A Hierarchical YOLO-Based Pipeline for Document Analysis in Natural Images

Seyed Mohammadreza Alavi*, Mohammad Mahdi Ebrahim Soltani*,
*Department of Electrical and Computer Engineering, University of Tehran, Tehran, Iran

***Abstract*—Extracting structured information from documents embedded in natural images remains challenging due to diverse layouts and environmental noise. We propose a hierarchical three-stage deep learning pipeline for robust document understanding: (1) document localization via instance segmentation, (2) layout analysis through semantic block-level detection, and (3) text transcription using Optical Character Recognition (OCR). We systematically benchmark recent YOLO variants (YOLOv5, v8, v11, and v12) on the localization and layout-analysis stages. Our results show that YOLOv11-Seg achieves a mask mAP of 0.935 for localization, while YOLOv11-Det attains a box mAP of 0.671 for layout analysis. In the transcription stage, EasyOCR provides strong character recovery, whereas Tesseract offers higher word-level accuracy and lexical coherence for structured images. A primary contribution is a new menu-image dataset with complex semantic structures, establishing a resource for evaluation and benchmarking. Together, our findings set a high-performance baseline for document analysis and highlight performance differences.**

***Index Terms*—Document Analysis, Instance Segmentation, Layout Analysis, Optical Character Recognition**

I. Introduction

Understanding documents captured in natural images remains a fundamental challenge in computer vision, with broad applications in augmented reality, document scanning, and intelligent data extraction. Unlike digitally born documents, images captured in unconstrained environments suffer from perspective distortions, occlusions, non-uniform illumination, specular reflections, and cluttered backgrounds. These degradations complicate document analysis by impairing layout understanding and reducing the accuracy of text extraction from semantically meaningful text blocks.

Conventional OCR engines are designed for clean, pre-aligned inputs and therefore fail to generalize effectively to natural scenes. To address this limitation, modern document understanding systems adopt multi-stage processing pipelines encompassing document localization, layout segmentation, and text recognition. While recent deep learning advances have improved each sub-task independently, relatively few studies have explored unified, multistage frameworks that integrate state-of-the-art detectors and OCR engines within a single, robust pipeline.

In this work, we present a hierarchical three-stage framework for document understanding in natural images. The pipeline first performs document localization via YOLO-based instance segmentation, followed by layout decomposition into semantically meaningful blocks-such as item sections in menus or product listings in catalogs-using YOLO-based object detection, and finally applies OCR for text recognition to generate structured outputs. To support systematic evaluation, we introduce a new dataset of restaurant menus characterized by complex block structures, stylized fonts, and high variability in design, lighting, and photographic conditions. This combination of real-world visual complexity and rich semantic structure makes menu images a compelling benchmark for robust document analysis.

Through extensive experiments, we show that careful stage-wise model selection results in a practical and high-performing pipeline for structured document analysis in natural images. We further benchmark multiple YOLO variants and OCR engines to provide detailed comparisons at each stage of the framework. Beyond menu images, our approach can also be applicable other semi-structured documents, including receipts, product catalogs, and real-world document photos captured under unconstrained conditions.

II. Related Works

Comprehensive document understanding in natural images can be decomposed into three sequential tasks: 1) **Document localization**: identifying the spatial boundaries of a document within an image; 2) **Layout analysis**: detecting semantically coherent regions, such as item sections or logical blocks; and 3) **Text recognition**: transcribing the text within each region using an OCR engine.

A. Document Localization

Document localization aims to determine the spatial boundaries of a document in a natural image. Early low-level methods relied on handcrafted features, including corners and contour-based methods. However, these methods were often sensitive to noise, scale selection, threshold choice, and edge quality. Model-based methods fit local image patches to predefined models, and some variants were computationally expensive [1].

Deep learning-based methods, particularly instance segmentation models, have substantially advanced document localization by learning document layout and contextual structure directly from data. Mask R-CNN [2] extended Faster R-CNN with a parallel segmentation branch and RoIAlign, achieving accurate instance segmentation in natural images. Nevertheless, two-stage architectures incur significant computational cost, motivating the development of single-stage

alternatives such as YOLO [3], which are optimized for real-time performance [4]. YOLOv5-Seg introduced instance segmentation models in version 7.0 [5], [6], while YOLOv8-Seg adopted an anchor-free split head and the C2f module [7], [8]. More recent variants further reduce post-processing overhead through NMS-free designs, such as YOLOv10 [9]. YOLOv11-Seg incorporates architectural enhancements including C3K2, C2PSA, and depthwise convolutions (DWConv), achieving strong performance with an mAP@0.5 of approximately 0.808 under challenging conditions [10].

B. Layout Analysis

Layout analysis, also referred to as semantic block detection, focuses on identifying meaningful content regions within a document. Early CNN-based pipelines employed region-based detectors such as R-CNN and Fast R-CNN [11], [12]. R-CNN achieved high accuracy but suffered from slow inference, while Fast R-CNN improved efficiency by sharing computation across proposals [11], [12]. Faster R-CNN [13] addressed the remaining proposal bottleneck by introducing a Region Proposal Network (RPN) that shares convolutional features with the detector, improving efficiency while retaining a two-stage structure.

Single-stage detectors, including YOLO, SSD [14], and RetinaNet [15], avoid an external proposal stage and offer substantially higher inference speed with competitive accuracy. For document layout analysis, DocLayout-YOLO [16] represents a notable advancement: a unimodal YOLO-based detector pre-trained on document-specific data and augmented with a global-to-local receptive-field module. It achieves accuracy comparable to or exceeding multimodal transformer-based baselines, while operating up to $14\times$ faster than DiT-Cascade-L on identical hardware.

Recent YOLO variants further improve performance through refinements in backbone design, detection heads, and training strategies [4]. YOLOv5 v7.0 introduced YOLOv5-seg instance segmentation models and segmentation workflows [5], [6]. YOLOv8 introduces an anchor-free split head [7]. YOLOv10-Det adopts an end-to-end, NMS-free detection framework to reduce latency without compromising accuracy [9]. YOLOv12 further integrates Area Attention and Residual Efficient Layer Aggregation Networks (R-ELAN), achieving improved accuracy while maintaining real-time throughput [17].

C. Text Recognition

Optical Character Recognition constitutes the final stage of the pipeline, converting localized text regions into machine-readable content. Early OCR systems relied on handcrafted features and rigid preprocessing pipelines, resulting in limited robustness to distortions, rotations, and diverse font styles. Contemporary OCR engines reflect a transition toward deep learning-based recognition architectures.

Tesseract [18], one of the most widely adopted OCR systems, provides efficient general-purpose text recognition. However, it remains susceptible to errors when applied to rotated or heavily stylized text. **EasyOCR** [19] employs a CRNN architecture combining ResNet-based feature extraction with CTC-based decoding, providing strong multilingual support but exhibiting reduced robustness on densely packed or irregular layouts. **Kraken** [20] uses a hybrid convolutional-recurrent neural network and supports custom training for specialized or historical scripts, making it well suited for non-standard OCR domains.

In summary, these systems illustrate the evolution from lightweight, general-purpose OCR engines toward more flexible and adaptable frameworks capable of handling the wide variability of text encountered in real-world document images.

III. Methodology

Our approach employs a three-stage modular pipeline for document understanding in natural images, explicitly balancing the accuracy–latency trade-offs observed in prior research. Stage 1 performs document localization through instance segmentation, separating the document from its background. Stage 2 detects semantically meaningful layout components, such as product or menu sections, within the localized document using object detection. Stage 3 applies an OCR engine to each identified block, transcribing the visual text into machine-readable form. This hierarchical design enables independent optimization of each stage and facilitates systematic benchmarking of detection and recognition modules. Although the pipeline can pass each stage's output to the next stage, our experiments use the corresponding annotated data at each stage for consistent stage-wise evaluation. We used Google Colab[1] to run our code. The materials for the Stages 1 and 2 are available on this GitHub repository[2] and the materials for Stage 3 on this GitHub Repository[3]. Also, our two datasets for Stages 1 and 2 are available on this GitHub repository[4].

A. Data Collection and Annotation

To evaluate the proposed methodology, we curated a domain-specific dataset comprising approximately 300 restaurant menu images captured by different users under diverse lighting, perspectives, noise levels, and cluttered backgrounds. Each image is annotated hierarchically at two levels corresponding to the first two stages of the pipeline using the Roboflow workspace[5]:

1) **Document Annotations (Stage 1)**: Polygonal instance segmentation masks specifying the full spatial extent of each document.
2) **Region Annotations (Stage 2)**: Axis-aligned bounding boxes defining semantically meaningful blocks within the localized document.

To preserve hierarchical consistency, region annotations are created within the document areas for Stage 1. This hierarchical annotation scheme reflects the design of the pipeline and

[1]https://colab.research.google.com/
[2]https://github.com/mralavi20/YOLO-OCR-Pipeline-for-Document-Analysis
[3]https://github.com/MahdiES0/Vision-Project-Document-Detection
[4]https://github.com/mralavi20/YOLO-OCR-Pipeline-for-Document-Datasets
[5]https://roboflow.com/

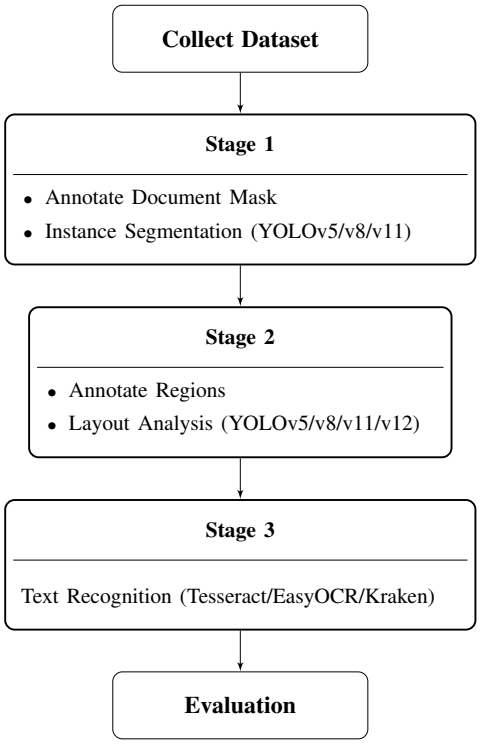


Fig. 1: Proposed hierarchical pipeline for structured document understanding in natural images.

maintains consistency by using the same train, validation, and test splits across stages. Additionally, we manually transcribed the text contained in the annotated Stage 2 regions to establish ground-truth references, for evaluating OCR outputs.

B. Model Architecture

The proposed system is implemented as a three-stage pipeline that converts raw document images into structured textual representations. As illustrated in Figure 1, the pipeline first localizes the document, then detects semantically coherent regions within it, and finally applies an OCR engine to transcribe the textual content. The following subsections detail the specific models deployed at each stage.

1) **Document Localization**: In Stage 1, we train and evaluate three YOLO-based instance segmentation models: YOLOv5-Seg, YOLOv8-Seg, and YOLOv11-Seg. In the full pipeline, the predicted document mask can be used to crop the image before layout analysis; in our experiments, we use the annotated document region for consistency.
2) **Layout Analysis**: In Stage 2, we assess four YOLO-based object detectors: YOLOv5, YOLOv8, YOLOv11, and YOLOv12, on the semantic block annotations within cropped document regions. All detectors are trained using identical region annotations and dataset splits to ensure fair and direct comparison.
3) **Text Recognition**: In Stage 3, we evaluate three open-source OCR engines: Tesseract [18], EasyOCR [19], and Kraken [20]. In the full pipeline, OCR can be applied to regions detected in Stage 2; in our experiments, OCR is applied to annotated Stage 2 regions for consistency.

TABLE I: Document localization performance on the validation set.

Model	Epoch	mAP@.5:.95	AP@.5	Prec.	Rec.
YOLOv5-Seg	192	0.898	0.987	0.996	0.932
YOLOv8-Seg	66	0.915	0.990	0.963	**1.000**
YOLOv11-Seg	192	**0.935**	**0.995**	**1.000**	**1.000**

IV. Results

We evaluate the proposed pipeline on the restaurant menu dataset using fixed training, validation, and test splits. The results are presented separately for each stage of the pipeline.

A. Document Localization

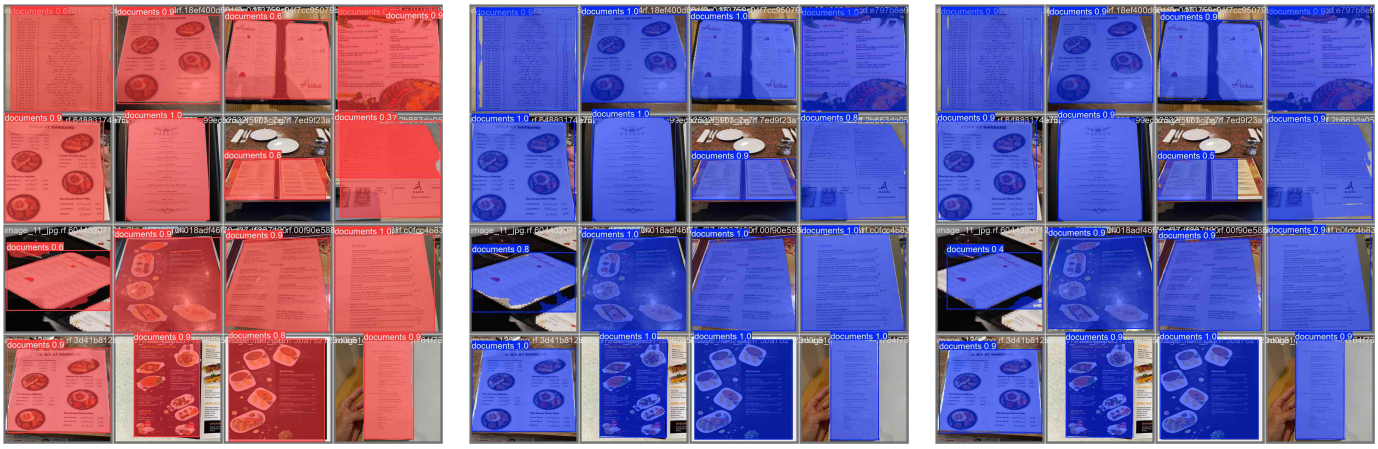
1) YOLOv5-Seg: The best-performing checkpoint (epoch 192) achieves a mask mAP@[0.5:0.95] of 0.898 and a mask AP@0.5 of 0.987, with precision 0.996 and recall 0.932. Validation losses at this epoch are 0.0038 for the mask head and 0.0039 for the box head. The precision–recall curve at IoU = 0.75 is shown in Fig. 3a, and qualitative overlays in Fig. 2a demonstrate accurate mask predictions under varying perspective and illumination conditions.

2) YOLOv8-Seg: The optimal checkpoint occurs at epoch 66, earlier than for YOLOv5-Seg. It achieves a mask mAP@[0.5:0.95] of 0.915 and a mask AP@0.5 of 0.990, with precision 0.963 and recall 1.000. The precision–recall curve is shown in Fig. 3b, and qualitative results in Fig. 2b indicate consistent and tightly aligned document masks across diverse imaging conditions.

3) YOLOv11-Seg: The best checkpoint occurs at epoch 192, yielding a mask mAP@[0.5:0.95] of 0.935 and a mask AP@0.5 of 0.995, with both precision and recall equal to 1.000. Validation losses at this epoch are 0.643 for the mask head and 0.385 for the box head. The precision–recall curve is shown in Fig. 3c, and the visual overlays in Fig. 2c highlight precise segmentation under perspective and illumination variations.

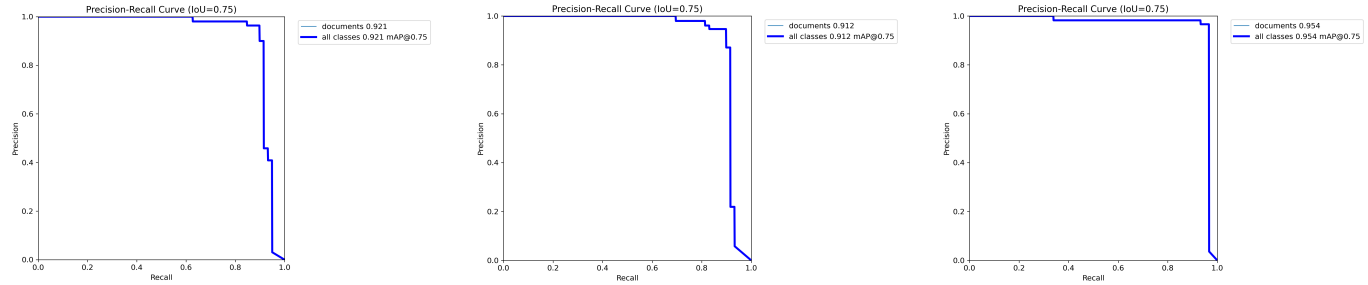
B. Layout Analysis

1) YOLOv5-Det: The best-performing checkpoint (epoch 297) achieves a box mAP@[0.5:0.95] of 0.602 and a box AP@0.5 of 0.804, with precision 0.743 and recall 0.739. The precision–recall curve in Fig. 4a shows an AP@0.5 of 0.804 and a decline in precision at very high recall, consistent with the summary metrics. Validation losses at this epoch are 0.0269 for the box regression loss and 0.0141 for the confidence (objectness) loss. Qualitative overlays in Fig. 5a illustrate reliable localization of large section blocks across varying perspectives and illumination conditions.



(a) YOLOv5-Seg (b) YOLOv8-Seg (c) YOLOv11-Seg

Fig. 2: Comparison of qualitative overlay results.



(a) YOLOv5-Seg (b) YOLOv8-Seg (c) YOLOv11-Seg

Fig. 3: Comparison of Precision-Recall (PR) curve for instance segmentation models.

2) YOLOv8-Det: The optimal checkpoint occurs at epoch 357. It attains a box mAP@[0.5:0.95] of 0.640 and a box AP@0.5 of 0.845, with precision 0.769 and recall 0.746. The precision–recall curve in Fig. 4b reflects this behavior, showing high precision up to very high recall levels. Validation losses at this epoch are 0.874 for the box regression loss, 1.016 for the classification loss, and 1.518 for the Distribution Focal Loss. Qualitative examples in Fig. 5b demonstrate consistent and accurate localization of section blocks under perspective change.

3) YOLOv11-Det: The best checkpoint occurs at epoch 355, yielding a box mAP@[0.5:0.95] of 0.671 and a box AP@0.5 of 0.842, with precision 0.878 and recall 0.783. The precision–recall curve in Fig. 4c shows stable precision across the recall range, consistent with these metrics. Validation losses at this epoch are 0.891 for the IoU-based regression loss, 0.870 for the classification loss, and 1.391 for the Distribution Focal Loss. Qualitative overlays in Fig. 5c highlight accurate localization of large semantic blocks under diverse conditions.

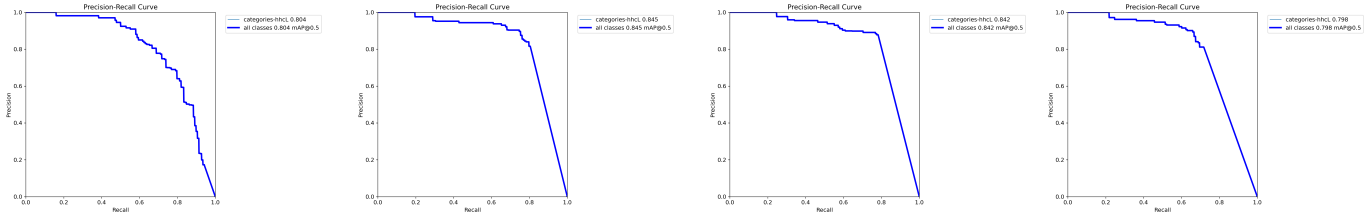
4) YOLOv12-Det: The best checkpoint occurs at epoch 453. It achieves a box mAP@[0.5:0.95] of 0.598 and a box AP@0.5 of 0.798, with precision 0.792 and recall 0.717. The precision–recall curve in Fig. 4d shows high precision up to mid–high recall levels, consistent with the overall evaluation metrics. Validation losses at this epoch are 0.976 for the IoU-based regression loss, 1.000 for the classification loss, and 1.448 for the Distribution Focal Loss. Qualitative overlays in Fig. 5d demonstrate reliable detection of large section blocks despite perspective and illumination variations.

TABLE II: Layout analysis results on the validation set. All metrics are box-based.

Model	Epoch	mAP@.5:.95	AP@.5	Prec.	Rec.
YOLOv5-Det	297	0.602	0.804	0.743	0.739
YOLOv8-Det	357	0.640	**0.845**	0.769	0.746
YOLOv11-Det	355	**0.671**	0.842	**0.878**	**0.783**
YOLOv12-Det	453	0.598	0.798	0.792	0.717

C. Text Recognition

We evaluated three OCR engines-EasyOCR, Tesseract, and Kraken-using complementary metrics: Character Error Rate (CER), Word Error Rate (WER), and Bag-of-Words (BoW) based precision, recall, and F_1-score. Lower CER and WER denote higher recognition accuracy, while higher BoW metrics reflect higher retrieval-oriented performance. The quantitative



(a) YOLOv5-Det (b) YOLOv8-Det (c) YOLOv11-Det (d) YOLOv12-Det

Fig. 4: Comparison of Precision-Recall (PR) curve for object detection models.

(a) YOLOv5-Det (b) YOLOv8-Det (c) YOLOv11-Det (d) YOLOv12-Det

Fig. 5: Comparison of qualitative overlay results.

results are summarized in Table III and runtime comparisons are reported in Table IV.

TABLE III: Performance comparison of OCR engines.

OCR Engine	CER	WER	Precision	Recall	F1
EasyOCR	**0.4924**	0.9271	0.6074	**0.5974**	0.5931
Tesseract	0.5002	**0.6748**	**0.6494**	0.5933	**0.6059**
Kraken	0.6624	0.9016	0.3051	0.2425	0.2587

TABLE IV: Runtime comparison of OCR engines

OCR Engine	Runtime (mm:ss)
EasyOCR	19:36
Tesseract	29:41
Kraken	**15:21**

1) EasyOCR: EasyOCR achieved the lowest CER of 0.4924 among the three systems, indicating strong character-level recognition accuracy. Its runtime lies between those of Tesseract and Kraken, balancing speed and accuracy. However, the WER of 0.9271 reveals frequent issues with word boundary detection and lexical consistency. Consequently, EasyOCR's BoW $F1$-score of 0.5931 suggests that while individual characters are often recognized correctly, they are not consistently grouped into valid tokens. This limits its suitability for tasks where word-level coherence and semantic structure are essential. Figure 6 presents a representative qualitative example for EasyOCR.

2) Tesseract: Tesseract obtained the lowest WER (0.6748) and the highest BoW $F1$-score (0.6059), producing more semantically coherent text than the other engines. Although its CER of 0.5002 was marginally higher than EasyOCR's, its balanced trade-off between character accuracy and lexical integrity makes it the most effective overall. Despite its longer runtime (29:41), Tesseract benefits from robust word-level alignment, yielding outputs that form valid lexical units-a key advantage for downstream tasks such as structured text extraction and semantic parsing. Overall, Tesseract demonstrates the best balance between recognition fidelity and semantic consistency. Figure 7 presents a representative qualitative example for Tesseract.

3) Kraken: Kraken performed the weakest across all evaluation metrics. Although it achieved the shortest runtime (15:21), this was partly due to skipped images on which recognition failed. It recorded the highest CER (0.6624) and WER (0.9016), as well as low BoW metrics (precision 0.3051, recall 0.2425, F_1 0.2587). These results reveal substantial generalization errors, likely due to domain mismatch: Kraken's pretrained models are optimized for digitized or historical scripts rather than natural-scene text. Despite its flexibility and support for custom training, its out-of-the-box performance is insufficient for robust recognition in natural images without domain-specific fine-tuning. Figure 8 presents a representative qualitative example for Kraken.

V. Conclusion

This paper presented a three-stage pipeline for document analysis in natural images. The proposed framework sequentially performs document localization via instance segmentation, layout analysis through detection of semantically coherent regions, and optical character recognition for textual transcription. A key contribution is the systematic benchmarking

Extracted text by EasyOCR:

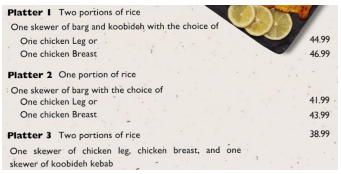
```
Appetizers
Miiza Ghassemi (smoked
leggplant &
tomato dip served with pita bread)
"vegetarian
.S6.50
Mast-o-Khiar (Yogurt cucumber dip
served with pita bread)
"vegetarian.
S3.99
Dolmeh (stuffed grape leaves)
"vegetarian_
S3.99
Hummus served with pita bread
"vegetarian_._
-53.50
Falafel (3pieces of falafel)
"vegetarian_
53.50
Small appetizer plate (1-Zpeople)
mix of all of the above _
.S9.99
Large appetizer plate .(2-4people)
mix of all the above
S16.99
```

Fig. 6: Representative qualitative OCR result obtained using EasyOCR.

Extracted text by Tesseract:

```
Shawarma Plate
with Basmati roe
and Salad, $11.99
```

Fig. 7: Representative qualitative OCR result obtained using Tesseract.



Extracted text by Kraken:

```
Platter 1 Two portlons of rice
One skewer of barg and koobideh wich the choice of
One chicken eg or
One chicken Breast
Platter 2 One portion of rice
One skewer of barg with the choice of
One chicken eg or
One chicken Breast
Platter 3 Two portions of rice

One skewer of chicken leg, chicken breast, and one
skewer of koobideh kebab
4499
4699
41,99
43.99
3899
```

Fig. 8: Representative qualitative OCR result obtained using Kraken.

of recent YOLO variants across the localization and layout analysis stages. The results establish a robust baseline for structured document understanding in natural images, emphasizing the performance differences among localization, layout-analysis, and OCR modules. The reported detection metrics, including mean Average Precision, precision, and recall, align with standard object detection evaluation protocols.

A. Document Localization

YOLOv11-Seg achieves the highest mask mAP@[0.5:0.95] at 0.935 and the highest AP@0.5 at 0.995, with both precision and recall equal to 1.000. YOLOv8-Seg ranks second in mask mAP@[0.5:0.95] at 0.915 and also achieves a recall of 1.000, while YOLOv5-Seg obtains a mask mAP@[0.5:0.95] of 0.898 and an AP@0.5 of 0.987. Overall, YOLOv11-Seg is preferred when mask quality across IoU thresholds is the primary concern, whereas YOLOv8-Seg is advantageous when high recall is critical while maintaining strong precision. YOLOv5-Seg remains a reliable baseline, demonstrating robust performance under perspective and illumination variations.

B. Layout Analysis

Across detectors, performance is consistent with clear trade-offs. In box mAP@[0.5:0.95], YOLOv11-Det ranks first (0.671), followed by YOLOv8-Det (0.640), YOLOv5-Det (0.602), and YOLOv12-Det (0.598). The box precision-recall curves support these results: AP@0.5 is 0.845 for YOLOv8-Det, 0.842 for YOLOv11-Det, 0.804 for YOLOv5-Det, and 0.798 for YOLOv12-Det. YOLOv11-Det attains the highest precision and recall (0.878 and 0.783), indicating strong localization with relatively few false alarms and misses. YOLOv8-Det follows (precision 0.769, recall 0.746) and provides a strong alternative with competitive AP@0.5. YOLOv5-Det remains a solid baseline (0.743, 0.739), while YOLOv12-Det lags in recall (0.717) and maintains moderate precision (0.792). Overall, YOLOv11-Det is preferred when maximizing box mAP across IoU thresholds is the priority, whereas

YOLOv8-Det is attractive when emphasizing AP@0.5 and balanced operating characteristics.

C. Text Recognition

Performance in the OCR stage strongly depended on the precision of document localization and layout segmentation. Using the best-performing detectors improved text coverage and reduced character and word error rates. Remaining recognition errors were largely caused by degradation factors such as glare, strong perspective distortion, and low-contrast fonts.

A comparative analysis of OCR engines revealed that EasyOCR provided high character-level accuracy, but often failed to maintain word integrity. In contrast, Tesseract offered a better balance, achieving high word-level accuracy and lexical coherence, making it ideal for parsing structured natural images. The Kraken engine, while adaptable in principle, performed poorly on our dataset without domain-specific fine-tuning. The optimal choice of the OCR engine depends on the specific downstream task. EasyOCR is preferable for fine-grained character recovery, while Tesseract offers a better trade-off when semantically coherent transcriptions are essential.

D. Future Work

Future research will focus on extending this pipeline toward an integrated, end-to-end architecture unifying localization, layout analysis, and text recognition to enhance computational efficiency. Further fine-tuning of YOLO detectors and OCR engines on domain-specific datasets is expected to mitigate common errors such as glare or low-contrast degradation. An additional research direction is the development of a feedback loop between the OCR and detection stages to iteratively refine bounding boxes based on recognition confidence, ultimately improving both segmentation accuracy and text quality.

REFERENCES

[1] J. Wang and W. Zhang, "A survey of corner detection methods," in *2nd International Conference on Electrical Engineering and Automation (ICEEA 2018)*, 2018.

[2] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

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

[4] C.-Y. Wang and H.-Y. M. Liao, "Yolov1 to yolov10: The fastest and most accurate real-time object detection systems," *APSIPA Transactions on Signal and Information Processing*, 2024.

[5] G. Jocher, A. Chaurasia, A. Stoken, J. Borovec, NanoCode012, Y. Kwon, K. Michael, TaoXie, J. Fang, imyhxy, Lorna, Z. Yifu, C. Wong, A. V, D. Montes, Z. Wang, C. Fati, J. Nadar, Laughing, UnglvKitDe, V. Sonck, tkianai, yxNONG, P. Skalski, A. Hogan, D. Nair, M. Strobel, and M. Jain, "ultralytics/yolov5: v7.0 - yolov5 sota realtime instance segmentation," 2022.

[6] G. Jocher, "v7.0 - yolov5 sota realtime instance segmentation," 2022. [Online]. Available: https://github.com/ultralytics/yolov5/releases/tag/v7.0

[7] Ultralytics, "Explore ultralytics yolov8." [Online]. Available: https://docs.ultralytics.com/models/yolov8/

[8] ——, "Yolov8 vs. yolov5: A comprehensive technical comparison." [Online]. Available: https://docs.ultralytics.com/compare/yolov8-vs-yolov5/

[9] A. Wang, H. Chen, L. Liu, K. Chen, Z. Lin, J. Han, and G. Ding, "Yolov10: Real-time end-to-end object detection," in *38th Conference on Neural Information Processing Systems (NeurIPS 2024)*.

[10] L. He, Y. Zhou, L. Liu, and J. Ma, "Research and application of YOLOv11-based object segmentation in intelligent recognition at construction sites," *Buildings*, 2024.

[11] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[12] R. Girshick, "Fast r-CNN," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[13] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-CNN: Towards real-time object detection with region proposal networks," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *Computer Vision – ECCV 2016*, 2016.

[15] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[16] Z. Zhao, H. Kang, B. Wang, and C. He, "Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception," 2024.

[17] Ultralytics, "Yolo12: Attention-centric object detection." [Online]. Available: https://docs.ultralytics.com/models/yolo12/

[18] R. Smith, "An overview of the tesseract ocr engine," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, 2007.

[19] JaidedAI, "Easyocr." [Online]. Available: https://github.com/JaidedAI/EasyOCR

[20] B. Kiessling, "Kraken - an universal text recognizer for the humanities," in *Digital Humanities 2019*, 2019.