European Journal of Interdisciplinary Research and Development
Volume-23 January 2024
Website: www.ejird.journalspark.org ISSN (E): 2720-5746

A COMPREHENSIVE EVALUATION OF YOLOv5s AND YOLOv5m FOR DOCUMENT LAYOUT ANALYSIS

1st Huda Salim
Department of Computer Engineering Technology,
Technical College of Mosul Northern Technical University, Mosul, Iraq
email: huda.salim@ntu.edu.iq

2st. Fadwa S. Mustafa
Technical Institute of Mosul,
Northern Technical University Mosul, Iraq
email: fadwaalezzo@ntu.edu.iq

Abstract
Document Layout Analysis (DLA) in images, is highly dynamic within computer vision. Presently, deep learning architectures, particularly YOLOv5s and YOLOv5m, take the forefront in addressing this challenge This paper meticulously examines their performance, both qualitatively and quantitatively, measured by Average Precision (AP) on COCO datasets. Significant improvements are observed through fine-tuning specific datasets, notably books in Arabic and English languages. A comparative evaluation of YOLOv5m and YOLOv5s in the realm of DLA unfolds. Despite YOLOv5s showcasing an impressive Frames Per Second (FPS) of 123, surpassing YOLOv5m by 2 units, the latter proves to be the optimal model for DLA systems. Its comprehensive performance superiority shines through, boasting an mAP of 94.2%, outperforming other models in this study. Noteworthy is YOLOv5m's lower FPS, compensated by its respectable detection speed, rendering it a pragmatic choice for real-world applications where accuracy is paramount.

Keywords: Computer Vision, Document Layout Analysis, YOLOvs, YOLOv5m.

INTRODUCTION

A document image is composed of a variety of physical entities or regions, such as text blocks, lines, words, figures, tables, and backgrounds. Document images are often generated from physical documents by digitization using scanners or digital cameras. The physical layout of a document refers to the physical location and boundaries of various regions in the document image. The process of Document Layout Analysis(DLA) aims to decompose a document image into a hierarchy of homogenous regions[1].

DLA is a task within the field of computer vision. The DLA plays a crucial role in applications like content comprehension and the extraction of knowledge from image-based documents. Despite its significance, DLA faces ongoing challenges in real-world scenarios due to language variations, font diversity, handwritten text, and document types. These factors bring about complexities that demand careful attention to guarantee accurate and robust DLA[2].

DLA allows the extraction of targeted information from specific regions within a document rather than extracting text from the entire document image, which is a common method using Optical Character Recognition (OCR). It forms an essential aspect of document image analysis and recognition procedures. During this process, the document images usually go through pre-processing steps, including noise reduction, followed by layout analysis, character recognition, and symbol identification[3].

Recently, the use of deep learning approaches has become more prevalent in DLA[4]One common approach in this context is to utilize general object detection models like You Only Look Once (YOLO) Version 5 [5].

The proposed system employed the DLA technique with computer vision, contributing to accurately identifying the structure and components of the document. This integrated approach maximizes the potential of analysis capabilities, ensuring effective extraction and precise conversion of information into usable outputs.

Consequently, the proposed system focuses on developing a system capable of extracting and recognizing different document layout categories, including title, sub-title, text, figure, table, page number, references, figure caption, and table caption, from document images.

LITERATURE REVIEW

This section offers a concise summary of the research compilation and prior studies related to DLA.

In 2017, Ibrahim M. Amer et al. introduced a method for Arabic DLA using DL. It was developed as an algorithm for analyzing Arabic newspapers using a Deep Convolutional Neural Network (D-CNN), as it achieves great results for tasks related to machine vision. It was shown that using a proper pre-trained model and a small dataset, it could achieve good results by applying the concept of transfer learning. used two classification methods: zone-based classification and patch-based classification. The system has been evaluated on 40 document images (10 skewed and 30 non-skewed); the best results achieved for text and non-text classification are precision = 0.96, recall = 0.87, and F-score = 0.91. Using ImageNet[6] and Arabic printed text image (APTI) datasets[6], trained a network to discriminate between images and text and then fine-tuned the network using a very small dataset collected from Arabic popular newspapers images. ImageNet was used to extract the visual features of the images regardless of their categories because the purpose was only to discriminate between text and non-text components; the APTI dataset was also used to capture the visual features of the text.[7]

In 2019, Hui-Yin Wu et al. introduced a paper[8] focused on the multi-layered analysis of newspapers. Their research employed computer vision techniques to segment newspapers into meaningful blocks. These blocks were subsequently processed by a Convolutional Neural Network (CNN), specifically a Residual Network (ResNet) with 50 layer, for classification purposes. The classification covered 23 different design-related labels at varying levels. Higher-level labels included articles, text boxes, or ads, while lower-level categories encompassed images, columns, and paragraphs. Additionally, they introduced various classes for text, such as headlines, captions, author attributions, and different image classes. The dataset utilized in this study was sourced from news publications, including well-known sources like the New York

Times, during the period of February to April 2019. The training dataset comprised 43 pages and featured a total of 1,600 annotations. Their research yielded a accuracy of 83% on their test dataset, which consisted of pages from the same newspapers as the training dataset but from different editions. The paper also presented an analysis of classification errors for each class and concluded that classes with higher error rates were often underrepresented in the training dataset.

In 2019, Carlos X. Soto et al. introduced an approach for segmenting elements in scientific papers, as detailed in[9]. They used a faster Region-based Convolutional Neural Network(R-CNN) object detection model to identify and categorize nine distinct region classes: titles, authors, abstracts, body text, figures, figure captions, tables, table captions, and references. In addition to the image features derived from the feature extraction process within Faster R-CNN, they leveraged contextual features related to the page and the Region of Interest (RoI) bounding boxes. Their rationale was grounded in the fact that, unlike object detection in images like MS COCO, where objects can appear anywhere, documents tend to follow specific layouts. For example, titles typically appear at the top of a page. Consequently, the inclusion of contextual features pertaining to the position and size of RoI bounding boxes had the potential to enhance performance. The dataset used for this study comprised 100 manually annotated scientific articles. These articles were utilized to train two models: a standard Faster R-CNN and one augmented with additional contextual features. They adopted a ResNet-101 backbone network pre-trained on ImageNet. Their results included per-class performance metrics during training, which exhibited considerable variation between classes, with the lowest for authors and table captions and the highest for the body text. They also presented the mean Average Precision (mAP) across all classes for both versions of Faster R-CNN. Additionally, they included YOLOv3 and RetinaNet as reference models. Notably, their implementation with contextual features demonstrated a significant improvement of over 20% compared to the baseline Faster R-CNN, with mAP increasing from 46.4% to 70.3%. The YOLOv3 and RetinaNet models fell somewhere in between these two results.

In 2020, Benjamin C et al. published a paper on the subject of DLA, The Newspaper Navigator Dataset [10] . The authors of this work employed a pre-trained Faster-RCNN object detection model, which they fine-tuned to recognize various elements within newspapers. The training data for this endeavor consisted of 48,000 annotations derived from historical newspapers of the World War I-era Chronicling America Newspapers. These annotations covered seven different classes: photographs, illustrations, maps, comics/cartoons, editorial cartoons, headlines, and advertisements. Their visual content recognition efforts resulted in a bounding box mAP of 63.4% on their validation set. Additionally, they explored the generalization capabilities of their visual content recognition model in 19th-century newspapers. In comparison, the mAP for headlines on their validation set reached 74.3%, while on newspapers from the years 1850 to 1875, the mAP was notably lower at 21.2%.

In 2023, Yani Siti Nurpazrin et al. They used the YOLOv5 algorithm, a cutting-edge computer vision model, for the purpose of rapidly identifying document layouts and extracting unstructured data. The present study establishes a conceptual framework for delineating the notion of objects as they pertain to documents, incorporating various elements such as paragraphs, tables, photos, and other constituent parts. The main objective is to create an

autonomous system that can effectively recognize document layouts and extract unstructured data, thereby improving the effectiveness of data extraction. In the conducted examination, the YOLOv5 model exhibits notable effectiveness in the task of document layout identification, attaining a high accuracy rate along with a precision value of 0.91, a recall value of 0.971, an F1-score of 0.939, and an area under the receiver operating characteristic curve (AUC-ROC) of 0.975. The remarkable performance of this system optimizes the process of extracting textual and tabular data from document images. Its prospective applications are not limited to document analysis but can encompass unstructured data from diverse sources, such as audio data. This study lays the foundation for future investigations into the wider applicability of YOLOv5 in managing various types of unstructured data[11].

PROPOED SYSTEM

The major goal is to use deep learning techniques to implement a detection of the document layout in document images in both Arabic and English. Yolov5s and YOLOv5m models are used for this task. The YOLOv5 model has demonstrated impressive capabilities in identifying text regions in document images and generating precise bounding boxes surrounding the text elements.

The comparison between the YOLOv5s and YOLOv5m models is based on performance and speed detection.

EXPERIMENTAL SETUP

1.1. Dataset

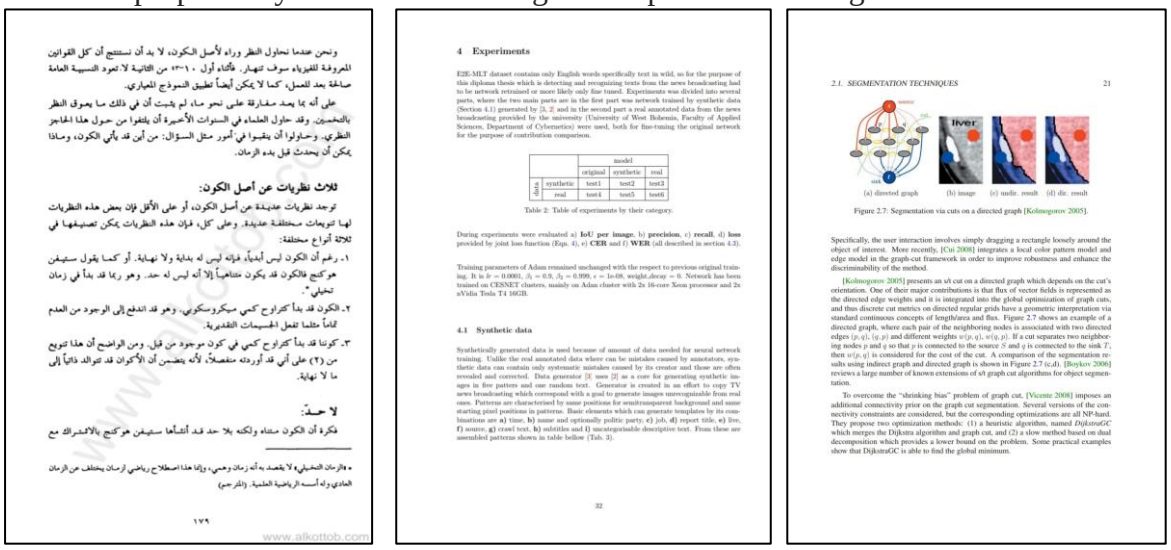
The custom dataset comprises 950 images of pages from assorted scientific and engineering books, available in both Arabic and English, sourced from accessible web repositories. The different text features of scientific and engineering books have been classified into ten main categories, as follows:

1. Title: This pertains to the primary title of the document.
2. Subtitle: A secondary title or heading that supplements the main title, often used to provide additional context or categorization.
3. Text: It includes paragraphs within the document.
4. Table: This class encompasses elements related to tables.
5. Table Caption: A descriptive text accompanying a table that provides context and explanation for the presented data.
6. Figure: It covers various visual elements like figures, charts, images, photographs, artistic creations, and visual illustrations that are part of the document image.
7. Figure Caption: A textual description accompanying a figure that explains and contextualizes the visual content.
8. List: A collection of items presented in an organized format, providing information or details on a specific topic.
9. Page Number: Page numbers are vital for cross-referencing pages and their content within the table of contents.

10. Reference: information pointing to a source or citation, typically used to acknowledge the origin of data, ideas, or quotations.

Figure 1 shows examples of custom dataset.

Figure 1 Samples of custom dataset



The proposed system relies on images as input data. Through the use of an Android application called PDF to JPG converter, the documents were divided into page images. Then, Roboflow was used to annotate these images. Text document annotation is the process of assigning labels to a text document or to its different content elements in order to distinguish the features of sentences. Figure 2 shows the document image after the annotation process in roboflow.

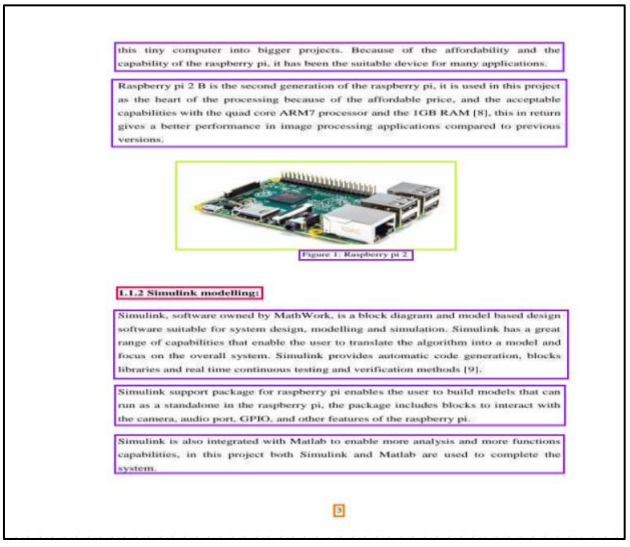


Figure 2 Document image after annotation.

Roboflow is a tool used for annotating page images within a custom dataset. Roboflow offers augmentations, which are various image processing techniques applied to the dataset to create variations of the original images. These variations enhance dataset diversity, model robustness, and generalization. The initial size of the custom dataset is 950 images, which expands to a total of 2470 images after augmentations. The image was resized to 640 x 640 pixels. the

augmentation process, including vertical flips, 90-degree rotations, cropping, rotation, brightness adjustment, Gaussian blur, and salt and pepper noise, with a 50% probability of each.

Tables 1 provide a comprehensive analysis of statistics across various object categories within custom datasets. Which is comprises 2,470 page images with 5,394 bounding boxes . There are much more samples in certain categories than in others due to differences in the occurrence frequency of those categories. Text is the most common element on most sites, making it the dominant category.

Table 1 comprehensive statistics of diverse object categories in the first dataset.

Categories	Number of bounding box
Title	641
Subtitle	352
Text	2507
Table	122
Table description	62
Figure	273
Figure description	239
List	262
Reference	65
Page number	871

YOLOv5 is a Python library that allows for the export of annotated data in a format compatible with PyTorch. This format includes JPG images, JSON annotations, and data split into training and validation folders. To use YOLOv5 in PyTorch, the data must be formatted according to the YOLO format, ensuring proper structuring of bounding box coordinates and class labels.

1.2. Deep Learning models (YOLOv5)

YOLO, short for (You Only Look Once), is a target object detection algorithm known for its compact model size and lightning-fast computation speed. YOLO's architecture is straightforward, allowing it to directly predict both the position and category of objects within bounding boxes through the neural network. YOLO's swiftness is attributed to its ability to efficiently process images, enabling real-time video detection. YOLO performs object detection using the entire image, effectively encoding global information and minimizing background detection errors. Furthermore, YOLO excels in its generalization capabilities, as it can acquire broadly applicable features that can be transferred to diverse domains[12].

YOLO encompasses multiple versions for object detection. In one of its iterations, YOLOv5 stands out as a state-of-the-art model at the forefront of object detection technology, which is developed by Ultralytics and built on the PyTorch library and is designed for practical applicability across diverse use cases. Building upon the successes of its YOLO predecessors, YOLOv5 introduces innovative features and enhancements to elevate performance and versatility. Engineered for speed, precision, and user-friendliness, YOLOv5 remains an

exceptional option for a broad spectrum of tasks, including object detection, instance segmentation, and image classification[13].

YOLOv5 adopts a pioneering architecture that integrates various levels of abstraction through a hybrid backbone network, enabling it to discern objects with heightened accuracy and efficiency. Additionally, YOLOv5 incorporates numerous enhancements, such as a revamped anchor box configuration, a Feature Pyramid Network (FPN), and refined training methodologies, all contributing to its enhanced performance[14].

YOLOv5 offers various architectures: YOLOv5n, YOLOv5s, YOLOv5m, YOLOv5l, and YOLOv5x, with differences in network depth and feature map width to meet specific needs[15].

YOLOv5s and YOLOv5m were chosen based on their compact model size and fast detection speed. Figure 3 illustrates the network architecture of YOLOv5, comprising four main components: input, backbone, neck, and head. The input component prepares the input image for processing using techniques like adaptive image scaling, mosaic data enhancement, and anchor frame calculation. The backbone component extracts features from the input image through convolutional layers, including the Conv module, C3 module, and Spatial Pyramid Pooling-Fast (SPPF) module. The neck component merges features from the backbone and enhances their representational power using the FPN and Pixel Aggregation Network structures. The head component predicts bounding boxes and class probabilities for each object in the input image using the Complete Intersection over Union (CIOU) loss and NMS to generate the final predicted image. The network aims to achieve a multi-scale feature fusion module that retains both large-scale and small-scale target feature information.[15]

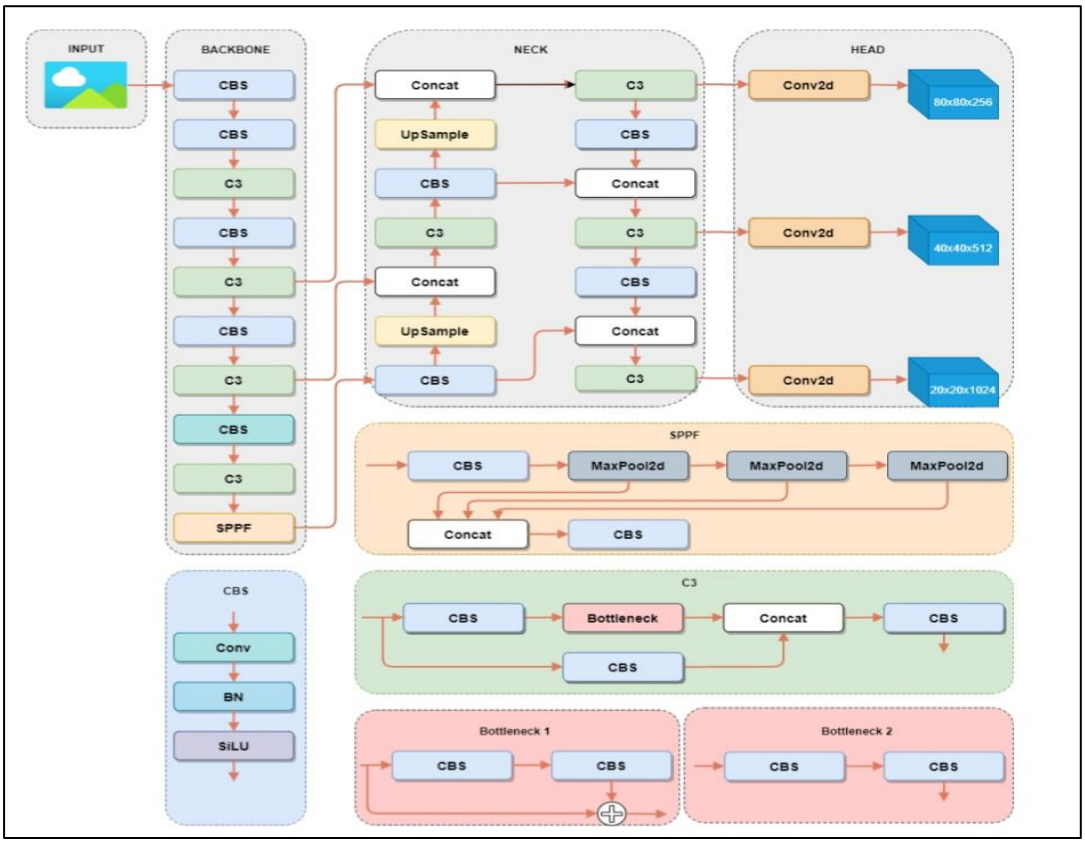


Figure 3 YOLOv5 model architecture.[15]

IMPLEMENTATION AND TRAINING RESULTS

The implementation phase serves as the core of the paper, encompassing the installation of the required software and the training of models using the provided training dataset. The objective is to enable these models to identify various categories within the test dataset containing document images. The chosen models for implementation are pre-trained models sourced from reliable pre-built models, specifically:

- YOLOv5s
- YOLOv5m.

These model files were hosted on Google Drive to facilitate access for Google Colab. The models underwent training on a customized dataset, involving code modifications to align with our specific requirements. Post-training, the models were evaluated on 20% of the original dataset to obtain mAP score. The speed of detection was measured by evaluating the time taken for each model.

There are two main steps that the implementation process takes: installing and training. The installation involves installing the necessary libraries, and training involves configuring the pre-trained models for the specific dataset, followed by the actual training process. The outcome is then evaluated by evaluating a subset of the original dataset to assess the models' performance.

The YOLOv5 model is pre-trained on the COCO dataset and trained using Google Colab. TensorBoard is used to visualize neural network training outcomes. The fine-tuning process involves preparing a custom dataset, selecting a pre-trained model from YOLOv5s/m, and adjusting its configuration file. The model is evaluated to calculate mAP and other metrics. However, this requires significant computational resources and time investment. The learning rate is 0.01, the batch size is 16, and the training epochs are 200. The optimizer is SGD.

In YOLOv5s and YOLOv5m, the findings have revealed that the best weights during the training process were saved depending on the highest value of mean Average Precision (mAP).

1.3. Training result of YOLOv5s

The total loss steadily decreased until it approached (0.0379) in YOLOv5s, as shown in Figure. 4. The train/box_loss and val/box_loss Expresses the bounding box loss during training or validation. Smaller bounding boxes lead to more accurate results. train/obj_loss and val/obj_loss are the assumed mean losses for target detection during training or validation, and smaller target detection results in more accurate detection. train/cls_loss and val/cls_loss are the assumed mean classification losses during training or validation, and a smaller classification results in a more accurate classification.

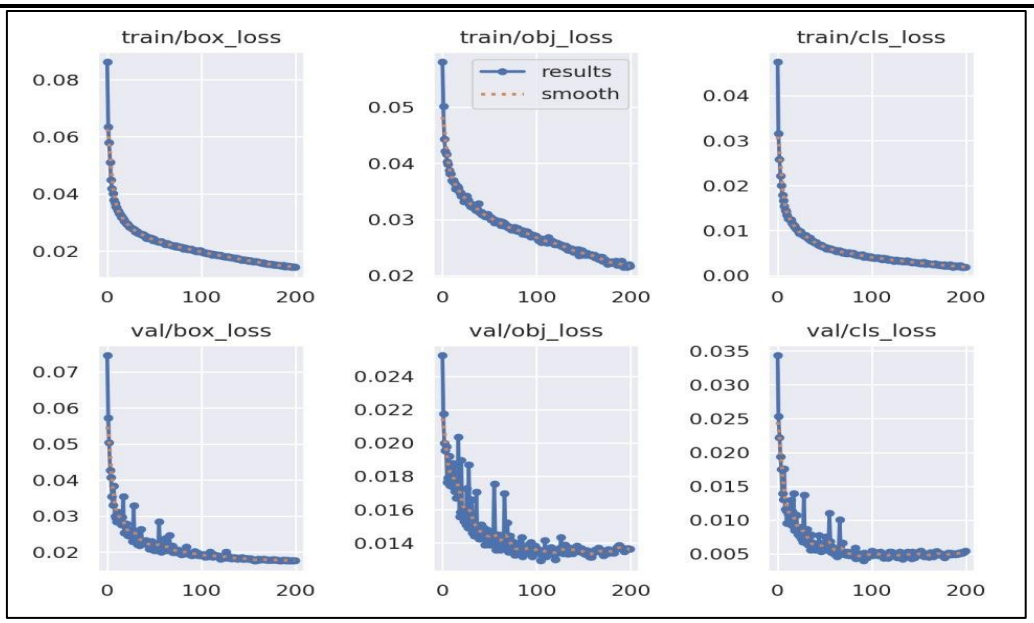


Figure 4. The Losses of Training and Validation Results of YOLOv5s.

In YOLOv5 training, it is crucial to optimize both precision and recall since high values for both metrics indicate strong overall model performance. Moreover, in practice, mAP at various Intersection Over Union (IOU) thresholds, like mAP@0.5_0.95, is often regarded as a more informative metric. This metric considers a broader range of IOU thresholds, offering a comprehensive assessment of the model's performance. Higher mAP values are indicative of superior object detection performance. This is depicted in figure 5.

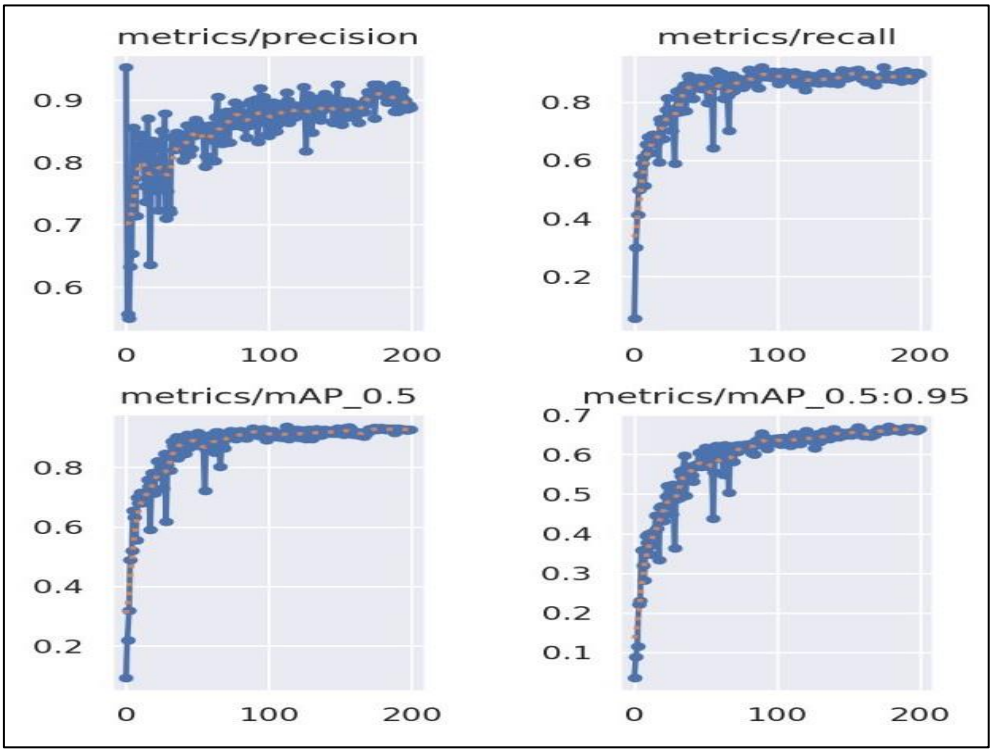


Figure 5. mAP, precision and recall Result of YOLOv5s.

1.4. Training result of YOLOv5m

The total loss steadily decreased until it approached (0.02997) as shown in Figure 6.

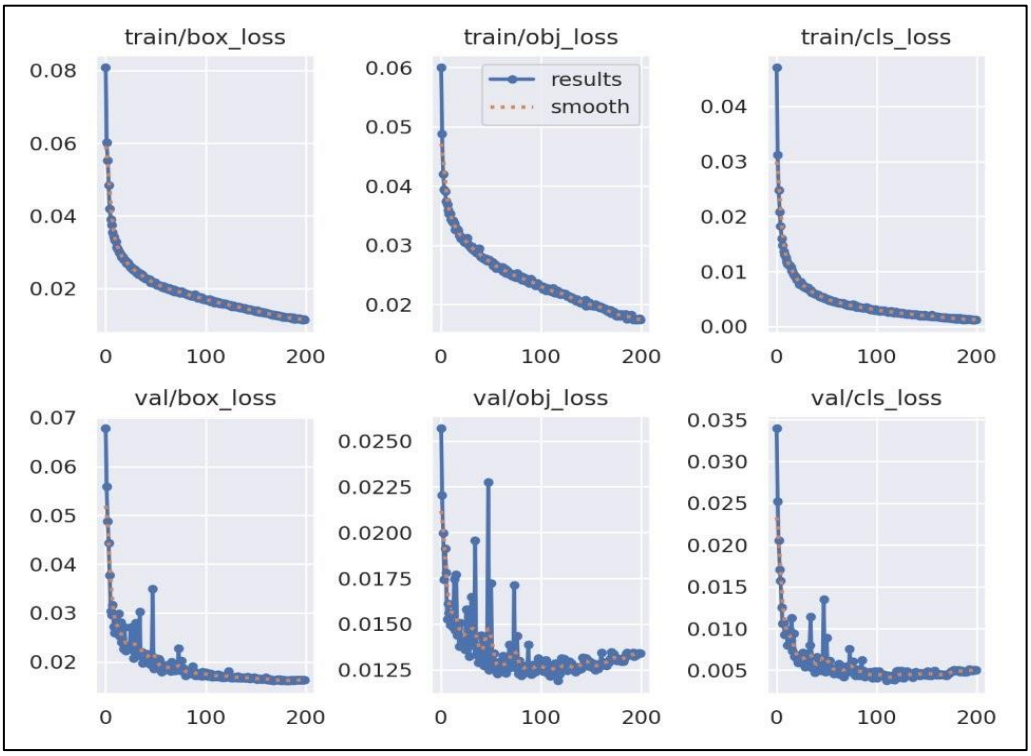


Figure 6. The Losses of Training and Validation Results of YOLOv5m.

Figure 7 Show mAP, precision and recall Result of YOLOv5m.

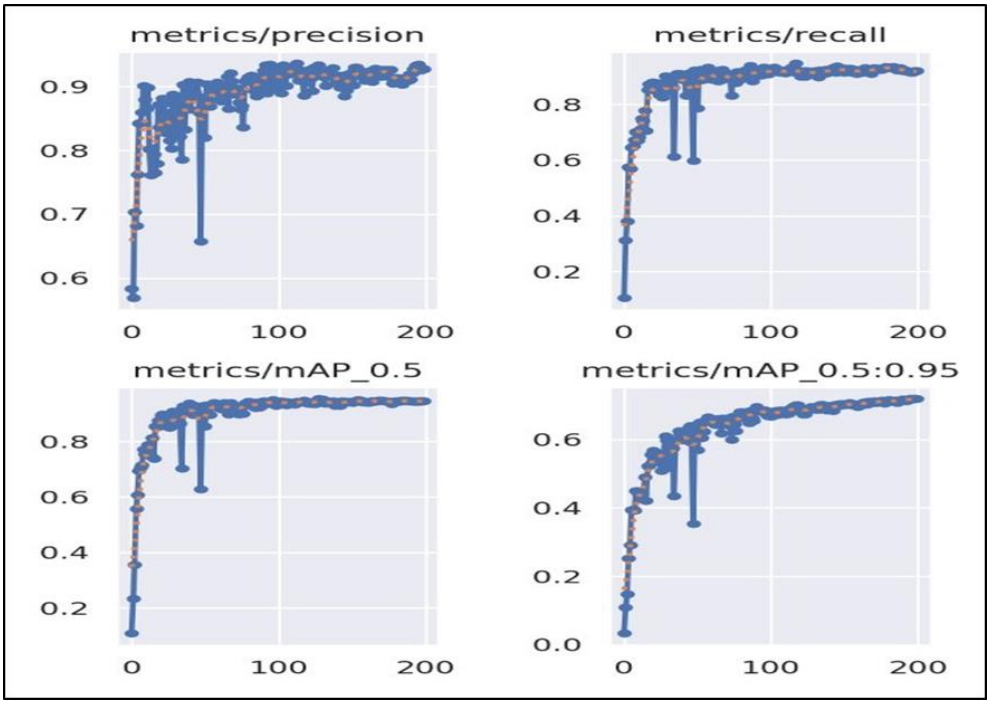


Figure 7 mAP, precision and recall Result of YOLOv5m.

2. COMPARISON IN THE EXPERIMENT

Table 1 shows a comparison of different categories between YOLOv5s and YOLOv5m at IOU=0.5.

Table 1 a comparison of the two models for different types of objects.

Category	AP@0.5[YOLOv5s](%)	AP@0.5[YOLOv5m](%)
Title	93.3	94.0
Subtitle	89.6	88.0
Text	98.0	97.5
Table	99.5	99.5
Table description	91.0	93.4
Figure	93.8	95.9
Figure description	96.8	97.6
List	84.9	86.0
Reference	88.4	95.4
Page number	97.7	98.2

YOLOv5s and YOLOv5m differ in model size and complexity. Because YOLOv5m is a larger model, it achieves higher mAP@50 values compared to YOLOv5s, as shown in the table 2, but it also requires more computational resources. The optimal choice depends on the trade-off between model accuracy and computational efficiency based on the specific use case and hardware constraints.

Table 2 mAP@0.5, Precision, Recall of YOLOv5s/m.

Model	mAP@0.5(%)	Precision(%)	Recall(%)	Model size	FPS
YOLOv5s	93.3	90.8	89.6	13.8	123
YOLOv5m	94.5	92.7	92.0	40.3	49

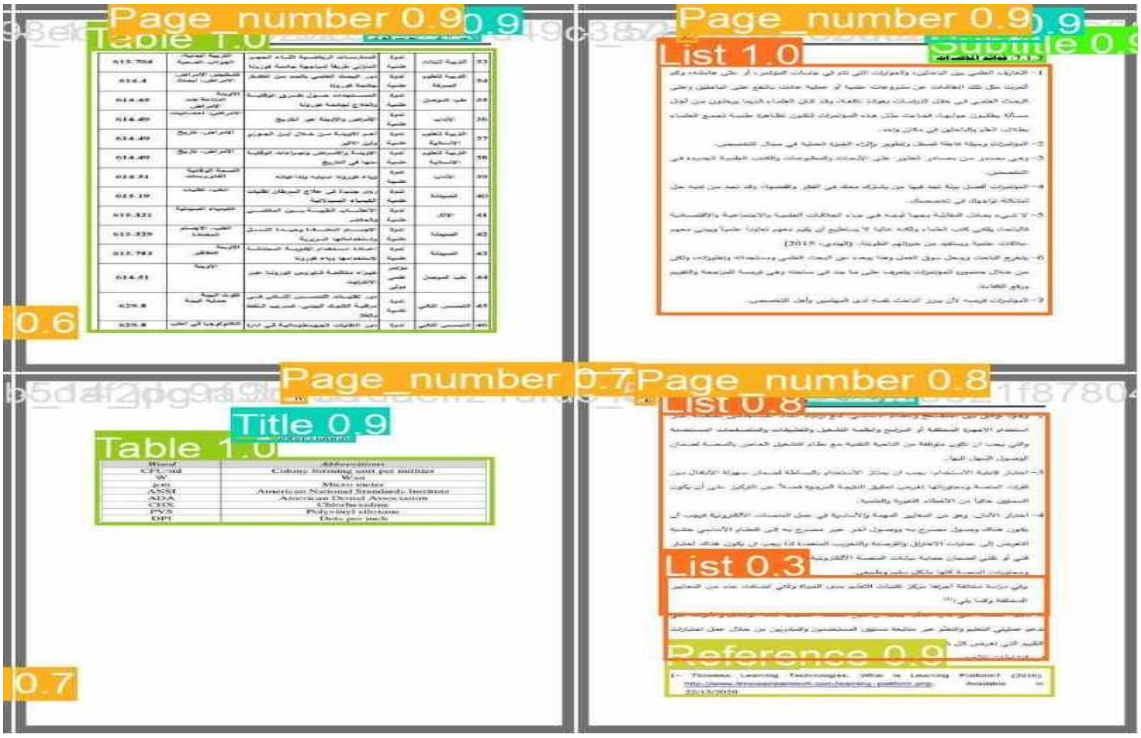
In summary, when evaluating a validation dataset, high precision is needed to minimize false positives, high recall is needed to capture most actual positives, and a high mAP@0.5 represents a balance between precision and recall at a specific IoU threshold.

The validation results indicate that the Frames Per Second (FPS), which measures the number of frames a model can process in one second, are lower for YOLOv5m compared to YOLOv5s, as shown in Table 2. The mAP serves as the primary performance indicator. FPS is utilized to assess the model's detection speed, where a higher value indicates faster detection. YOLOv5s achieves an FPS of 123, which is 2 units higher than that of YOLOv5m.

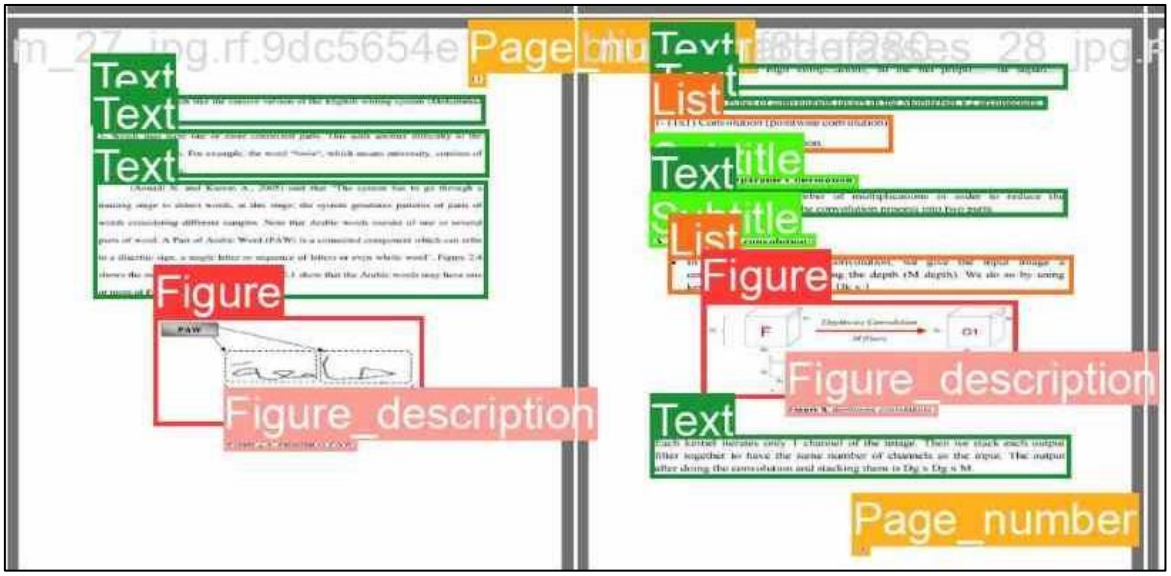
YOLOv5m stands out as an optimal model for Document Layout Analysis (DLA) systems due to its superior comprehensive performance. With an mAP reaching 94.2%, it outperforms other models in this study. Despite YOLOv5m having a lower Frames Per Second (FPS) compared to YOLOv5s, it still offers a respectable detection speed.

3. QUALITATIVE ANALSIS

Figure 8 displays outputs produced by the Yolov5s and YOLOv5m.



(a) YOLOv5s



(b) YOLOv5m

Figure 9. Samples output of Yolov5

CONCLUSION

This work presents a novel framework and dataset for parsing Arabic and English books. Going future, we want to broaden the scope of the research to include other document kinds, such ancient archival records, which frequently have a lot of noise in them. Furthermore, maintain improving minority category performance.

REFERENCES

[1] A. M. Namboodiri and A. K. Jain, "Document Structure and Layout Analysis," no. March 2007, pp. 29–48, 2007, doi: 10.1007/978-1-84628-726-8_2.

[2] Q. Peng et al., "ERNIE-Layout: Layout Knowledge Enhanced Pre-training for Visually-rich Document Understanding," Find. Assoc. Comput. Linguist. EMNLP 2022, no. 2, pp. 3744–3756, 2022, doi: 10.18653/v1/2022.findings-emnlp.274.

[3] "Deep Learning-based Table Detection in Documents."

[4] B. Wang, J. Zhou, and B. Zhang, "MSNet: A Multi-scale Segmentation Network for Documents Layout Analysis," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 12511 LNCS, no. September, pp. 225–235, 2021, doi: 10.1007/978-3-030-66906-5_21.

[5] "YOLOv5 | PyTorch." https://pytorch.org/hub/ultralytics_yolov5/ (accessed Oct. 02, 2023).

[6] Y. Takamitsu and Y. Orita, "Effect of glomerular change on the electrolyte reabsorption of the renal tubule in glomerulonephritis (author's transl)," Japanese J. Nephrol., vol. 20, no. 11, pp. 1221–1227, 1978.

[7] I. M. Amer, S. Hamdy, and M. G. M. Mostafa, "Deep Arabic document layout analysis," 2017 IEEE 8th Int. Conf. Intell. Comput. Inf. Syst. ICICIS 2017, vol. 2018-Janua, pp. 224–231, 2017, doi: 10.1109/INTELCIS.2017.8260051.

[8] H.-Y. Wu, P. Kornprobst, and P. Kornprobst Multilayered, "Multilayered Analysis of Newspaper Structure and Design," 2019, [Online]. Available: https://hal.inria.fr/hal-02177784

[9] C. X. Soto and S. Yoo, "Visual detection with context for document layout analysis," EMNLP-IJCNLP 2019 - 2019 Conf. Empir. Methods Nat. Lang. Process. 9th Int. Jt. Conf. Nat. Lang. Process. Proc. Conf., pp. 3464–3470, 2019, doi: 10.18653/v1/d19-1348.

[10] B. C. G. Lee et al., "The Newspaper Navigator Dataset: Extracting Headlines and Visual Content from 16 Million Historic Newspaper Pages in Chronicling America," Int. Conf. Inf. Knowl. Manag. Proc., pp. 3055–3062, 2020, doi: 10.1145/3340531.3412767.

[11] H. Sugiharto, Y. Silviana, and Y. S. Nurpazrin, "Unveiling Document Structures with YOLOv5 Layout Detection," pp. 1–13, 2023, [Online]. Available: http://arxiv.org/abs/2309.17033

[12] P. He, W. Huang, T. He, Q. Zhu, Y. Qiao, and X. Li, "Single Shot Text Detector with Regional Attention," Aug. 2017, [Online]. Available: http://arxiv.org/abs/1709.00138

[13] Ultralytics, "YOLOv5 | PyTorch," 2021. https://pytorch.org/hub/ultralytics_yolov5/ (accessed Jan. 13, 2024).

[14] D. Acharya, W. Yan, and K. Khoshelham, "Real-time image-based parking occupancy detection using deep learning," CEUR Workshop Proc., vol. 2087, pp. 33–40, 2018.

[15] R. Arifando, S. Eto, and C. Wada, "Improved YOLOv5-Based Lightweight Object Detection Algorithm for People with Visual Impairment to Detect Buses," Appl. Sci., vol. 13, no. 9, 2023, doi: 10.3390/app13095802.