Computer Systems
Science and Engineering

Tech Science Press

DOI: 10.32604/csse.2023.038319
Article

Automatic Examination of Condition of Used Books with YOLO-Based Object Detection Framework

Sumin Hong[1] **and Jin-Woo Jeong**[2,*]

[1]Department of Industrial Engineering, Seoul National University of Science and Technology, Seoul, 01811, Korea
[2]Department of Data Science, Seoul National University of Science and Technology, Seoul, 01811, Korea

*Corresponding Author: Jin-Woo Jeong. Email: jinw.jeong@seoultech.ac.kr

Received: 07 December 2022; Accepted: 25 April 2023; Published: 28 July 2023

Abstract: As the demand for used books has grown in recent years, various online/offline market platforms have emerged to support the trade in used books. The price of used books can depend on various factors, such as the state of preservation (i.e., condition), the value of possession, and so on. Therefore, some online platforms provide a reference document to evaluate the condition of used books, but it is still not trivial for individual sellers to determine the price. The lack of a standard quantitative method to assess the condition of the used book would confuse both sellers and consumers, thereby decreasing the user experience of the online secondhand marketplace. Therefore, this paper discusses the automatic examination of the condition of used books based on deep learning approaches. In this work, we present a book damage detection system based on various You Only Look Once (YOLO) object detection models. Using YOLOv5, YOLOR, and YOLOX, we also introduce various training configurations that can be applied to improve performance. Specifically, a combination of different augmentation strategies including flip, rotation, crop, mosaic, and mixup was used for comparison. To train and validate our system, a book damage dataset composed of a total of 620 book images with 3,989 annotations, containing six types of damages (i.e., Wear, Spot, Notch, Barcode, Tag, and Ripped), collected from the library books is presented. We evaluated each model trained with different configurations to figure out their detection accuracy as well as training efficiency. The experimental results showed that YOLOX trained with its best training configuration yielded the best performance in terms of detection accuracy, by achieving 60.0% (mAP@.5:.95) and 72.9% (mAP@.5) for book damage detection. However, YOLOX performed worst in terms of training efficiency, indicating that there is a trade-off between accuracy and efficiency. Based on the findings from the study, we discuss the feasibility and limitations of our system and future research directions.

Keywords: Deep learning; object detection; damage detection; book damage

This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1 Introduction

Recently, the used book market has been growing steadily. The used book market varies greatly in scale and forms, from online bookstores to direct transactions between individuals. According to Persistence Market Research [1], the global used book market increased at a compound annual growth rate (CAGR) of 4.9% over the five years from 2017 and reached 22.77 billion dollars in 2021. The market in 2022 holds a market value of 24.03 billion dollars and is anticipated to grow at a CAGR of about 6.6% till 2032, resulting in 45.53 billion dollars estimated. In particular, popular social media platforms and live broadcasting services (e.g., Instagram and YouTube) have made it easier for people to access and consume the content of books in a variety of genres. For example, with "YouTube Shorts", a spot to share short videos (i.e., less than 1 min), users now create and share a lot of video content regarding both fiction as well as non-fiction books (e.g., biographies, travel books, how-to books, etc.). Furthermore, nowadays numerous movies and television series based on original fiction, comics, and webtoons have been released and are receiving a lot of interest and favorable comments from the audience. This new trend of content creation, sharing, and consumption has increased consumers' interest in the original content and also contributed to the growth of the on/off-line book market. Moreover, as online secondhand market platforms emerge, the demand for online used-book sales is increasing as well. The online secondhand transaction is one of the emerging trends among modern people who value practical and economic matter. The online used-book market now attracts a lot of people with its various unique characteristics. For example, online used book shops can be a channel to find unique or worthy collections. As the Internet connects sellers and buyers anytime anywhere, transaction of worthy books, such as the first edition of a famous work and a limited edition of a particular work is now available [2].

In the used book market, the price of a particular book is usually determined based on various factors, such as a condition, the value of possession, the rareness, and the level of stock (i.e., how many a store holds a particular book). In the case of the state of preservation, most bookstores offer guidelines to help used-book sellers determine the status of their books [3]. However, a quantitative assessment of the condition of the used book is not straightforward, so a consistent and objective assessment is still impractical. Even though the quality assessment is an important step in determining the price of the used book, it has still been manually made by booksellers according to their arbitrary guidelines. For example, Amazon booksellers can choose the condition of books using a predefined category (i.e., New, Used-Like New, Used-Very Good, Used-Good, Used-Acceptable, and Unacceptable) when registering the product. In this course, a guideline to check the condition of a book is provided to the seller. For example, the Amazon guideline defines that the "Used-Like New" and "Used-Very Good" categories may include "Minor cosmetic defects" on the book, such as marks, wears, cuts, bends, and crushes. Therefore, each bookseller should refer to this guideline and manually assign the rating when registering their products, which largely depends on a subjective evaluation. This will result in confusion between the seller and the buyer, which raises quality control issues for both customers and sellers. Continuous customer issues regarding product quality can cause a loss of customer trust and reluctance to further (re-) purchases. Therefore, there is an increasing demand for a system to check the condition of a book in an objective manner, even though there still exist various challenging issues to overcome.

This study tries to solve the aforementioned problem with a deep-learning-based object detection method. Object detection is a computer vision task for identifying a category and a location of an object in images or videos. We propose a technique to automatically inspect the quality of used books by quantifying the type, size, and the number of book damages. For this, first, a set of used books in various genres was collected, then front/back cover and side images of them were captured from various angles, views, and distances. The images were examined to define a set of classes that can represent book damages commonly found in the dataset. The damage classes defined in this work include *Wear* (i.e., worn-out paper), *Spot* (i.e., contaminated marks), and so on. More explanation of the book damage classes will be provided in Section 3.1. Afterward, we manually annotated the collected images for generating a dataset for the book damage detection task. Based on the constructed dataset, various object detection models were trained and evaluated. The object detection model used in our framework is You Only Look Once (YOLO) series which is widely known for its high performance in terms of detection accuracy and processing speed, which can make a system more practical for real-world applications. For example, lightweight detection models can be easily integrated into a resource-constraint system used by libraries, offline bookstores, or even mobile phones so that both offline and online users can evaluate the quality of books in an automatic and objective way. Specifically, an offline bookseller can use a point-of-sale system with the proposed framework, so that he/she can automatically check the condition of (used) books before the sale. The online market users also can use a mobile phone to evaluate his/her (used) books to evaluate the value of them before registering the product. The first version of YOLO series was proposed in 2016 [4], and follow-up variants [5–9] have been presented recently. Among various YOLO models, this study employs several state-of-the-art versions, YOLOR [5], YOLOX [6], and YOLOv5 [7], to address our challenging topics and compare each model and method, thereby presenting design guidelines for improving the performance of book damage detection based on object detection.

The main contributions of this study can be classified into:

1. Generation of a new dataset for detecting book damages comprised of 6 categories (i.e., Wear, Spot, Notch, Barcode, Tag, and Ripped). Each class included in the dataset was designed to represent the characteristics of the common book damages.
2. Evaluation of the performance of the proposed book damage detection system based on various YOLO object detection methods. In particular, we implemented our framework with various data augmentation techniques, such as basic transformations (e.g., Flip, Rotate, Crop, etc.) and advanced methods (e.g., Mosaic [8], Mix-up [10], etc.). Through the experiments, we discussed the strength and limitations of the proposed system.

The rest of this paper is as follows: Section 2 reviews several studies related to defect detection in books and other domains. The overall framework and details are described in Section 3. Experimental results and analysis are presented in Section 4 and the conclusion and discussion are given in Section 5.

2 Related Work

2.1 Applications of Defect Detection

The research topic of this study is a specialized case of defect detection that has been studied actively in diverse industrial domains. Defect detection is a task to automatically recognize anomalous regions on the surfaces of products from various domains, such as steel, printed circuit board (PCB), and fabric industries. There exist various types of defects depending on the type of product. For example, scratches, dents, and black spots are common cases that can be found on the surface of the steel. However, most of the defects have inconsistent shapes and textures, complex patterns, and from tiny to large sizes, so that they cannot be detected well by manual human inspection. Therefore, various studies to effectively perform automatic defect detection without human expert knowledge have been proposed. In the steel manufacturing industry, automatic defect detection on the flat steel surface has received continuous attention to improve the quality of a product for a long time. Auto visual inspection (AVI) instruments were used as a standard configuration to validate the surface quality [11]. However, the traditional AVI system had limitations to be used in real-world steel mills in terms of accuracy, time efficiency, and robustness. Hence, several studies based on machine learning techniques were presented to overcome this problem. Zhao et al. [12] proposed a steel detection network that applied deformation convolutions to Fast Region-based Convolutional Neural Network (R-CNN) for the extraction of more abundant features. Li et al. [13] designed a steel surface detector based on the improved YOLO network and achieved a 99% detection rate with a speed of 83 frame per second (FPS). The defect detection method was also employed in the printed circuit board domain. The high complexity and miniaturization of PCBs have led to the development of AVI systems based on machine vision techniques [14]. Law et al. [15] proposed a defect classification system based on deep convolutional neural networks (CNNs) to decrease the false alarm rate of AVI systems. Kim et al. [16] also proposed an advanced PCB inspection system based on a skip-connected convolutional autoencoder to solve the challenge of learning new types of defects based on only a few of samples. Defect detection is also important to control the quality of fiber products. In particular, fiber products have much more complex patterns and textures and this characteristic resulted in more than 70 textile defect categories in the textile industry [17]. Li et al. [18] presented a compact CNN architecture for fabric defects classification by employing several microarchitectures. Jing et al. [19] proposed a Mobile-UNet architecture to achieve efficient end-to-end defect segmentation.

2.2 Book Recognition and Analysis

In recent years, many researchers have proposed various studies and applications based on computer vision and artificial intelligence (AI) techniques to support automatic library management, genre classification, and so on. For example, a library automation system based on book cover recognition using deep learning was proposed by Parikh et al. [20]. They employed a neural network model for text detection and recognition using their own optical character recognition and text-matching algorithms to reduce the manual labor of the human staff in library management. In addition, the proposed network was fine-tuned to deal with noisy backgrounds and distorted images. Yang et al. [21] also proposed a dep learning-based scene text detection and recognition system for identifying books in a bookshelf library. With a combination of CNN and bidirectional long short-term memory architectures, they established a book spine text recognition method and evaluated their retrieval performance. Zhou et al. [22] tried to address a library inventory management problem through segmentation and recognition tasks. In their work, book spine masks were extracted from the image of on-shelf books with deep learning-based segmentation models, and then the spine feature encoder was trained to learn the deep visual features of book spines. Afterward, those feature representations were used to recognize the target book identity. Also, several attempts to handle the classification of book genres using book covers have been made. Biradar et al. proposed an approach to classify book genres based on book cover and title by applying the logistic regression method [23]. Xia et al. also presented a CNN-based approach to handling the recognition and classification of the front and back covers of books [24]. Through various training and testing, they built a set of CNN models that can save human resources required for library management. A similar approach based on CNN was also proposed by [25] in which a set of book cover images and textual descriptions were downloaded and then refined to train the classification model. The authors of [25] compared the performance of a CNN-based image classification model and a textual description-based model for 14 different book genres. Furthermore, Rasheed et al. proposed a deep learning-based book genre recognition model which exploits multi-modality input data, achieving better performances compared to single-modality models [26].

Despite the recent development of various AI-based applications using book cover images, automatic book damage detection has not been actively studied. Therefore, this paper aims at addressing this problem by applying deep learning-based object detection techniques.

2.3 Deep Learning-Based Object Detection

Since the successful application of deep learning-based image classification approaches (e.g., CNNs), numerous studies based on deep learning technologies have been proposed to address various vision tasks including feature extraction, object detection, object tracking, semantic segmentation, and so on. Among these tasks, book damage detection, the topic of this study, is highly related to the object detection problem because such defects can be treated as objects that need to be recognized and classified. Recently, a variety of CNN-based object detection models have been proposed [27,28], and they can be generally classified into one-stage and two-stage detectors. Two-stage detectors attempt to find candidate regions and then classify the revision into object categories while one-stage detectors regard the object detection problem as a regression task to predict bounding boxes and class probabilities in a single evaluation. Representative works belonging to the two-stage detector include spatial pyramid pooling (SPP) [29], regions with CNN features (R-CNN) [30], Fast R-CNN [31], Faster R-CNN [32], feature pyramid networks (FPN) [33], and Mask R-CNN [34]. The one-stage detector also can be represented by the single-shot multi-box detector (SSD) [35], focal loss for object detection (RetinaNet) [36], and You Only Look Once (YOLO) variants. Generally, it was pointed out that the region proposal-based methods (i.e., two-stage detectors) are high in accuracy but low in speed; the regression-based methods (i.e., one-stage detectors) are high in speed but low in accuracy. However, with subsequent development and improvement in one-stage detectors, YOLO-based architectures (e.g., YOLOv5, YOLOR, and YOLOX) have provided competitive performance in terms of both accuracy and efficiency in recent years. In this regard, this paper utilizes these methods as our base models for book damage detection and then explores how each model can be optimized with various training configurations.

3 Method

First, we explain the training process of the proposed book damage detection system. As shown in Fig. 1, the proposed framework consists of i) image preprocessing and input, ii) training configuration with data augmentation, iii) training detection models, and iv) generating book damage detections. First, a set of book images collected and their corresponding annotations are pre-processed and passed to the model training component as input. In the training module, various kinds of training configurations are set based on the combination of basic and advanced augmentation techniques. With these configurations, object detection models based on YOLOv5, YOLOR, and YOLOX are trained. In this work, the recent YOLO-based models were employed as our base object detection network of the system. Finally, the category and location of book damages are detected through the trained models.

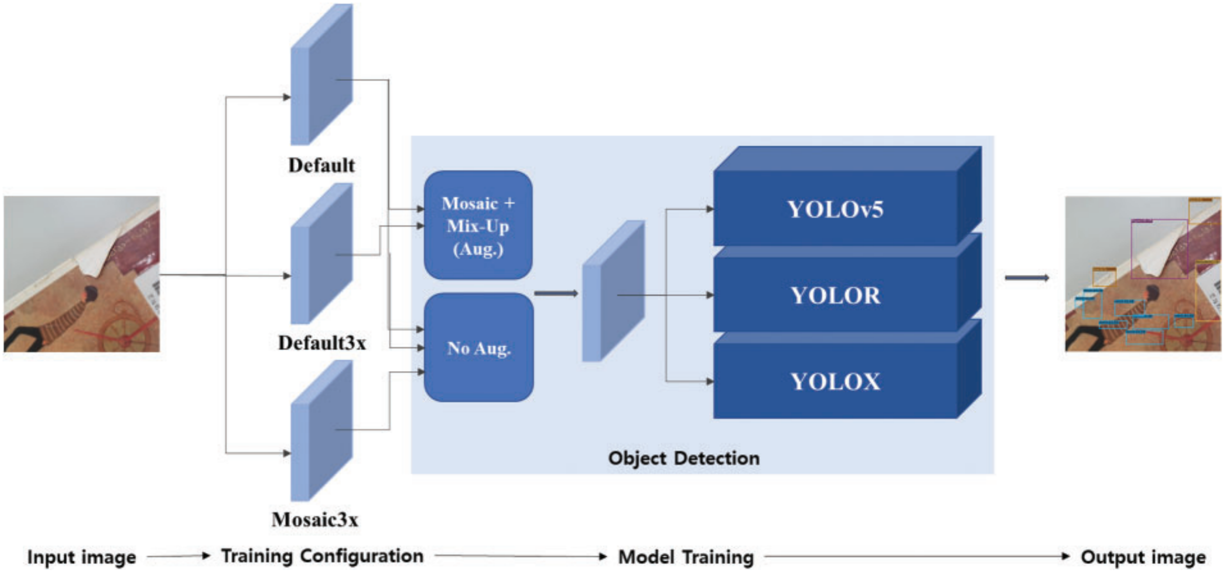


Figure 1: Overall workflow of model training in the proposed approach

3.1 Image Acquisition & Annotations

A set of books with arbitrary damages in literature, history, philosophy, and natural sciences were collected from the university library and took photos of them in various configurations to construct a dataset for train/valid/test to be used in this study. To build a more robust book damage detection model, we attempted to obtain as many book images with diverse visual appearances as possible. Fig. 2 shows examples of the collected images used in our study. For this, image samples of the books were taken with various views/angles, such as the front view (Fig. 2a), a zoom-in view of the part of a book (Fig. 2b), the side of a book (Fig. 2c), and a 45° angle view of a book (Fig. 2d). The image was taken under consistent daytime lighting conditions, and only images containing several book damages were included in our dataset. Finally, the collected dataset consists of a total of 620 images.

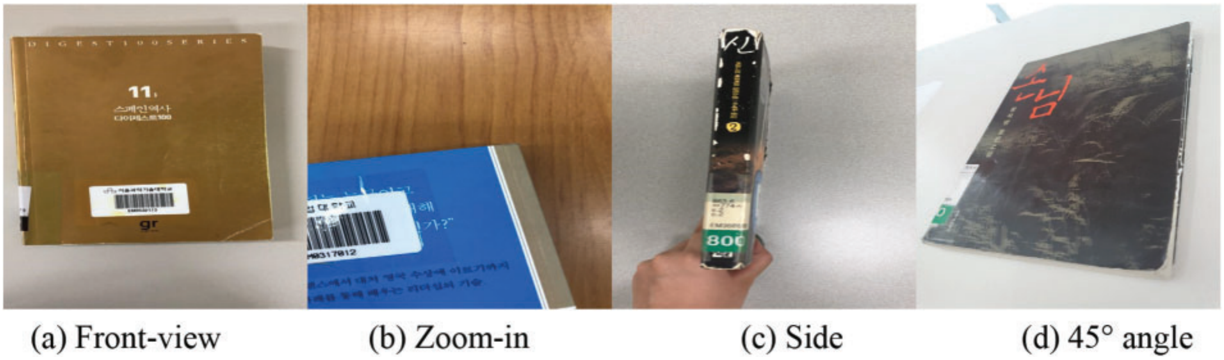
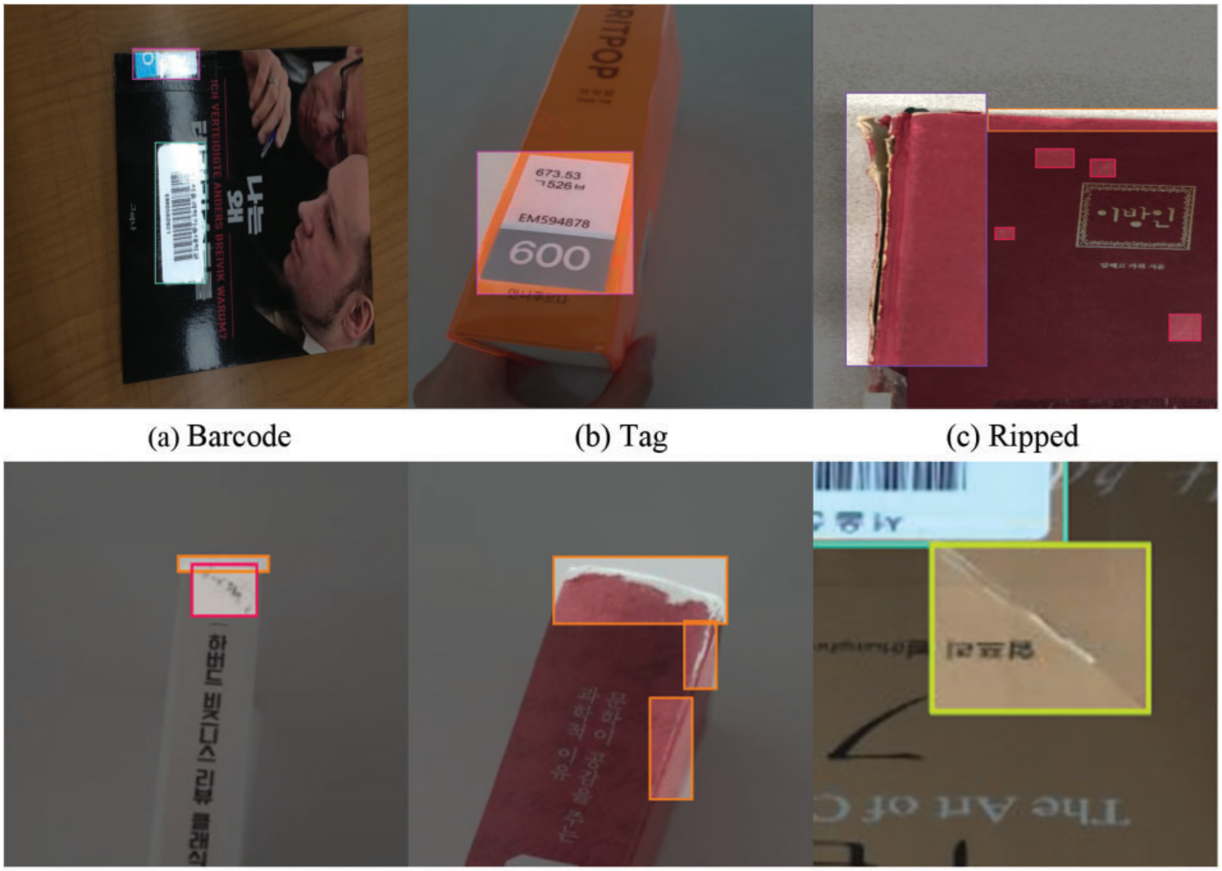

(a) Front-view (b) Zoom-in (c) Side (d) 45° angle

Figure 2: Example of view and angle for capturing book conditions

As shown in Fig. 3, the damage to the book can be categorized into 6 classes: *Barcode* (i.e., library label), *Tag* (i.e., library identification label), *Ripped*, *Spot*, *Wear*, and *Notch*. The *Barcodes* and *Tags* are usually attached to library books. They were chosen because most bookstores forbid selling books brought from libraries, the system therefore should be able to distinguish the books owned by the library from the used books owned by individuals. The *Ripped* class represents a case where a book cover is torn and cracked, so most of the *Ripped* damages are found on the edge of a book. The *Spot* class is a contaminated region with various colors and sizes, whose boundary is usually dim. The *Wear* class is a kind of damage that can be generally found in old books, in which the paper is worn out. Finally, the *Notch* is a folded or scratched mark on the cover. After the book damage class definition, we manually annotated each book image by using a web-based annotation tool provided by Roboflow [37] service that offers various computer vision-related functions, such as dataset management, annotation, and training. As a result of annotation, the dataset has 3,989 bounding boxes (i.e., Barcode: 1,339, Tag: 937, Ripped: 411, Spot: 729, Wear: 379, and Notch: 194). The book damage annotation was done after resizing all images to 640 $*$ 640 scale.



(a) Barcode (b) Tag (c) Ripped

(d) Spot (e) Wear (f) Notch

Figure 3: List of book damage categories and example images in our dataset

3.2 Object Detection Models

To utilize an automatic book damage detection system in bookstores and libraries, a fast, lightweight, and accurate object detection model that can be embedded in portable or mobile devices needs to be considered. There exist popular object detection models based on deep learning, such as R-CNN, and Faster R-CNN, but their processing speed was slow because they perform a classification task after the region proposal is done [38]. In contrast, a single-stage detector performs these two processes at the same time, thereby drastically improving the processing speed while minimizing performance loss. The first version of well-known single-stage detectors called YOLO was proposed in 2016 [4], followed by numerous variants which have proven themselves suitable for real-time object detection tasks. Therefore, this paper presents a book damage detection system using YOLO-based models. Among various versions of YOLO methods, we exploited YOLOv5, YOLOR, and YOLOX models which have shown impressive performances recently.

YOLOv5 consists of three main components: backbone, neck, and head output as depicted in Fig. 4. The backbone extracts feature maps from an input image. The neck connects the backbone and head, refining and reconfiguring the feature map. The head is a module in which classification and regression of detections are made. As shown in Fig. 4, the cross-stage partial network (CSPNet) [39] with BottleneckCSP and SPP [29] is used as a backbone of YOLOv5. The CSPNet enables the architecture to extract rich information from the feature map while reducing the amount of computation by integrating feature maps from the beginning and the end of a network stage [39]. Each module in the backbone and neck has a bottleneck structure that can reduce the parameters by decreasing the channel of input data. The SPP structure can generate a fixed-size representation regardless of the image resolution through multi-scale pooling operations, so the models do not need to require a fixed-size input image. In YOLOv5, an SPP module is placed at the end of the backbone, contributing to the improved performance of the model, as shown in Fig. 4. Also, YOLOv5 employs a top-down pyramid network called Path Aggregation Network (PANet) [40] based on FPN [33] as a neck. The FPN layer conveys powerful semantic features from top to down and the feature pyramid passes powerful positioning features from bottom to top. By incorporating the features from various layers, YOLOv5 can accurately recognize the target object of multiple sizes and scales.

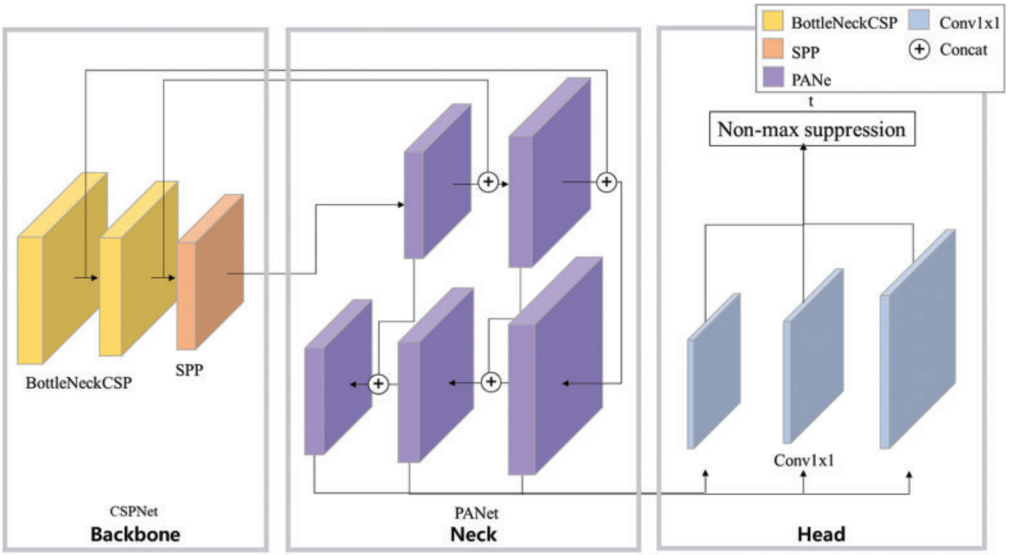


Figure 4: YOLOv5 architecture

YOLOR is a model that can perform multiple tasks using a single unified model. This work collects explicit knowledge and implicit knowledge from one representation. Fig. 5 shows the architecture of the unified network introduced in YOLOR. According to [5], explicit knowledge can refer to visual features of an image that could be acquired from shallow layers of a neural network. On the other hand, implicit knowledge can be defined as knowledge learned in a subconscious state, which corresponds to the features obtained from deeper layers of neural networks. The YOLOR method extracts and integrates both explicit and implicit information to learn a general representation for further multiple kinds of tasks. The authors of [5] argued that they achieved object detection results that are sufficient to match the state-of-the-art approaches by introducing a unified network for learning implicit knowledge together.

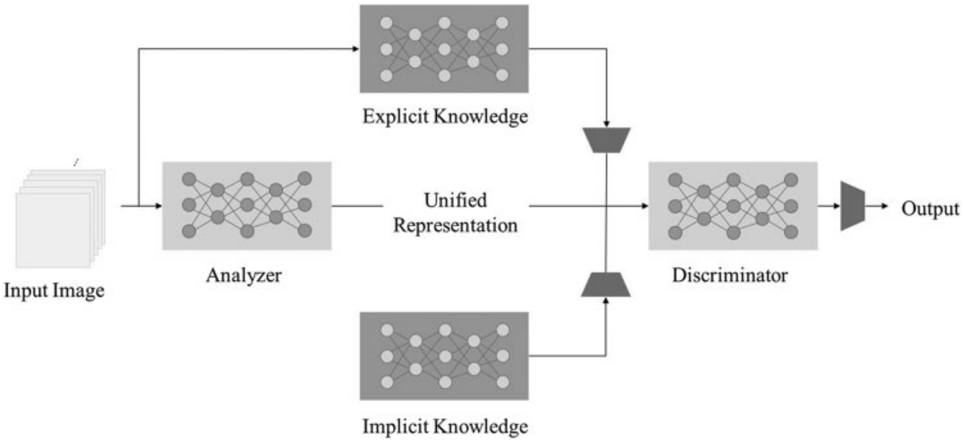


Figure 5: YOLOR architecture

YOLOX is an extended version of the YOLOv3-SPP [41] baseline which adopted various recent advanced detection techniques (e.g., anchor-free manner, decoupled heads, strong augmentation, etc.). In an architecture based on a coupled head like YOLOv5, the conflict between classification and regression tasks occurs because classification and regression are handled in the same head even though they have different characteristics. As depicted in Fig. 6, YOLOX adopted the decoupled head structure which has two separate branches for classification and localization. Also, YOLOX is an anchor-free detector in contrast to anchor-based detectors [7,8] such as YOLOv4 and YOLOv5, which must determine a set of optimal anchors before training in a heuristic way. This method was known to be less generalizable and to cause the increased complexity of the detection head. Recent anchor-free detectors [42–44] have shown a similar level of performance compared to the anchor-based detector with a simple training process. Through various experiments, the authors showed that YOLOX achieves a better trade-off between efficiency and accuracy than other detection models for all model sizes.

3.3 Implementation

3.3.1 Data Augmentation

Data augmentation is widely used to increase the number of training samples as well as the diversity of visual appearances in training data, thereby preventing overfitting problems. Simple affine transform-based augmentation techniques have been considered one of the default training strategies in the deep learning domain. In recent years, more advanced augmentation techniques based on sample mixing mechanisms, such as Mix-up and CutMix, have been proposed to improve the performance of CNN for image classification and object detection. Among various advanced augmentation techniques, this work applied Mosaic [8] & Mix-up [10] as our main augmentation techniques for the object detection models. Mosaic is a well-known effective augmentation technique in the computer vision domain, which results in a single image where four original images with different scales are integrated. Mix-up is another popular advanced augmentation technique that convex-combinates two raw images into a single soft-label training sample. In this work, each object detection model was trained with five different model training configurations shown in Table 1. Each training configuration employs a different data augmentation strategy. The "Default" configuration is the default training strategy that utilizes the original dataset without any advanced augmentations applied. The "Default + Aug" refers to the "Default" configuration with advanced augmentation methods applied. Conversely, the "Default3x" training configuration utilizes 3x more training samples added by basic augmentation methods (i.e., Flip, Rotation, and Crop), but without advanced augmentation methods applied. Similarly, the "Default3x + Aug" refers to the "Default3x" configuration with advanced augmentation methods applied. Finally, the "Mosaic3x" configuration utilizes 3x more training samples added by the Mosaic augmentation method. Based on these configurations, we train YOLOv5, YOLOR, and YOLOX frameworks and compare their performance in detecting book damages.

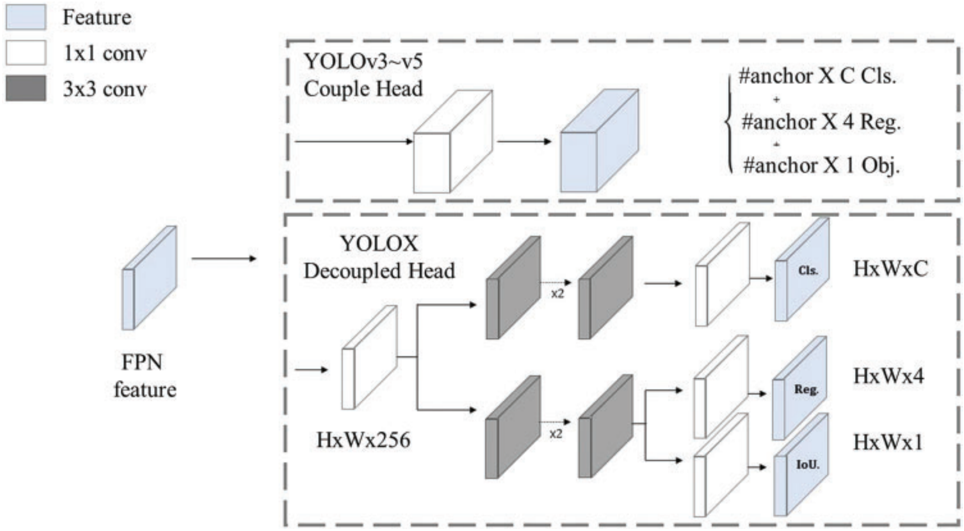


Figure 6: YOLOX architecture

Table 1: List of training configurations and augmentation techniques

Training configuration	Augmentation strategy
Default	N/A
Default + Aug.	Mosaic + Mixup
Default3x	Basic
Default3x + Aug.	Basic + Mosaic + Mixup
Mosaic3x	Mosaic

3.3.2 *Training Details*

All experiments were performed on a server equipped with a single NVIDIA GeForce RTX 2080 Ti graphic processing unit (GPU), 128 GB random access memory (RAM), and Intel i9-7920X central processing unit (CPU). Training and validation of YOLO models were performed through implementations based on the Pytorch framework. All the models, YOLOR, YOLOX, and YOLOv5 used stochastic gradient descent (SGD) optimizer and cosine learning rate scheduler and were trained for 200 epochs with a batch size of 8. The YOLOR and YOLOv5 models were trained with 3 warmup epochs and YOLOX adopted 1 warmup epoch. The parameter to control the probability of Mosaic and Mix-up augmentation was both set to 1.0. The dataset was divided into train:valid:test set with a ratio of 70%:20%:10%, which contains 434, 124, and 62 images with 2,800, 788, and 401 bounding boxes with annotations, respectively. The number of training images for Default3x, Default3x + Aug, and Mosaic3x training configurations was increased to 1,302 since additional images were included by the augmentation technique.

4 Experiments and Results

This section discusses the performance of each model in the detection of book damages both in quantitative and qualitative ways. In the quantitative analysis, the performance of each model in terms of the metric of object detection tasks called mean average precision (mAP) is reported. For the qualitative analysis, we provide various examples of book damages detected by each model, which show the characteristics and differences between each configuration.

4.1 Quantitative Analysis

4.1.1 *Overall Performance*

Table 2 summarizes the performance of all models trained with different training configurations. First, we report the results of our experiments in terms of mAP with Intersection over Union (IoU) threshold of 0.5 (notated as mAP@.5 hereafter) and averaged mAP with IoU threshold ranging from 0.5, 0.55, 0.6, … to 0.95 (notated as mAP@.5:.95, hereafter). In the case of mAP with IoU threshold 0.5, the detection result from each model is considered true positive if the IOU between the predicted detections and the ground-truth detections for the book damages is equal to or higher than 0.5. That is, the predicted detections are deemed correct if they have at least half overlap with the ground truth damages. Conversely, averaged mAP with different IoU thresholds from 0.5 to 0.95 more rigorously validate how well each model detects and localizes the damages.

Table 2: Summary of quantitative performance in book damage detection (unit: 'M' indicates million, 'h' represents hour)

Models	Train configuration	mAP@.5 (%)	mAP@.5:.95 (%)	Parameters (M)	GFLOPS	Training time (h)
YOLOv5	Default	36.5	24.9	46.2 M	108.3	0.7
	Default + Aug.	37.2	25.9			0.7
	Default3x	36.9	26.2			1.5
	Default3x + Aug.	39.5	26.7			1.4
	Mosaic3x	**75.9**	**57.5**			1.6
	Default	35.4	24.3			0.8
	Default + Aug.	36.1	25.8			0.8

(Continued)

Table 2 (continued)

Models	Train configuration	mAP@.5 (%)	mAP@.5:.95 (%)	Parameters (M)	GFLOPS	Training time (h)
YOLOR	Default3x	35.6	26.0	52.5 M	119.8	1.9
	Default3x + Aug.	37.9	26.9			1.9
	Mosaic3x	**72.3**	**51.2**			2.0
YOLOX	Default	69.2	52.5	54.2 M	155.7	1.3
	Default + Aug.	70.3	53.3			1.3
	Default3x	39.1	28.8			3.5
	Default3x + Aug.	39	29			3.3
	Mosaic3x	**72.9**	**60.0**			3.3

First, all of the models could achieve the best performance in terms of mAP@.5 with Mosaic3x configuration. Specifically, YOLOv5, YOLOR, and YOLOX resulted in an mAP of 75.9%, 72.3%, and 72.9%, respectively. Similarly, in terms of mAP@.5:.95, Mosaic3x training configuration was most effective for both YOLOv5, YOLOR, and YOLOX methods, resulting in 57.5%, 51.2%, and 60.0%, respectively. Among the tested models, YOLOv5 did achieve the best performance in terms of mAP@.5, while YOLOX performed best in terms of mAP@.5:.95. In both metrics, the YOLOR model yielded the worst performance. These results indicate that YOLOv5 could produce better results compared to YOLOR and YOLOX when a low threshold of true positive is applied (i.e., mAP@.5), however, YOLOX worked the best in the more rigorous test (i.e., mAP@.5:.95). Considering the complexity of each model shown in the 5/6/7th columns in Table 2, YOLOv5 with Mosaic3x configuration would be the best option for book damage detection in both metrics. More details about the model complexity in terms of training efficiency will be discussed in Section 4.1.4. In addition, we also evaluated the performance of Deformable Detection Transformer (DETR) method [45], which is one of state-of-the-art object detection models, to compare YOLO-based models with transformer-based ones. The Deformable DETR method was trained and validated on the same dataset. Of various training configurations, we chose Default and Mosaic3x configurations for comparison. As a result, the DETR/Default and DETR/Mosaic3x models yielded mAP@.5:95 of 48.2 and 25.4, respectively. The comparative result validates that YOLO-based models trained with appropriate configurations can outperform transformer-based models for handling a specific domain problem, which also implies the feasibility of the proposed method.

Second, applying basic augmentation methods to increase the amount of training data was helpful for YOLOv5 and YOLOR, resulting in a slight performance gain. Specifically, compared to Default configuration, Default3x method led to an average performance improvement of 1.5%p (mAP@.5:.95.) and 0.3%p (mAP@.5). Similarly, the performance of both models trained with Default3x + Aug. configuration increased by 1%p (mAP@.5:.95) and 2.1%p (mAP@.5) on average, compared to the models trained with Default + Aug. configuration. However, we could observe a different pattern in YOLOX with respect to the effects of the use of basic augmentation techniques. For example, compared to the Default baseline, the performance of YOLOX/Defalut3x rather decreased by 23.7%p (mAP@.5:.95) and 30.1%p (mAP@.5), respectively. Similarly, YOLOX showed a drastic performance drop of 24.3%p (mAP@.5:.95) and 31.3%p (mAP@.5) when trained with Default3x + Aug instead of Default + Aug. These results show that YOLOX tends to not benefit from increasing the number of training samples if the samples do not have diverse visual appearances.

Third, the effect of the application of advanced augmentation methods was similar to that of basic augmentation methods. Specifically, YOLOv5 and YOLOR could benefit from Mosaic and Mixup augmentation, yielding the average performance improvement of 1.58%p (mAP@.5:.95) and 0.98%p (mAP@.5) However, there was only a slight difference in the performance of YOLOX when advanced augmentation was applied, resulting in an average performance gain of 0.5%p in both metrics. In summary, YOLOv5 and YOLOR could benefit most from Default3x configuration, in which both basic and advanced augmentation methods are applied, with average performance improvement of 2.75%p (mAP@.5) and 2.2%p (mAP@.5:.95) while only advanced augmentation method was useful for YOLOX. Nevertheless, as shown in Table 2, the best performance was observed when only Mosaic3x augmentation is used, yielding a significant performance difference compared to other configurations.

Finally, YOLOR and YOLOv5 models showed a similar trend in performance across all training configurations. In particular, it was observed that the models trained with Mosaic3x achieved improved performance which significantly outperformed the models trained with other configurations. However, YOLOX showed an entirely different pattern. For example, YOLOX models trained with Default configurations (i.e., YOLOX/Default, YOLOX/Default + Aug.) performed much better than YOLOR/YOLOv5 Default configurations and even comparable to YOLOX/Mosaic3x.

4.1.2 Effect of Mosaic Augmentation

Next, as described in Table 2, the extensive use of Mosaic augmentation (i.e., Mosaic3x) significantly improved the model performance; therefore, we further explore if the Mosaic augmentation is also useful in other training configurations. Setting Default3x as a baseline, the performance of other models trained with Mosaic augmentation is compared to figure out how Mosaic augmentation affects the overall performance. For this, the models trained with Mosaic3x, Default3x + Aug (i.e., Default3x + Mosaic + MixUp) were included in the analysis. In addition, we also trained YOLO models using Default3x + Mosaic-only configuration and denoted them as YOLO[v5,X,R]/Default3x + Mosaic.

Table 3 describes the performance of the models trained with Mosaic3x, Default3x + Aug., Default3x + Mosaic, and Default3x. As shown in Table 3, the effect of Mosaic augmentation is different according to the YOLO model used. When applying Mosaic augmentation only to the baseline (i.e., Default3x + Mosaic), YOLOv5 gained the highest performance improvement by 2.2%p (mAP@.5) and 1.3%p (mAP@.5:.95), respectively. Both YOLOR and YOLOX also benefitted from the use of Mosaic augmentation only, achieving an average performance gain of 1.5 (mAP@.5) and 0.5%p (mAP@.5:.95). Conversely when MixUp augmentation is added (i.e., Default3x + Mosaic + Mixup), only YOLOv5 and YOLOR could have benefits. Specifically, YOLOv5 had an additional performance improvement of 0.4%p in terms of mAP@.5, whereas −0.8%p in mAP@.5:.95. The YOLOR model showed better results, yielding 37.9% (mAP@.5) and 26.9% (mAP@.5:.95). On the other hand, adding MixUp adversely affect the performance of YOLOX, resulting no significant difference compared to that of the baseline (Default3x). Finally, as discussed above, all the models achieved drastic performance improvement when they were trained with Mosaic3x configuration. In summary, the advantage of Mosaic augmentation could be validated through various experiments. Furthermore, the experimental results showed that Mosaic3x is more effective to organize the dataset to have more visual diversities.

Table 3: Impact of Mosaic augmentation on the overall performance

Models	Methods	mAP@.5 (%)	mAP@.5:.95 (%)
YOLOv5	Mosaic3x	75.9	57.5
	Default3x + Mosaic + MixUp	39.5	26.7
	Default3x + Mosaic	39.1	27.5
	Default3x	36.9	26.2
YOLOR	Mosaic3x	72.3	51.2
	Default3x + Mosaic + MixUp	37.9	26.9
	Default3x + Mosaic	37.1	26.5
	Default3x	35.6	26.0
YOLOX	Mosaic3x	72.9	60.0
	Default3x + Mosaic + MixUp	39.0	29.0
	Default3x + Mosaic	40.6	29.4
	Default3x	39.1	28.8

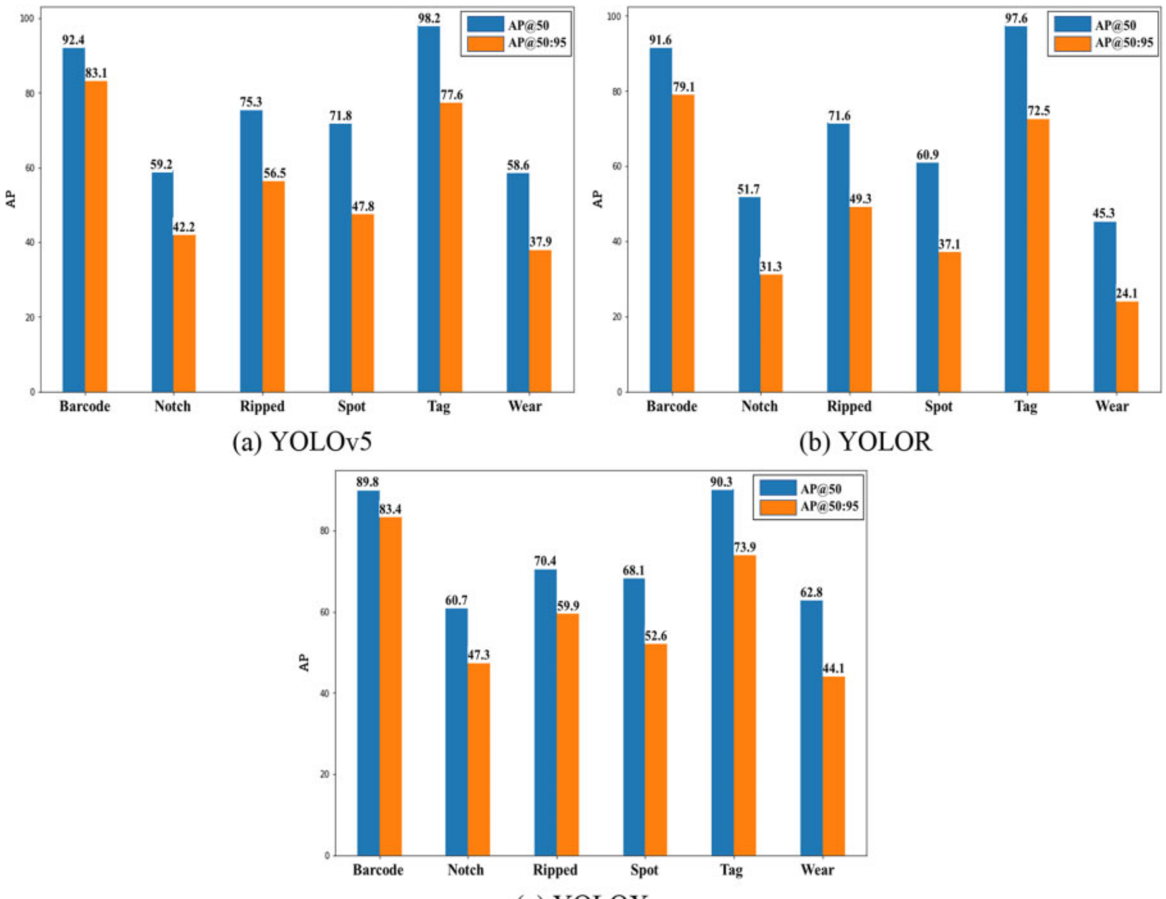
4.1.3 Class-Wise Performance

Next, as mentioned in Section 3.2, six book damage classes that have different characteristics and granularities were defined in this study. Therefore, it is also worth inspecting the AP of each class to figure out how each YOLO-based method works in detecting book damage. As shown in the experimental results described in Section 4.1.1, since the models trained with Mosaic3x configuration generally showed outstanding performance, we report and analyze their class-wise APs. Fig. 7 depicts a summary of APs for each class from each YOLO model. First, the damage classes that have large sizes and distinctive shapes, such as Barcode and Tag, were detected well by all models with an average performance of 93.3 (mAP@.5) and 78.3 (mAP@.5:.95). Second, Notch, Spot, and Wear damages are generally small-sized and have more ambiguous appearances/shapes, thereby resulting in much lower APs from all the models. For example, the APs of Notch are 42.2% (YOLOv5), 31.3% (YOLOR), and 47.3% (YOLOX) in terms of AP@.5:.95, which are much worse than those for Barcode and Tag classes. In particular, YOLOR showed much worse performance for small-size damage classes, yielding −10%p and −15%p lower than YOLOv5 and YOLOX in terms of mAP@.5 and mAP@.5:.95, respectively. Finally, the Ripped class was relatively well detected by every model, with mAP@.5 of 72.4% on average across the models. To summarize, large-size book damages tend to be detected well by all the models while the detection of small-sized ones shows limitation that needs to be addressed in our future study. The detection performance for a damage class that has a distinct visual appearance, like Ripped, was better than that for other small-sized damages.

4.1.4 Training Efficiency

Finally, we report the relationship between the model training time and the detection accuracy (mAP@.5:.95), through Fig. 8. Generally, the training time required for each model depends on the architecture and its training configuration. First, YOLOX tends to take more training time than other YOLO-based networks in every training configuration. However, the YOLOX-based model showed better detection outcomes when compared to other models. Second, when considering the number of parameters, giga floating operation per second (GFLOPS), and model training time reported in

Table 2, it could be observed that the model training time is largely affected by GFLOPS rather than the number of parameters. For example, YOLOv5 and YOLOR spent a similar amount of time training for the same training configuration, even though the difference in the number of parameters between them is quite large. More specifically, in the case of Default configuration, both YOLOv5 and YOLOR took 0.7∼0.8 h to finish the model training. The GFLOPS of YOLOv5 and YOLOR was reported as 108.3 and 119.8, respectively, which implies that both models have a similar level of model complexity. This trend becomes more obvious when YOLOR and YOLOX models are compared. As described in Table 2, even though YOLOR and YOLOX have the same level of the number of parameters (i.e., 52.5 and 54.2 M, respectively), the training time required for YOLOX models was much longer than that for YOLOR models. Similarly, the GFLOPS of YOLOX (119.8) was much higher than that of YOLOR (155.7), which implies that model complexity in terms of GFLOPS significantly affected the training efficiency. Third, the models trained with 3x-series configurations (i.e., Defatult3x, Default3x + Aug., and Mosaic3x) required more time to train than the ones with Default configurations. However, this is natural because 3x-series training configurations utilize 3x more training samples. Finally, the training time and detection accuracy trade-off could be found in every training configuration. For example, YOLOR and YOLOX generally took more time for training than YOLOv5 but produced better detection results.



(a) YOLOv5

(b) YOLOR

(c) YOLOX

Figure 7: Average Precision (AP) for each class from each YOLO-based detection architecture

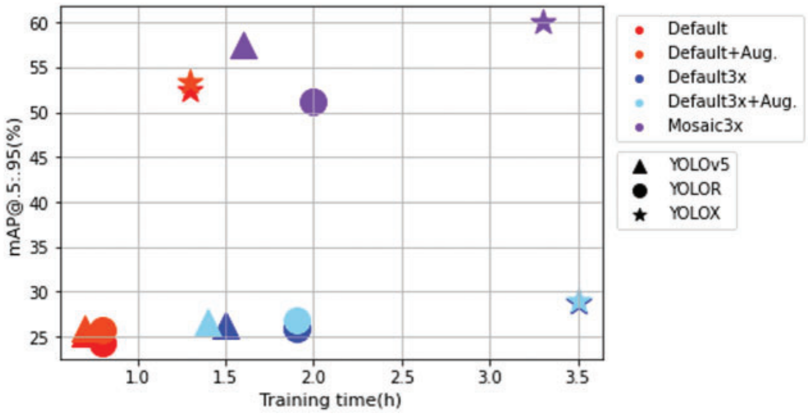


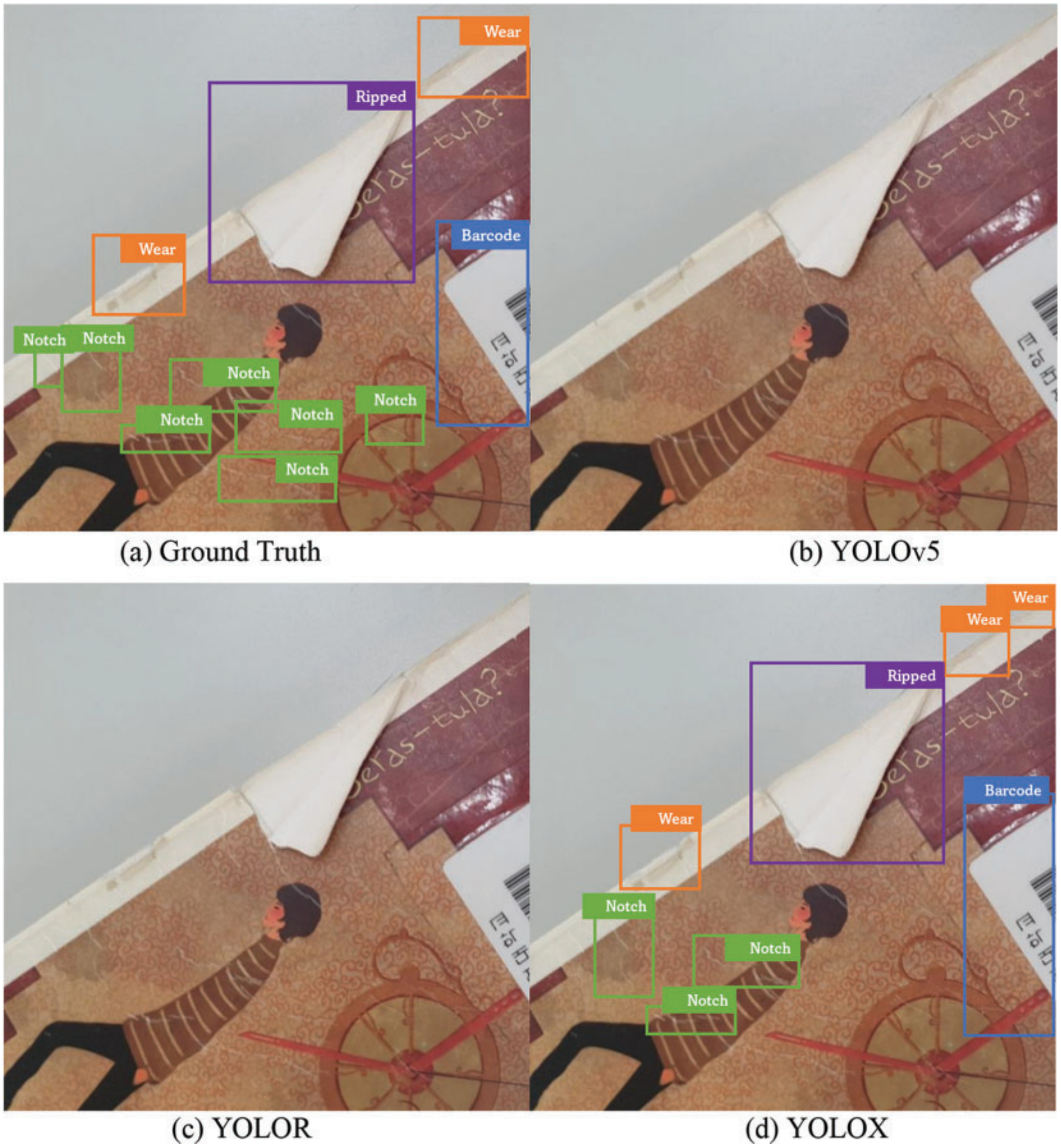
Figure 8: Relationship between training time (h) and detection accuracy (mAP)

Therefore, it could be concluded that if a domain problem is more sensitive to the detection performance (e.g., requiring high mAP with IoU threshold 0.9) choosing YOLOX as a detector would be a proper option. However, if a training environment is highly resource-constraint or a problem domain does not require fine-grained detection results, YOLOv5 and YOLOR models also would be eligible to meet the requirement.

4.2 Qualitative Analysis

In this section, the difference in performance between each model through various illustrative examples is presented. First, Fig. 9 shows examples of book damage detection results from YOLO models trained with the "Default" configuration. As shown in Fig. 9, YOLOX succeeded in detecting some of the book damages (Fig. 9d) while YOLOv5 and YOLOR failed to detect damages (Figs. 9b and 9c). This result is consistent with the quantitative analysis described in Table 2 which represented that YOLOX worked much better than YOLOv5 and YOLOR in case the models were trained with the Default configuration. Only YOLOX/Default model partly detected book damages like Ripped, Barcode, Wear, and some Notch damages. This illustrative example also implies that the models trained with Default configuration (i.e., without any augmentation technique) do not provide sufficient performance for practical use.

Next, we discuss the detection results from each YOLO model trained with a configuration that achieved the best performance. As depicted in Fig. 10, all the models illustrated similar book damage detection capabilities. Compared to the ground truth (Fig. 10a), most of the book damages were detected correctly by YOLOv5 (Fig. 10b) and YOLOX (Fig. 10d) models. The YOLOR model failed to detect some of Notch, Spot, and Wear classes (Fig. 10c). It is worth noting that all YOLO-based models even detected the damage that was not found and labeled by human annotators. For example, as shown in Fig. 10a, even though the Wear type damage on the left edge of a book is not labeled in the ground truth, all the YOLO models detected and classified it correctly (Figs. 10b–10d), revealing an outstanding performance of the trained models. The result of qualitative analysis is also similar to the pattern found from the quantitative evaluation of YOLOv5, YOLOR, and YOLOX shown in Table 2, indicating that YOLOv5 and YOLOX models worked better for book damage detection applications.



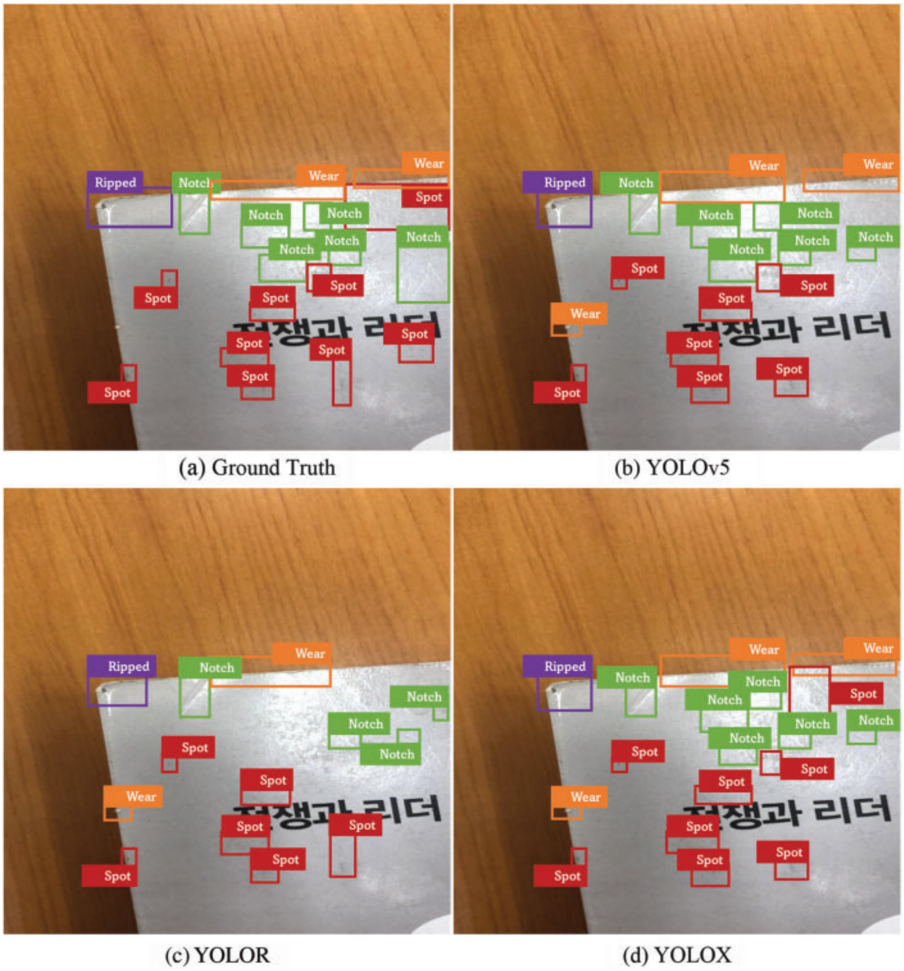
(a) Ground Truth

(b) YOLOv5

(c) YOLOR

(d) YOLOX

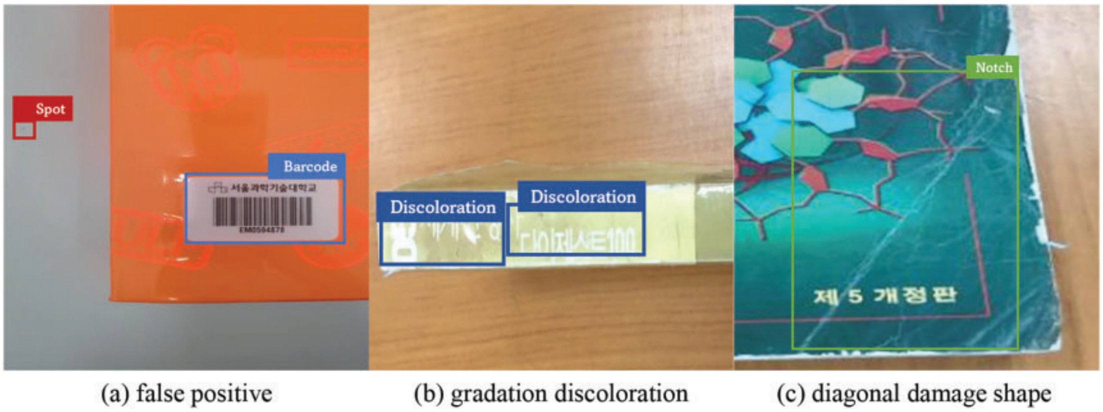
Figure 9: Detection result from each model trained with "Default" training configuration



(a) Ground Truth (b) YOLOv5

(c) YOLOR (d) YOLOX

Figure 10: Detection results from each model trained with the best training configuration (Mosaic3x)



(a) false positive (b) gradation discoloration (c) diagonal damage shape

Figure 11: Example of limitations in book damage detection

5 Conclusion

This paper proposed a technique for detecting damages in the used book using deep learning-based object detection algorithms. Also, this study presented the results of a comparative study for the book damage detection model by assessing and analyzing the performance shown by YOLO series models. To this end, we collected a set of images of used books and manually annotated them to construct a new dataset, and performed several experiments to evaluate the performance of YOLO-based methods with various model training configurations. The experimental results showed that the Mosaic3x training strategy was the most effective one for all YOLO models. Conversely, Default + Aug. and Default training strategy only worked well for YOLOX model. The training configurations to merely increase the number of training samples (i.e., Default3x + Aug. and Default3x) were not useful to improve the performance. Among the book damage classes, Barcode and Tag classes were detected well by all the models because of their relatively larger and simpler appearances. The Wear and Notch categories were the most difficult ones to detect due to their ambiguous patterns. Finally, through various illustrative examples, we showed the feasibility of YOLO-based models to recognize various book damages.

However, there are still several limitations that need to be addressed in our future work. First, it was found that there exist various detection failure cases. As shown in Fig. 11a, the model could produce a set of false positives (e.g., detecting a background floor as a spot). This problem was caused by detecting damage to the background rather than the book itself, which inspires us to combine book detection and damage detection steps in the future. To address this issue, we plan to integrate methodologies to recognize a book cover and book spine [21,22] into our system to improve detection performance. Second, the detection models could not successfully detect some damage categories, such as small-size damages (e.g., spot, notch, and wear) and gradation discoloration (Fig. 11b), due to the lack of enough training data and complex book cover design. Even though various augmentation strategies are applied in this study, the benefit from it was limited, resulting in an overall performance of lower than 0.8 and 0.6 in terms of mAP@.5 and mA@.5:.95. Therefore, to overcome this point, more advanced augmentation techniques or effective training approach will be utilized. For example, a generative approach [46] can be used to generate realistic images of damage classes that have fewer training samples or shows relatively lower classification accuracy. In addition, data augmentation methods [47,48] designed for performance improvement towards small-size objects will be considered to make the proposed system more robust. Weakly supervised object detection and localization approaches [49] can be considered as well to utilize more data for training the model even though the samples do not have complete annotations. Third, as depicted in Fig. 11c, if the shape of the damage has diagonal or complex patterns, the area of the generated bounding box could be much larger than the actual size of the damage. This inconsistency makes it hard to define the size of damage used to quantitatively evaluate the condition of a book. We plan to solve this limitation by adopting various segmentation techniques in the future. In particular, vision transformer-based approaches have been proposed recently to address segmentation tasks in various domains [50]. For example, [51,52] have shown that vision transformer-based architecture which can learn multi-scale, multi-level features yield state-of-the-art performances on various segmentation benchmarks. Through a future study, it will be attempted to apply these approaches to the book damage recognition domain for achieving performance improvement. Finally, for more practical applications and use cases, deep learning models for book damage detection and recognition need to be optimized in terms of computational resources while preserving performance. There have been various studies focusing on optimizing model architectures as well as hardware implementation for resource-constrained environments [53,54]. The future version of our work can be integrated with these approaches to meet various requirements from real-world conditions.

Funding Statement: This study was financially supported by Seoul National University of Science and Technology.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Persistence Market Research. Second Hand Books Market, 2022. [Online]. Available: https://www.persistencemarketresearch.com/market-research/second-hand-books-market.asp

[2] BBC News. The Booming Trade in Second-Hand Books, 2018. [Online]. Available: https://www.bbc.com/news/business-46386557

[3] Amazon, Condition Guidelines: Books, 2023. [Online]. Available: https://sellercentral.amazon.sg/help/hub/reference/external/GZBC9BLXDJVBZPAR

[4] J. Redmon, S. K. Divvala, R. B. Girshick and A. Farhad, "You only look once: Unified, real-time object detection," in *Proc. IEEE/CVF CVPR*, Las Vegas, USA, pp. 779–788, 2016.

[5] C. Y. Wang, I. H. Yeh and H. Y. M. Liao, "You only learn one representation: Unified network for multiple tasks.", 2021. [Online]. Available: https://arxiv.org/abs/2105.04206

[6] Z. Ge, S. Liu, F. Wang, Z. Li and J. Sun, "Yolox: Exceeding yolo series in 2021." 2021. [Online]. Available: https://arxiv.org/abs/2107.08430

[7] Ultralytics. YOLOv5, 2023. [Online]. Available: https://github.com/ultralytics/yolov5

[8] A. Bochkovskiy, C. Y. Wang and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection." 2020. [Online]. Available: https://arxiv.org/abs/2004.10934

[9] C. Y. Wang, A. Bochkovskiy and H. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022. [Online]. Available: https://arxiv.org/abs/2207.02696

[10] H. Zhang, M. Cisse, Y. N. Dauphin and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," in *Proc. ICLR*, Vancouver, Canada, pp. 1–13, 2018.

[11] Q. Luo, X. Fang, L. Liu, C. Yang and Y. Sun, "Automated visual defect detection for flat steel surface: A survey," *IEEE Transactions on Instrumentation and Measurement*, vol. 69, no. 3, pp. 626–644, 2020.

[12] W. Zhao, F. Chen, H. Huang, D. Li and W. Cheng, "A new steel defect detection algorithm based on deep learning," *Computational Intelligence and Neuroscience*, vol. 2021, no. 10, pp. 10–13, 2021.

[13] J. Li, Z. Su, J. Geng and Y. Yin, "Real-time detection of steel strip surface defects based on improved YOLO detection network," *IFAC-PapersOnLine*, vol. 51, no. 21, pp. 76–81, 2018.

[14] X. Zheng, S. Zheng, Y. Kong and J. Chen, "Recent advances in surface defect inspection of industrial products using deep learning techniques," *International Journal of Advanced Manufacturing Technology*, vol. 113, no. 6, pp. 35–58, 2021.

[15] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proc. ECCV*, Munich, Germany, pp. 734–750, 2018.

[16] J. Kim, J. Ko, H. Choi and H. Kim, "Printed circuit board defect detection using deep learning via a skip-connected convolutional autoencoder," *Sensors*, vol. 21, no. 15, pp. 4968, 2021.

[17] S. Mei, Y. Wang and G. Wen, "Automatic fabric defect detection with a multi-scale convolutional denoising autoencoder network model," *Sensors*, vol. 18, no. 4, pp. 1064, 2018.

[18] Y. Li, D. Zhang and D. Lee, "Automatic fabric defect detection with a wide-and-compact network," *Neurocomputing*, vol. 329, pp. 329–338, 2019.

[19] J. Jing, Z. Wang, M., Rätsch and H. Zhang, "Mobile-unet: An efficient convolutional neural network for fabric defect detection," *Textile Research Journal*, vol. 92, no. 1–2, pp. 30–42, 2020.

[20] J. V. Parikh, A. Natarajan and B. S. Babu, "Library automation system: Book cover recognition using deep learning," in *Proc. CSITSS*, Bangalore, India, pp. 1–5, 2019.

[21] X. Yang, D. He, W. Huang, A. Ororbia, Z. Zhou *et al.,* "Smart library: Identifying books on library shelves using supervised deep learning for scene text reading," in *Proc. JDCL*, Toronto, Canada, pp. 1–4, 2017.

[22] S. Zhou, T. Sun, X. Xia, N. Zhang, B. Huang *et al.,* "Library on-shelf book segmentation and recognition based on deep visual features," *Information Processing & Management*, vol. 59, no. 6, pp. 1–17, 2022.

[23] G. R. Biradar, R. JM, A. Varier and M. Sudhir, "Classification of book genres using book cover and title," in *Proc. IEEE ICISGT*, Gramado, Brazil, pp. 72–723, 2019.

[24] H. Xia, Y. Qi, Q. Zeng, Y. Li and F. You, "CNN-based book cover and back cover recognition and classification," in *Proc. TridentCom*, Melbourne, Australia, pp. 59–70, 2021.

[25] P. Buczkowski, A. Sobkowicz and M. Kozlowski, "Deep learning approaches towards book covers classification," in *Proc. ICPRAM*, Funchal, Portugal, pp. 309–316, 2018.

[26] A. Rasheed, A. I. Umar, S. H. Shirazi, Z. Khan and M. Shahzad, "Cover-based multiple book genre recognition using an improved multimodal network," *International Journal on Document Analysis and Recognition*, vol. 26, no. 1, pp. 65–88, 2023.

[27] Z. Ren, F. Fang, N. Yan and Y. Wu, "State of the art in defect detection based on machine vision," *International Journal of Precision Engineering and Manufacturing-Green Technology*, vol. 9, no. 2, pp. 661–691, 2022.

[28] A. A. Tulbure, A. A. Tulbure and E. H. Dulf, "A review on modern defect detection models using DCNNs-deep convolutional neural networks," *Journal of Advanced Research*, vol. 35, pp. 33–48, 2022.

[29] K. He, X. Zhang, S. Ren and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1904–1916, 2015.

[30] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE CVPR*, Columbus, USA, pp. 580–587, 2014.

[31] R. Girshick, "Fast R-CNN," in *Proc. IEEE ICCV*, Santiago, Chile, pp. 1440–1448, 2015.

[32] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[33] T. Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan *et al.,* "Feature pyramid networks for object detection," in *Proc. IEEE CVPR*, Honolulu, USA, pp. 936–944, 2017.

[34] K. He, G. Gkioxari, P. Dollár and R. Girshick, "Mask R-CNN," in *Proc. IEEE ICCV*, Venice, Italy, pp. 2980–2988, 2017.

[35] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed *et al.,* "SSD: Single shot multibox detector," in *Proc. ECCV*, Amsterdam, Netherlands, pp. 21–37, 2016.

[36] T. -Y. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, "Focal loss for dense object detection," in *Proc. IEEE ICCV*, Venice, Italy, pp. 2999–3007, 2017.

[37] Roboflow. Roboflow: Give your software the sense of sight, 2023. [Online]. Available: https://roboflow.com

[38] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li *et al.,* "A survey of deep learning-based object detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019.

[39] C. Y. Wang, H. Y. M. Liao, Y. H. Wu, P. Y. Chen, J. W. Hsieh *et al.,* "CSPNet: A new backbone that can enhance learning capability of CNN," in *Proc. IEEE/CVF CVPRW*, Virtual, pp. 390–391, 2020.

[40] S. Liu, L. Qi, H. Qin, J. Shi and J. Jia, "Path aggregation network for instance segmentation," in *Proc. IEEE/CVF CVPR*, Salt Lake City, USA, pp. 8759–8768, 2018.

[41] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2018. [Online]. Available: https://arxiv.org/abs/1804.02767

[42] Z. Tian, C. Shen, H. Chen and T. He, "FCOS: Fully convolutional one-stage object detection," in *Proc. IEEE/CVF ICCV*, Seoul, Korea, pp. 9626–9635, 2019.

[43] X. Zhou, D. Wang and P. Krähenbühl, "Objects as Points," 2019. [Online]. Available: https://arxiv.org/abs/1904.07850

[44] Y. -S. Deng, A. -C. Luo and M. -J. Dai, "Building an automatic defect verification system using deep neural network for PCB defect classification," in *Proc. ICFSP*, Poitiers, France, pp. 145–149, 2018.

[45] X. Zhu, W. Su, L. Lu, B. Li, X. Wang *et al.,* "Deformable DETR: Deformable transformers for end-to-end object detection," in *Proc. ICLR*, Virtual, pp. 1–16, 2021.

[46] H. Shin, Y. Ahn, S. Tae, H. Gil, M. Song *et al.,* "Enhancement of multi-class structural defect recognition using generative adversarial network," *Sustainability*, vol. 2021, no. 13, pp. 1–13, 2021.

[47] B. Bosquet, D. Cores, L. Seidenari, V. M. Brea, M. l. Mucientes *et al.,* "A full data augmentation pipeline for small object detection based on generative adversarial networks," *Pattern Recognition*, vol. 133, pp. 1–12, 2023.

[48] B. Zoph, E. D. Cubuk, G. Ghiasi, T. Lin, J. Shlens *et al.,* "Learning data augmentation strategies for object detection," in *Proc. ECCV*, Virtual, pp. 1–19, 2020.

[49] F. Shao, L. Chen, J. Shao, W. Ji, S. Xiao *et al.,* "Deep learning for weakly-supervised object detection and localization: A survey," *Neurocomputing*, vol. 496, no. 28, pp. 192–207, 2022.

[50] K. Han, Y. Wang, H. Chen, X. Chen, J. Guo *et al.,* "A survey on vision transformer," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 45, no. 1, pp. 87–110, 2023.

[51] S. Zhang, J. Lu, H. Zhao, X. Zhu, Z. Luo *et al.,* "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," in *Proc. CVPR*, Nashville, USA, pp. 6877–6886, 2021.

[52] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez *et al.,* "SegFormer: Simple and efficient design for semantic segmentation with transformers," in *Proc. NeurIPS*, Virtual, pp. 12077–12090, 2021.

[53] W. Fang, L. Wang and P. Ren, "Tinier-YOLO: A real-time object detection method for constrained environments," *IEEE Access*, vol. 8, pp. 1935–1944, 2019.

[54] D. T. Nguyen, T. N. Nguyen, H. Kim and H. Lee, "A High-throughput and power-efficient FPGA implementation of YOLO CNN for object detection," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 27, no. 8, pp. 1861–1873, 2019.