>2</sup>, Sangmin Kim\*\*\*<sup>3</sup>, Yung Na\*<sup>2</sup>, and Yujin Jang\*\*\*\*<sup>4</sup>

본 논문은 모두의연구소 지원을 받아 작성되었습니다

### 요약

본 논문에서는 객체 탐지와 광학 문자 인식(이하 OCR)을 복합적으로 사용하여 책장 내의 다수의 책이 꽂혀 있을 때 책등을 탐지하여 책 제목을 추출하고 이를 통해 책에 대한 추가적인 정보를 확인할 수 있는 모델을 구현한 결과에 대해 다루고 있다. 객체 탐지 과정은 책등과 책 제목 영역 탐지로 단계를 분리하였고 2개의 YOLOv5 모델을 사용하여 객체 탐지에 대한 성능을 높였다. OCR 단계에서는 출력되는 텍스트의 정확도를 높이기 위하여 이미지가 OCR 모델에 입력되기 전 단계에서 흑백 변환, 크기 변환, 밝기 조절과 같은 전처리를 수행하였다. 느린 OCR 처리 시간에 대한 문제를 보완하기 위해 설계한 모델을 웹 애플리케이션에 적용하는 과정에서 사용자가 원하는 책을 선택하는 기능을 추가하였다. 추가적으로 객체 탐지와 광학 문자 인식을 통해 추출된 제목을 이용해 저자, 출판사, 가격, 후기와 같은 정보를 확인할 수 있도록 하였다.

### Abstract

In this paper, we discuss the results of implementing a model that uses a combination of object detection and Optical Character Recognition(OCR) to detect book spine and extract book titles when multiple books in a bookcase are plugged in, thereby confirming additional information about the book. The object detection process was divided into book spine detection and book title area detection, and two YOLOv5 models were used to increase the performance of object detection. As an OCR result, pre-processing such as grayscale, resizing, and brightness adjustment was performed before entering images into OCR model to increase the accuracy of text output. In the process of applying a model designed to compensate for the problem of slow OCR processing time to web applications, it has added a function that allows users to select the book they want. In addition, information such as author, publisher, price, and reviews can be confirmed through book titles extracted through object detection and OCR.

### Keywords

artificial intelligence, deep learning, yolov5, google cloud vision, object detection, optical character recognition

\* 모두의연구소

- ORCID<sup>1</sup>: <https://orcid.org/0000-0003-3268-4630>

- ORCID<sup>2</sup>: <https://orcid.org/0000-0002-5968-7076>

\*\* 중앙대학교 AI대학원 석사과정(교신저자)

- ORCID: <https://orcid.org/0000-0003-4927-1058>

\*\*\* 성균관대학교 대학원 전자전기컴퓨터공학과 석.박사통합과정연구원

- ORCID: <https://orcid.org/0000-0002-3768-590X>

\*\*\*\* 동국대학교 컴퓨터공학전공 학사과정

- ORCID: <https://orcid.org/0000-0001-8382-5232>

· Received: Aug. 31, 2022, Revised: Feb. 17, 2023, Accepted: Feb. 20, 2023

· Corresponding Author: Jinwoo Park

B310 #624, Chung-Ang University, 84, Heukseok-ro, Dongjak-gu, Seoul, Republic of Korea

Tel.: +82-02-401-4981, Email: jinwoo1203@gmail.com

## 1. 서 론

현재 OCR(Optical Character Recognition)은 다양한 분야(데이터 입력 자동화, 문서 편집, 문서 분류 등)에서 활용되고 있다. 그리고 구글, 카카오, 네이버 등 이러한 OCR을 제공하는 서비스도 다양해지고 있다[1]-[3]. 다양한 활용성에 비해 이러한 기술에 한계도 있다. 제공되는 OCR을 사용해 보면 OCR이 이미지 전체적으로 적용이 되기 때문에 특정 위치에만 OCR을 적용해야 할 경우에는 그에 맞는 템플릿을 사용해야 한다. 그렇지 않으면 이미지 내에 있는 모든 텍스트가 뒤섞여서 출력되는 경우가 발생한다. 또한, 이미지가 회전된 경우에는 OCR이 제대로 적용되지 않는 문제가 있다. 이러한 문제점에서 착안하여 이 논문에서는 원하는 위치에만 OCR을 적용하여 그 부분의 문자만을 추출하는 모델을 제안한다. 이 모델은 크게 3개의 단계로 구성된다. 사전 학습된 object detection 모델에 추가 학습을 진행하여 OCR을 적용하고자 하는 위치를 탐지하고 (stage 1, stage 2) 해당 위치에만 OCR을 적용하여 텍스트를 추출한다(stage 3). 자세한 모델 설명은 모델 설계 및 학습 부분에서 다룬다.

이와 비슷한 과제를 수행한 논문으로 도서관 책 재고관리를 위해 책 제목을 추출하는 모델을 제안한 [4][5]가 있다. 우리는 [4][5]와 달리 object detection과 OCR을 결합한 방식을 사용하였고 이러한 모델을 실생활에서 사용되는 서비스 중 하나인 책 검색 서비스에 적용하였다. 책 검색 서비스는 책 표지 이미지를 찍으면 이미지로부터 책 제목, 저자, 출판사 등을 텍스트로 뽑아내어 이미지 속의 책을 검색해주는 서비스이다. 우리는 이러한 책 제목 검출 서비스를 책장에 다수의 책이 꽂혀 있는 이미지가 제공될 때 해당 이미지 내에 있는 책들 중 책 제목의 위치만을 탐지한 후 책 제목을 텍스트로 변환시켜 여러 권의 책을 한번에 검색해주는 모델을 구현하였다. 추가적으로 실제 서비스에서 사용될 때의 속도 개선을 위해 선택적으로 모델을 적용하는 방법도 제시한다.

## II. 데이터 수집 및 데이터 전처리

본 논문에서는 책장 내 다수의 책 정보를 추출하는 모델을 구축하기 위해 책장에 수직으로 꽂혀있는 책등 사진과 한 권의 책등 이미지 데이터를 학습에 사용하였다.

데이터 정보는 표 1과 같다. 우선, 3024x3024 사이즈의 다수의 책이 꽂혀있는 책장 이미지 1500장을 수집하였다. 그 후 수집한 책장 이미지 1500장을 4분할하여 1512x1512 사이즈로 변환함으로써 총 2480장의 데이터를 확보하였다. 수집한 데이터에서 책장 내의 책이 기울어진 이미지, 빈 책장 이미지, 책이 가로로 눕혀져 있는 이미지는 제거하였다.

다수의 책이 존재하는 책장에서 책등을 탐지하는 stage 1의 YOLOv5 모델 학습을 위해 총 2480장의 책장 이미지 데이터 중 871장을 사용하였으며 Labelimg를 이용하여 책장 내에 수직으로 위치한 책등에 대한 라벨링을 진행하였다[6]-[8].

단 권의 책등 이미지에서 제목 영역을 탐지하는 stage 2의 YOLOv5 모델 학습을 위해 책등 이미지 데이터 1524장을 사용하였다. 데이터 중 제목 부분에 일러스트 및 부제가 과도하게 들어간 데이터는 제거하였고 나머지 데이터에 대해서 Labelimg를 이용하여 제목 영역에 대한 라벨링을 진행하였다. 각 단계에 사용되는 데이터의 형태는 그림 1과 같다.

표 1. 이미지 데이터 세트  
Table 1. Image data set

Collected	Preprocessed	Train data (stage 1)	Train data (stage 2)
1500 (3024x3024)	2395 (1512x1512)	871	1524

## III. 모델 설계 및 학습

논문에서 사용된 모델은 그림 2와 같이 총 3개의 단계(stage 1, stage 2, stage 3)로 구성된다. Stage 1과 stage 2는 object detection을 위한 과정으로 stage 1에서는 이미지 내에 있는 책들의 책등을 탐지하는 단계이고 stage 2는 stage 1에서 탐지한 책등 영역에서 책 제목 부분을 다시 탐지하는 단계이다[9]. 마지막 stage 3는 stage 2를 통해 검출된 제목 영역에 OCR을 적용해 제목 텍스트를 뽑아내는 단계이다.

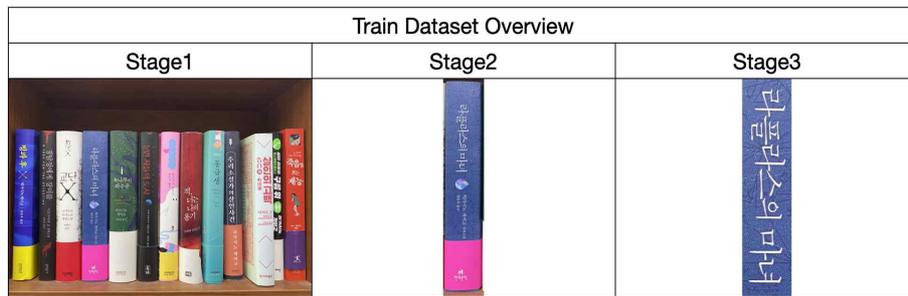


그림 1. 단계별 학습 데이터셋 예시  
Fig. 1. Train dataset example per stage

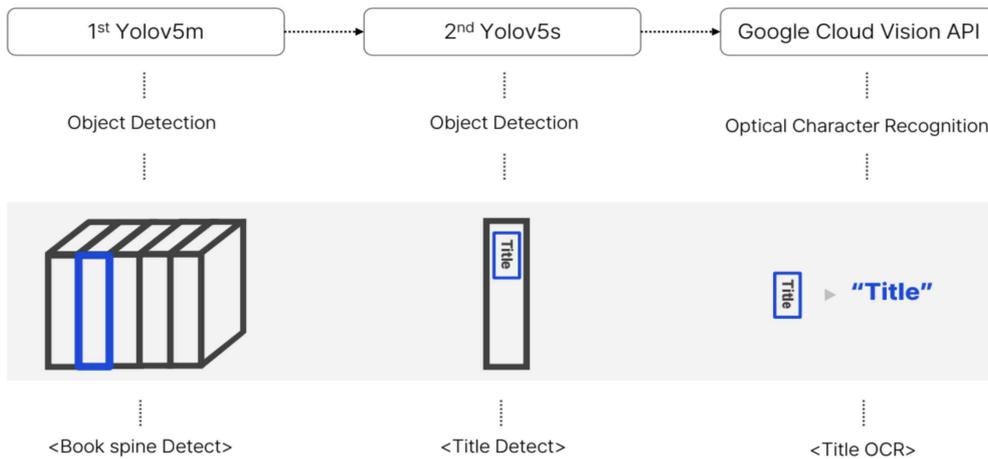


그림 2. 모델 설계  
Fig. 2. Model design

모델을 3개의 단계로 나누는 이유는 원본 이미지에 바로 OCR을 적용할 경우 책등에 있는 책 제목, 저자, 출판사 등의 텍스트가 뒤섞여서 출력되는 문제가 있었기 때문에 이를 방지하기 위함이다.

Stage 1은 원본 이미지를 넣어줬을 때 이미지 내에 있는 각각의 책에서 책등 부분만을 탐지하기 위한 단계로 object detection 모델을 사용하였다. Object detection을 위해 사용한 모델은 COCO dataset으로 pretrained된 Yolov5m 모델에 우리가 수집한 학습데이터 721장을 넣어 추가 학습을 시킨 모델을 사용하였다. Optimizer는 Stochastic Gradient Descent(SGD)를 사용하였고 batch size는 16으로 설정하였다. 다른 하이퍼 파라미터 값은 동일하게 유지한 상태에서 epochs만 변경하며 모델을 학습시켰다. 학습된 모델들에 test dataset을 넣어줬을 때 epochs 100으로 설정한 모델이 가장 높은 recall값과 인식률을 보여주어 epochs 100으로 설정한 모델을 사용하였다.

표 2. stage 1 모델 설정  
Table 2. Model configuration of stage 1

Input size	Batch size	Epochs	Weights	Optimizer
640x640	16	50, 100, 150	Yolov5m *	Stochastic gradient descent

\* Pretrained by COCO dataset[6]

Stage 2는 stage 1에서 탐지한 책등 이미지를 입력 데이터로 받아 책등 내에서도 책 제목만을 다시 탐지하는 단계로 object detection 모델을 사용하였다. Object detection을 위해 사용한 모델은 COCO dataset으로 pretrained된 Yolov5s 모델에 표 1에 있는 책등 이미지만으로 구성된 stage 2용 train data 1524장을 사용하여 추가 학습을 시킨 모델을 사용하였다. 다른 하이퍼파라미터 설정은 stage 1과 동일하고 epochs만 150, 200으로 설정하여 각각 모델을 학습시켰다.

#### 4 Object detection과 OCR을 활용한 다중 책 제목 검출 모델

학습된 모델들에 test dataset을 넣어줬을 때 epochs 200으로 설정한 모델이 가장 높은 recall값과 인식률을 보여주어 해당 모델을 선택하여 사용하였다.

표 3. stage 2 모델 설정  
Table 3. Model configuration of stage 2

Input size	Batch size	Epochs	Weights	Optimizer
640x640	16	150, 200	Yolov5s *	Stochastic gradient descent

\* Pretrained by COCO dataset[6]

Stage 3는 stage 2에서 탐지한 책 제목 이미지를 입력데이터로 받아 제목부분을 텍스트로 추출하는 단계로 OCR 기법을 사용하였다. OCR 모델로는 Google Cloud Vision API를 활용하였다. 해당 모델을 바로 사용할 경우 성능이 좋지 않아 OCR 모델에 데이터를 넣어주기 전 전처리를 진행하였다 [10][11]. 전처리로는 1.5배 resize한 후 보간법으로 INTER AREA를 적용하였고, 흑백변환 후 밝기조절을 적용하였다. 전처리가 적용된 입력데이터를 OCR 모델에 넣어 결과값으로 제목 텍스트를 출력한다. 출력된 제목 텍스트에도 특수문자, 공백제거, 중복 단어 제거와 같은 후처리를 적용하여 최종 결과물을 도출하였다.

이 논문에서 제안한 모델은 OCR을 적용해야 할 책이 많아질 경우 속도가 느려지는 문제가 있었다. 설계한 모델을 웹 애플리케이션에 적용하여 사용성을 높이고 느린 OCR 속도 문제를 보완하고자 하였다. 사용자가 웹 애플리케이션에 접속하여 책장의 이미지를 업로드하면 stage 1의 객체 탐지 모델이 추출한 bbox를 기반으로 사용자가 원하는 책(책등)을 선택할 수 있도록 하였다. 사용자가 책을 선택하면 stage 2의 객체 탐지 모델과 OCR 모델을 거쳐 최종 결과물을 도출하였다.

### IV. 학습 결과 및 정량 평가

#### 4.1 Stage 1

Stage 1의 학습 결과는 표 4와 같다. 모델 특성상

책등을 놓치지 않고 탐지하는 것이 중요하기 때문에 Recall값을 위주로 확인했고 epochs 100에서 0.96으로 가장 높은 것으로 나타났다. 모델에 대한 테스트는 테스트 데이터 30장(학습에 사용되지 않은 책이 꽂혀있는 이미지로 총 517권의 책이 있음)으로 진행했다. 테스트 결과는 인식률과 오검출 비율을 비교하였다. 표 5와 표 6을 보면 confidence threshold (이하 conf)는 0.35, epochs은 100으로 설정한 모델이 인식률은 가장 높고 오검출 비율은 낮았다. Yolov5를 적용하여 학습 및 평가를 진행했고 stage 3는 Google Cloud Vision API를 이용했다. stage 1 모델의 학습은 학습데이터 721장, 검증데이터 150장을 이용해 진행했다. 모델 환경 설정은 표 2와 같다.

표 4. Stage 1 학습 결과  
Table 4. Train result of stage 1

Epochs	50	100	150
Box loss	0.0281	0.02568	0.02454
Obj loss	0.06253	0.05613	0.05203
Cls loss	0	0	0
Precision	0.9275	0.9254	0.9385
Recall	0.9546	<b>0.9612</b>	0.9413
mAP(0.5)	0.9699	<b>0.9704</b>	0.96354
mAP (0.5 ~ 0.95)	0.7674	0.7849	0.774

표 5. Stage 1 epochs, conf.에 따른 인식률  
Table 5. Recognition rate of stage 1 by epochs and conf.

Conf.	Epochs		
	50	100	150
0.25	95.36%	91.49%	91.30%
0.3	94.78%	90.91%	90.52%
0.35	93.62%	<b>89.94%</b>	89.56%
0.4	90.91%	89.17%	82.98%

\* Recognition rate : (# Detected bbox / # Labeled bbox)

표 6. Stage 1 오검출 개수  
Table 6. Number of misdetection for stage 1

Conf.	Epochs		
	0.3	<b>0.35</b>	0.4
Recognize book cover as spine	1	<b>0</b>	0
Recognize two book as one book	2	<b>1</b>	1
Recognize one book as two book	1	<b>1</b>	1
# Total misdetection	4	<b>2</b>	2

### 4.2 Stage 2

Stage 2 모델의 평가는 S1, S2 두 모델에 대해 테스트 데이터 325장(학습에 사용되지 않은 날권의 책등 이미지)으로 진행했고 labeled bbox와 predicted bbox의 IoU(Intersection of Union. 이하 IoU)값의 범위에 따른 bbox 검출 개수와 검출 비율을 비교하였다. Labeled bbox와 predicted bbox를 시각화해 확인해 본 결과 IoU가 0.6이상인 경우만 활용할 수 있는 유의미한 결과라고 판단하여 0.6이상인 경우만 비교해 보았다. IoU 범위에 따른 검출결과를 보면 표 7과 같다. IoU값이 0.6이상에서 검출 개수와 검출 비율 모두 S2모델이 더 높은 것으로 확인할 수 있다. Stage 2에서 책 제목 영역을 검출하는 것 자체에 목적이 있기 때문에 재현율(Recall)을 고려하여 IoU를 조절한 결과를 비교하였다. 책 제목 외의 다른 요소가 일부 함께 검출 되어 정확도가 떨어지더라도 올바른 영역을 검출하는 것을 목표로 하였다.

표 7. Stage 2 bounding box 검출 개수  
Table 7. Stage 2 # of detected bounding box

IoU	# Detected bbox		Bbox detection rate	
	S1.	S2.	S1.	S2.
0.9 - 1.0	154	175	47.38	53.85
0.8 - 1.0	244	256	75.08	78.77
0.7 - 1.0	265	268	81.54	82.46
0.6 - 1.0	<b>275</b>	<b>281</b>	<b>84.62</b>	<b>86.46</b>

### Evaluation method

- **Step1.** Use SequenceMatcher in the difflib library to compare the actual title with the predicted value to calculate accuracy (0-100%).
- **Step2** Calculating Accuracy by Error Range

Number of data within the reference error / Range total number of data X100	
Error Range	Details
0 %	100% Accuracy Data
0 ~ 10 %	Data with 90% or more accuracy
0 ~ 20 %	Data with 80% or more accuracy

### 4.3 Stage 3

Stage 3 모델의 평가는 제목 크롭 이미지 380장을 이용하여 진행했고 전처리 기법으로 사용한 1.5배 resize, 흑백변환, 대비조절, 밝기조절에 따른 정확도와 오차 범위 별 성능을 비교하였다. 정확도는 그림 3과 같이 difflib 라이브러리의 SequenceMatcher를 사용해 실제 제목과 예측 값을 비교하여 %로 나타내었다. 그림 3의 우측 예시와 같이 입력값과 OCR 결과 값을 일치하는 글자를 기준으로 비교하여 일치율을 계산한 후 각 일치율 범위에 해당하는 데이터 수와 전체 데이터 수를 비교하여 오차범위에 따른 정확도를 계산하였다. 오차 범위에 따른 정확도를 계산한 식은 그림 3의 좌측에 표기한 내용과 같다.

Stage 2에서 IoU를 조절함에 따라 책제목 외의 요소도 탐지하여 OCR의 정확도가 떨어지는 문제를 보완하기 위해 OCR 단계에서 이미지 전처리 기법을 적용하였다. 그림 4와과 같이 여러 전처리 기법을 혼합하여 적용한 결과에 따른 성능을 비교하였다. 비교 결과 흑백변환, resize(영역), 밝기조절을 적용했을 때 정확도가 79%로 가장 높았다. 그림 5는 전처리 기법이 적용된 이미지에 대해 오차 범위(그림 3에 대한 설명 참고)에 따른 성능을 비교한 결과를 보여준다. 오차범위 10%의 결과까지만 비교해 보았을 때 흑백변환, resize(영역), 밝기조절을 적용했을 때 89%로 가장 높은 것을 확인할 수 있다.

**Step1.** Compare the actual title with the predicted title



**Step2.** Calculating Accuracy by Error Range

Accuracy	data	Error Range	Accuracy
100%	262	0%	74%
More than 90%	294	0~10%	85%
More than 80%	324	0~20%	88%
Total	380		

그림 3. Stage 3 오차 범위별 정확도  
Fig. 3. Accuracy of stage 3 by arrange of error

6 Object detection과 OCR을 활용한 다중 책 제목 검출 모델

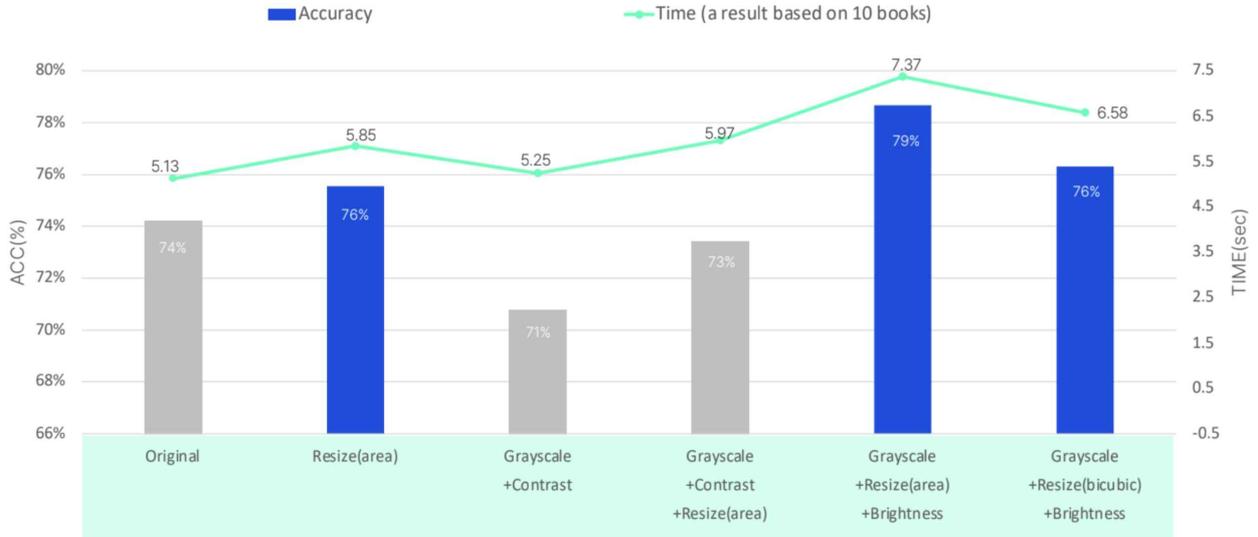


그림 4. Stage 3 전처리 기법에 따른 성능 비교  
 Fig. 4. Performance comparison of stage 3 by preprocessing method

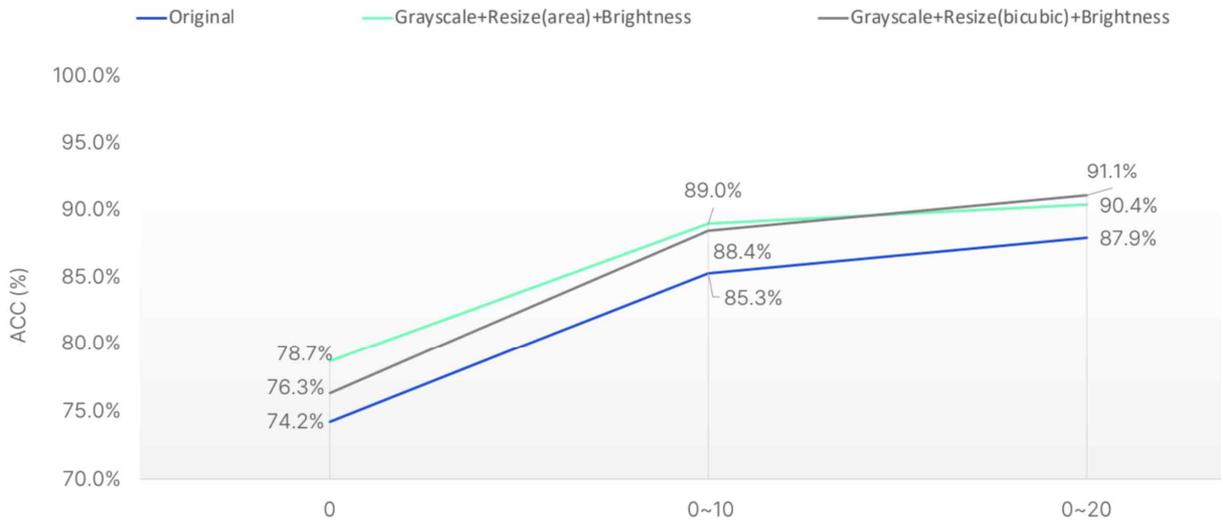


그림 5. Stage 3 오차 범위에 따른 성능 비교  
 Fig. 5. Performance comparison of stage 3 by the range of error

전체 모델을 설계한 후 stage 1부터 stage 3까지의 전체 처리 시간을 확인하였다. 표 8은 처리해야 할 책의 수에 따른 전체 처리 시간을 비교한 결과를 나타낸다. 그 결과를 확인해보면, 책 1권을 처리하는데 걸리는 시간이 0.95초이고 10권을 처리하는데 걸리는 시간은 8초로 증가한다. 각 단계의 처리 시간 중 stage 3 단계의 시간이 가장 긴 것을 확인할 수 있고 책 권수가 많아짐에 따라 처리 시간이 더 길어지는 것을 볼 수 있다.

표 8. 총 처리시간

Table 8. Total processing time

	1 book	2 books	3 books	4 books	5 books
stage 1	0.14	0.12	0.14	0.14	0.15
stage 2	0.01	0.02	0.02	0.03	0.03
stage 3	0.79	1.52	2.5	3.36	3.82
<b>Total</b>	<b>0.95</b>	<b>1.67</b>	<b>2.67</b>	<b>3.54</b>	<b>4.01</b>
	6 books	7 books	8 books	9 books	10 books
stage 1	0.14	0.15	0.15	0.15	0.16
stage 2	0.04	0.04	0.05	0.06	0.06
stage 3	4.73	5.57	5.82	6.63	7.76
<b>Total</b>	<b>4.92</b>	<b>5.78</b>	<b>6.04</b>	<b>6.85</b>	<b>8</b>

## V. 결론 및 향후 과제

본 논문의 모델은 다수의 책이 있는 책장에서 여러권의 책 제목을 한번에 인식하는 기능을 제공한다. 본 논문에서는 책등 영역과 책 제목 영역을 검출하는 과정을 두 단계로 분리하여 2개의 YOLOv5 모델을 사용해서 객체 탐지의 정확도를 높이고자 하였다.

각 단계에서 정량 평가를 실시하여 stage 1 단계에서는 confidence threshold 0.35, epochs 100으로 학습한 모델, stage 2 단계에서는 epochs 200으로 학습한 모델을 사용하였다. stage 3 단계에서는 정량평가를 통해 입력 이미지에 흑백변환, resize(영역), 밝기조절을 적용하도록 하였다.

이미지 전처리를 통해 문자 인식 성능은 상승하였지만, OCR 단계에서 처리 속도가 길어지는 문제가 여전히 남아있었다. 이에 웹 애플리케이션에 해당 모델을 적용할 때 책등 검출 단계에서 사용자가 원하는 책을 선택하는 기능을 추가하여 느린 OCR 처리 시간에 대한 문제를 보완하였다.

향후에는 수직으로 세워진 책뿐만 아니라 기울어진 책에도 적용 가능하도록 Rotate YOLOv5 또는 객체 분할(Object segmentation)을 적용하고자 하며, 디자인 폰트에 대한 인식률을 향상시키고 실행 시간을 단축시키기 위해 오픈 소스를 활용하여 OCR 모델을 제작하고자 한다.

## References

[1] Kakao OCR, <https://www.kakaoicloud.com/service/detail/6-9> [accessed: Feb. 05, 2022]

[2] Naver Clova OCR, <https://clova.ai/ocr> [accessed: Feb. 05, 2022]

[3] Google Cloud Vision, <https://cloud.google.com/vision> [accessed: Feb. 05, 2022]

[4] X. Yang, et al., "Smart library: Identifying books in a library using richly supervised deep scene text reading", arXiv preprint, Nov. 2016. <https://doi.org/10.48550/arXiv.1611.07385>.

[5] M. P. Nevetha and A. Baskar, "Automatic book

spine extraction and recognition for library inventory management", Proc. of the Third International Symposium on Women in Computing and Informatics, Kochi India, pp. 44-48, Aug. 2015. <https://doi.org/10.1145/2791405.2791506>.

[6] Tsutalin, "LabelImg", <https://github.com/tsutalin/labelImg> [accessed: Feb. 05, 2022]

[7] Ultralytics, "Yolov5", <https://github.com/ultralytics/yolov5> [accessed: Feb. 05, 2022]

[8] M. Karthi, V. Muthulakshmi, R. Priscilla, P. Praveen, and K. Vanisri, "Evolution of yolo-v5 algorithm for object detection: Automated detection of library books and performance validation of dataset", 2021 International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems(ICSES), Chennai, India, pp. 1-6, Sep. 2021. <https://doi.org/10.1109/ICSES52305.2021.9633834>.

[9] J. S. Choi, D. Jung, K. Min, and B. Lee, "Multi-stage Object Tracking Technique for Label Recognition", Proc. of the Korea Information Processing Society Conference, pp. 972-975, Oct. 2019. <https://doi.org/10.3745/PKIPS.y2019m10a.972>.

[10] M. Brisnello, R. Grbic, M. Pul, and T. Andelic, "Improving Optical Character Recognition performance for low quality images", 2017 International Symposium ELMAR, Zadar, Croatia, Sep. 2017. <https://doi.org/10.23919/ELMAR.2017.8124460>.

[11] C. B. Jang and K. B. Kim, "Implementation of Pre-Post Process or Accuracy Improvement of OCR Recognition Engine Based On Deep-Learning Technology", Journal of Convergence for Information Technology, Vol. 12, No. 1, pp. 163-170, Jan. 2022. <http://doi.org/10.22156/CS4SMB.2022.12.01.163>.

저자소개

김 성 연 (Seongyeon Kim)



2019년 8월 : 한림대학교 청각학과 (이학사)  
2021년 9월 ~ 2022년 3월 : 모두의연구소(아이펠)  
2022년 6월 ~ 2022년 11월 : 가톨릭대학교 산학협력단 연구원  
관심분야 : Computer Vision,

Deep Learning

박 진 우 (Jinwoo Park)



2018년 2월 : 중앙대학교 도시계획부동산학과(학사)  
2021년 9월 ~ 2022년 3월 : 모두의연구소(아이펠)  
2022년 9월 ~ 현재 : 중앙대학교 AI대학원 석사과정  
관심분야 : Multimodal, NLP

김 상 민 (Sangmin Kim)



2021년 2월 : 목원대학교 지능로봇공학과(공학사)  
2022년 2월 ~ 현재 : 성균관대학교 대학원 전자전기컴퓨터공학과 제어로봇 연구실 석·박통합과정 연구원  
관심분야 : Robotics, Computer

Vision, Deep Learning

나 용 (Yung Na)



2019년 2월 : 숭실대학교 기계공학과 공학사  
2021년 2월 : 숭실대학교 일반대학원 기계공학부 공학석사  
2022년 10월 ~ 현재 : 모두의연구소 교육담당  
관심분야 : 임베디드 시스템,

정밀농업, 적정기술

장 유 진 (Yujin Jang)



2021년 9월 ~ 2022년 3월 : 모두의연구소(아이펠)  
2019년 3월 ~ 현재 : 동국대학교 컴퓨터공학전공 학사과정  
관심분야 : 객체 인식(Object Detection), 딥러닝을 통한 의료영상 분석