SAHI-BAR: An Instance Segmentation Model for Medical Prescriptions

G. R. Rekha
Department of Computer Applications, JSS Science and Technology University, Mysuru, Karnataka, India
rekha_gr@jssstuniv.in

S. Siddesha
Department of Computer Applications, JSS Science and Technology University, Mysuru, Karnataka, India
siddesh.shiv@jssstuniv.in (corresponding author)

V. N. Manjunath Aradhya
Department of Computer Applications, JSS Science and Technology University, Mysuru, Karnataka, India
aradhya@jssstuniv.in

Received: 21 November 2025 | Revised: 12 December 2025, 31 December 2025, and 7 January 2026 | Accepted: 9 January 2026

 DOI: https://doi.org/10.48084/etasr.16418

ABSTRACT

Medical prescription documents pose significant challenges for automated information extraction due to dense layouts, small text, and heterogeneous field structures. This study presents a modular pipeline that augments a YOLO-based segmentation baseline with two lightweight strategies: (i) Sliced Aided Hyper Inference (SAHI) for tiled processing with post-hoc merging and (ii) Block Aware Routing (BAR) mechanism that fuses baseline and tiled predictions while enforcing a one-entity-per-class-per-block constraint to segment prescription parameters into eight different classes. Experiments on a custom prescription dataset with eight semantic classes, namely Block_id, Med_Name, Med_Type, Dose_strength, Frequency, Duration, Quantity, and Instructions, show that the proposed approach improves recall on dense textual regions without sacrificing precision. In addition, the newer YOLOv11 architecture was evaluated, demonstrating that inference-time tiling and routing remain the dominant contributors to small-object performance gains. The proposed framework is fully compatible with the Ultralytics ecosystem, does not require retraining for tiling benefits, and produces class-specific crops for downstream OCR and archival. These results indicate a practical and deployment-friendly approach to document parsing that balances accuracy, interpretability, and efficiency.

Keywords-prescription analysis; parameter segmentation; instance segmentation; YOLOv8; YOLOv11; SAHI; routing

I. INTRODUCTION

Digitized medical prescriptions require a bridge between clinical decision-making and downstream systems such as pharmacy automation, Electronic Health Records (EHRs), and large-scale medical analytics. Reliable extraction of structured fields from prescription images, including the name, type, dosage strength, frequency, duration, and instructions of the drug, is a prerequisite for accurate treatment monitoring and patient safety. However, this task remains challenging due to handwriting variability, cluttered layouts, low print quality, and dense small-font text that frequently appears in documents [1-2].

Traditional OCR pipelines, although mature, often struggle when applied to semi-structured documents with noisy backgrounds or overlapping entities. Classical OCR engines such as Tesseract and ABBYY FineReader have long been used to digitize printed text, but often underperform on handwritten or noisy medical prescriptions. Recent surveys highlight that OCR accuracy deteriorates when text is small, cluttered, or irregularly aligned [1]. Early word segmentation in document images relied on classical image processing with Connected Components (CC) analysis with morphological filtering and size heuristics [3], Graph-cut formulations [4], and Run-Length Smoothing Algorithm (RLSA) [5]. These classical methods are efficient and interpretable, but degrade under handwriting variability, skew, low contrast, and complex documents. Modern approaches treat word (or text)

segmentation as dense prediction. Binary quadratic approaches can calculate inter- and intra-word gaps between words using support vector machines [6], scale space techniques, seam carving [7], and vertical projection [8].

Some deep learning approaches performed word segmentation with BLSTM-CTC [9], EAST, and DB (Differentiable Binarization) regress geometry/probability with a learnable threshold [10]. PSENet and PAN expand kernels (aggregate pixels) to separate adjacent instances [11-12]. PixelLink and SegLink predict link affinities among pixels/segments to isolate words [13]. End-to-end spotting systems unify detection and recognition and can produce instance masks for words [14].

In [15], domain-specific keywords were identified in medical prescriptions using a hidden Markov model. In [16], an edge detection Sobel operation was implemented in a CNN to detect and identify words in prescriptions. In [17], a transformer-based multimodal model, encompassing stroke enhancement, was implemented to classify the medicine type. Compared to these approaches, this study targets semantic entities (e.g., Instructions, Dose strength) rather than generic word instances within dense, fine-print, and sometimes handwritten prescriptions. Generic word segmentation can over-fragment long instruction lines or miss domain semantics. The proposed framework complements these tasks by using YOLO-based instance segmentation with Slicing Aided Hyper Inference (SAHI) and Block-Aware Routing (BAR) to recover small/dense entities and return exactly one entity per class per block, aligning with downstream OCR/EHR integration needs. Recently, the introduction of YOLOv11 has demonstrated architectural refinements that further improve precision and convergence dynamics for dense prediction tasks [18].

Medical prescription analysis poses unique demands that have not been fully met by these baselines. First, small and densely packed entities, such as dosage, frequency, or instructions, are often missed at standard input resolutions due to scale trade-offs. Second, duplicate predictions can arise when similar text segments appear repeatedly across blocks. To address these issues, two complementary strategies have emerged in related domains: sliced inference, where images are partitioned into overlapping tiles to boost recall for small objects, and post-hoc merging, which consolidates duplicate detections across tiles [19]. However, tiling alone can harm precision by producing redundant predictions, although it has been effective in aerial imagery [20] and medical imaging [21]. Naive tiling introduces duplicate detections across slices, which can reduce precision. Merging algorithms, such as NMS [22], Soft-NMS, and Weighted Boxes Fusion (WBF) [23], have been employed to reconcile overlaps, but often rely on fixed heuristics. Prescription parsing benefits from SAHI, as dense instructions and dosage fields occupy small regions that are easily missed by global detectors. A related stream of work focuses on routing and deduplication of overlapping predictions. More recently, task-specific routers have been applied in document extraction, where a prediction is enforced per class per block to align with the semantic structure [24]. Such strategies align with the proposed block-aware routing: by combining baseline YOLO predictions with tiled SAHI outputs and enforcing a one-entity-per-class constraint, duplication can be mitigated while improving the completeness of prescription blocks.

This motivated the current work, wherein a lightweight and deployment-friendly pipeline integrates SAHI-style tiled inference with a block-aware routing mechanism. The routing step enforces a one-entity-per-class-per-block policy, suppressing duplicates while retaining recall gains. The systematic evaluation of the approach uses both YOLOv8 and YOLOv11 backbones, demonstrating improved recall and mean Average Precision (mAP) on a custom prescription image dataset. The contributions of this study are threefold:

- Benchmarks YOLOv8 and YOLOv11 for handwritten medical prescription segmentation and analyzes their performance trade-offs.
- Augmented inference with SAHI tiling shows consistent recall improvements on small and dense classes.
- Invokes a block-aware routing phase that deduplicates predictions per semantic block, yielding valid output suitable for downstream OCR/EHR integration.

II. PROPOSED METHODOLOGY

The proposed framework integrates a baseline instance segmentation backbone with a tiling-based enhancement strategy and a block-aware routing mechanism. The system is designed to operate on scanned prescription pages and deliver structured, deduplicated entity crops for downstream recognition tasks. Figure 1 illustrates the overall pipeline, which builds upon a YOLO-based segmentation backbone augmented with inference tiling and block-wise routing. The goal is to maximize recall for dense, small-text fields while preserving precision and delivering one consistent entity per semantic class per block. The approach is split into four stages: detection backbone, tiled SAHI inference, block-wise routing, and export.

A. Problem Definition

Let a prescription image be $I \in \mathbb{R}^{H\times W\times 3}$ with ground-truth instances

$$y = \{(b_j, m_j, c_j)\}_{j-1}^{N} \quad (1)$$

where b_j is a bounding box, m_j is an instance mask, and $c_j \in \mathcal{C}$ denotes the class label among eight categories: Block id, Med Name, Med Type, Dose strength, Frequency, Duration, Quantity, and Instructions. The detector must predict:

$$\hat{Y} = \{(\hat{b}_k, \hat{m}_k, \hat{c}_k, \hat{s}_k)\}_{k=1}^{N} \quad (2)$$

such that recall and mask coverage are maximized, while enforcing at most one entity per class per block.

B. Dataset Description

The custom dataset used in this study consists of Region of Interest (RoI) images (Body region) extracted from handwritten prescription templates collected from multiple healthcare organizations [25]. These prescriptions reveal substantial variability in layout, handwriting style, and spatial resolution,

reflecting real-world clinical documentation. The extracted RoIs undergo a standardized preprocessing pipeline that includes normalization and removal of noise, skew, and line artifacts, resulting in a grayscale representation. The dataset comprises 1040 images with varied resolutions and content, enabling robust evaluation under realistic conditions. Currently, the dataset has not been made publicly available as it will be utilized in subsequent research activities. However, it will be made available upon request in the future.

For each image in the dataset, manual annotations were created in Label Studio using a predefined class taxonomy, with eight distinct classes or text regions: Med_Name, Med_Type, Frequency, Dose_Strength, Duration, Instructions, Quantity, and Block_id. The related entities belonging to the same prescription entry were grouped under a shared Block_id. This imposes layout-level consistency, ensures that semantically related fields are treated as a coherent structural unit, and block coherence maintains consistent spatial grouping across annotations. All annotations were visually verified and iteratively refined to ensure consistency and correctness across the dataset. Figure 2 illustrates an example handwritten prescription image along with its corresponding annotations. The left panel shows the original handwritten prescription input, while the right panel represents the annotated regions, highlighting different semantic classes such as Med_Name, Frequency, Instructions, and their grouping through the Block_id class. Each shaded region corresponds to a distinct annotated class, and the blocks are visually grouped to represent individual prescription entries.

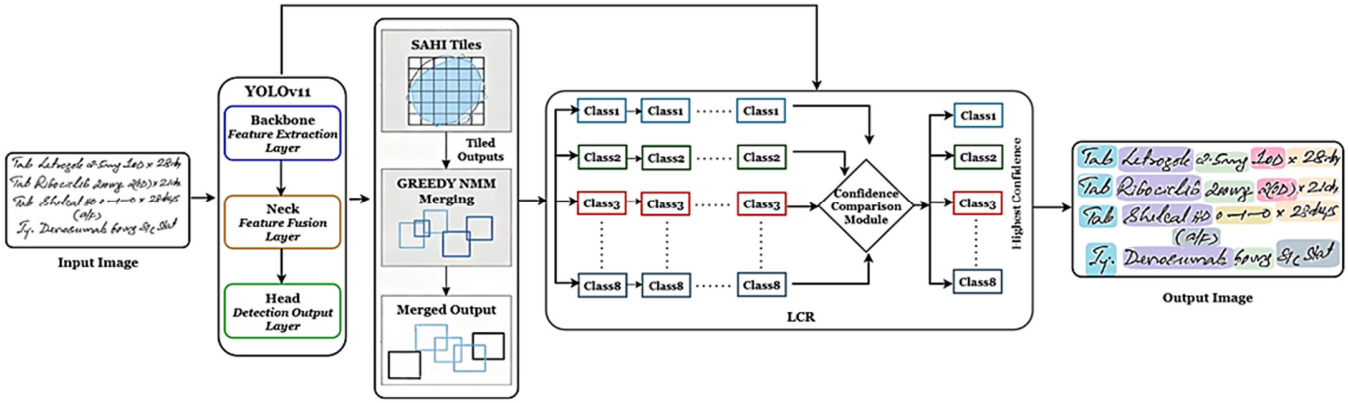

Fig. 1. End-to-end architecture of the proposed YOLO model

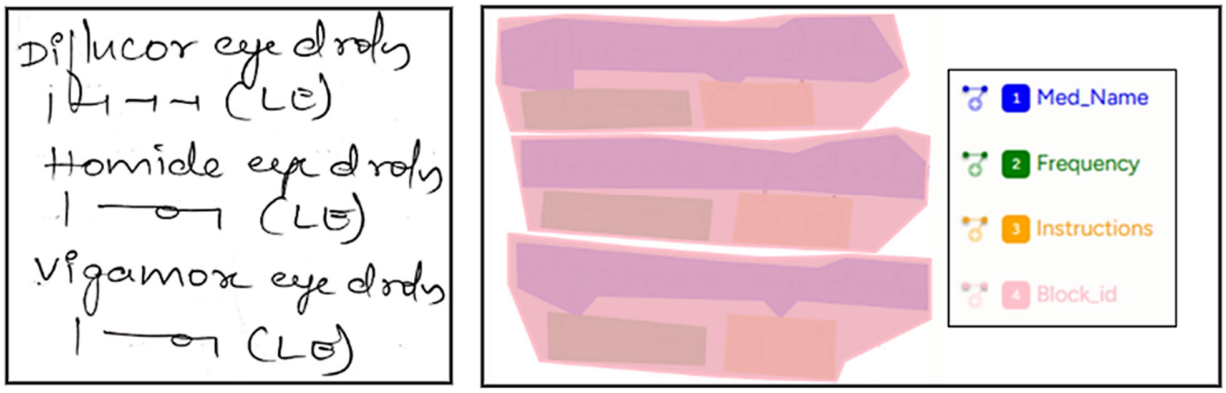

Fig. 2. Sample data with its corresponding annotation

C. Baseline YOLO

The baseline employs a YOLO-based segmentation detector (YOLOv8 or YOLOv11), trained on the annotated prescription dataset with eight semantic classes. A single forward pass is carried out at resolution 896×896, producing per-instance boxes and prototype-driven masks. This module excels in recognizing medium-to-large entities such as Med_Name and Quantity, offering low latency. The proposed model adopts the YOLOv8m-seg and YOLOv11m-seg architectures as baselines. These models employ CSP-based backbones and PAN/FPN necks for feature aggregation, trained with multi-task losses:

$$L = \lambda_{box} L_{CIoU} + \lambda_{cls} L_{cls} + \lambda_{mask}(L_{BCE} + L_{Dice}) \quad (3)$$

Training is performed with AdamW, cosine learning rate decay, and augmentations that preserve text readability (copy-paste, mosaic, HSV jitter, mild perspective).

D. Slicing Aided Hyper Inference (SAHI)

To address the under-detection of small or densely packed text, the system incorporates a tiling strategy via SAHI. The image is partitioned into overlapping tiles (896×896 with 35% overlap), and the detector is applied independently on each tile.

This resolution preserves the details of fine handwritten strokes by limiting the number of slices per page so that small strokes and character details remain well represented for the detector, and 35% overlap ensures that the words near tile boundaries are fully captured at least in one of the slices. Large slices reduce their total number but increase the per-slice computation and GPU memory. Predictions are remapped to global coordinates and merged through operators such as NMS, Soft-NMS, Greedy-NMM, or Weighted Boxes Fusion (WBF). This path improves recall for classes like Dose_strength, Duration, and Instructions.

$$T(I;\ S, o) = \{\ t_k\ \}_{k=1}^{N} \quad (4)$$

where S and o are the tile size and fractional overlap.

E. Block-Aware Routing (BAR)

Since both streams may produce duplicates or conflicting predictions, a Lowest-Cost Routing (LCR) module is introduced. Each prescription page is segmented into blocks (via Block_id), and candidates from both paths are grouped per block and per class. Within each block, the router selects at most one candidate per class based on a utility function that balances confidence score, spatial alignment, and overlap penalty. This ensures a deduplicated, one-entity-per-class layout. Both baseline and tiled predictions are combined through an LCR scheme. For each detected block B, candidate detections are grouped by class. A composite utility is defined as:

$$u(\ d\ \mid\ B\ ) =$$
$$\alpha \cdot s_d - \beta \cdot dist(b_d, B) - \gamma \cdot conflict(d, B) \quad (5)$$

where s_d is the confidence score, $dist$ is the normalized spatial distance, and $conflict$ penalizes overlaps with high-priority classes. The router selects

$$\hat{d}_c(B) =$$
$$argmax(d \in \mathcal{D}_B, c_d = c)u(d \mid B) \quad (6)$$

to ensure one entity per class per block.

F. Exports and Visualizations

The routed predictions are exported in two forms:

- Full-page overlays with color-coded class outputs for human validation.
- Class-specific crops saved in separate folders (e.g., Med_Type, Instructions).

This modular design allows flexibility, and practitioners can choose baseline-only crops for reliability, tiled-only crops for recall, or routed outputs for balanced performance.

The proposed method improves recall on small-text fields by tiling, preserves precision via routing, and produces clean and deduplicated outputs suitable for real-world deployment. Compared to baseline YOLO, the combination of YOLO +SAHI+BAR balances recall and precision while standardizing outputs for block-structured recognition.

III. RESULTS

The baseline YOLO model was trained on an NVIDIA T4 GPU using a fixed input resolution of 896×896 pixels, a batch size of 2, a learning rate of 3×10^{-4}, and optimized with the AdamW optimizer. This resolution preserves fine stroke-level details of handwritten words while limiting the number of inference tiles per page. For slice-based inference, SAHI was employed with a slice size of 896×896 and 35% overlap, ensuring that the words crossing the slice boundaries completely appear inside at least one slice. The custom dataset comprises 1040 images with different spatial resolutions, reflecting real-world document variability. During evaluation, both grayscale and color image representations were examined to assess the robustness of the model under distinct input conditions. During inference, the merging parameters and confidence thresholds were kept stable with $confidence \in [0.20 - 0.25]$, and NMS IoU of 0.55, and performance was measured using standard metrics such as Box mAP@50, Box mAP@50:95, Mask mAP@50, Mask mAP@50:95, and Recall@0.5. To ensure fair and consistent comparison, all experiments were trained and evaluated using similar hyperparameter settings.

Table I summarizes the performance of the baseline YOLOv11-seg model over different training-validation data split ratios, such as 50:50, 60:40, 70:30, and 80:20. The evaluation metrics measure the detection and segmentation accuracy across various IoU thresholds. The results demonstrate that the 70:30 ratio yielded the most balanced and notable performance. At this ratio, the baseline YOLOv11-seg achieved the highest Box mAP50 (0.932), Box mAP50:95 (0.715), Mask mAP50 (0.934), and Mask mAP50:95 (0.648), with a peak Recall@0.5 of 0.929. This configuration provides an optimal balance between the amount of training data and the model's ability to generalize adequately to unseen samples. The persistent high recall and precision indicate that the model can accurately localize and segment instances across diverse prescription templates without significant false negatives.

TABLE I. PERFORMANCE COMPARISON OF BASELINE YOLOV11-SEG ACROSS VARIED TRAIN AND VALIDATIO SPLITS

Ratio	Box mAP50	Box mAP50:95	Mask mAP50	Mask mAP50:95	Recall @0.5
50:50	0.905	0.673	0.822	0.556	0.831
60:40	0.916	0.680	0.838	0.564	0.837
70:30	0.932	0.715	0.934	0.648	0.929
80:20	0.920	0.690	0.884	0.590	0.854

In contrast, the 50:50 and 60:40 splits indicate comparatively lower performance across all metrics, likely due to insufficient training data, which restricts the model's ability to learn robust spatial and contextual representations. Although the 80:20 split offers a slightly higher training volume, it shows a diminished generalization performance, with a decrease in Mask mAP50:95 (0.59) and Recall@0.5 (0.854). This suggests a mild overfitting, where the model becomes more tuned to the training data distribution and less effective on validation samples.

A. Ablation Study

Table II describes a comparison of YOLOv8 and YOLOv11 models, with and without integrating the proposed SAHI and BAR modules. SAHI merging thresholds were tuned by sweeping the IoU and confidence thresholds of baseline and SAHI over the validation split and selecting the setting with the best recall without inflating duplicate detections. The study evaluated with an $IoU \in [0.4 - 0.6]$ and $confidence \in [0.20 - 0.55]$ and selected the thresholds that yielded the best balance in retaining true positives. This tuning helps the merging stage by preserving high recall and maintaining stable precision.

TABLE II. PERFORMANCE COMPARISON OF MODEL WITH SAHI TILING + BAR

Method	Box mAP50	Box mAP50:95	Mask mAP50	Mask mAP50:95	Recall @0.5
YOLOv8	0.925	0.703	0.925	0.640	0.913
YOLOv8+ SAHI +BAR	0.927	0.707	0.93	0.642	0.926
YOLOv11	0.932	0.715	0.934	0.648	0.929
YOLOv11+ SAHI +BAR	**0.934**	**0.718**	**0.936**	**0.651**	**0.938**

YOLOv11 outperformed YOLOv8 across all metrics, confirming the remarkable feature extraction and spatial representation capabilities of the newer backbone. When incorporating the SAHI and BAR modules, performance was further improved, particularly in the YOLOv11 framework. The YOLOv11+SAHI+BAR configuration achieved the highest values across all parameters, with Box mAP50 of 0.934, Box mAP50:95 of 0.718, Mask mAP50 of 0.936, Mask mAP50:95 of 0.651, and Recall@0.5 of 0.938. These persistent gains demonstrate that the SAHI module improves localization precision and segmentation by enhancing the model's ability to capture discriminative spatial and contextual features, while BAR selects at most one candidate per class based on a utility function that balances confidence score and overlap penalty. The enhancements are evident in the finer-grained metrics (mAP50:95 and Recall), which highlight the benefit of the SAHI and BAR modules in refining multi-scale features and improving overall generalization.

Table III presents the per-class Recall@IoU0.50 comparison, offering a more intricate perspective of SAHI+BAR modules, which influence individual class performance. Among all classes, Block-id achieves the highest recall (up to 0.994), while Instructions remains the most challenging category due to its dense and complex text structure and few samples. The YOLOv11+SAHI+BAR configuration consistently achieved the best per-class results, showing a distinct improvement in Frequency (0.977), Med-Name (0.966), Dose-strength (0.894), and Duration (0.892). The overall recall rose from 0.913 (YOLOv8) to 0.938 (YOLOv11+SAHI+BAR), confirming the reliability of the proposed enhancement in segmenting different and fine-grained entities. These results demonstrate that YOLOv11 provides a stronger baseline, and the SAHI and BAR modules further refine class-specific sensitivity and improve model robustness across all prescription parameter categories, making it a superior configuration for segmentation tasks.

TABLE III. PER-CLASS RECALL@IOU_0.5

Class	YOLOv8	YOLOv8+ SAHI +BAR	YOLOv11	YOLOv11+ SAHI +BAR
Block_id	0.990	0.992	0.993	0.994
Dose_strength	0.872	0.876	0.882	0.894
Duration	0.870	0.878	0.880	0.892
Frequency	0.961	0.968	0.971	0.977
Instructions	0.774	0.835	0.839	0.847
Med_Name	0.947	0.952	0.961	0.966
Med_Type	0.953	0.955	0.963	0.975
Quantity	0.936	0.948	0.950	0.955
Overall	**0.913**	**0.926**	**0.929**	**0.938**

B. Quantitative Analysis

Table IV shows a quantitative evaluation of the proposed YOLOv11+SAHI+BAR model against different widely adopted advanced object detection and instance segmentation frameworks, including Mask-RCNN [26], Mask-DETR [27], and EfficientDet-D3 [28]. The comparison was performed across the evaluation metrics Box mAP50, Box mAP50:95, Mask mAP50, Mask mAP50:95, and Recall@0.5. Considering detection accuracy, YOLOv11+SAHI+BAR attained the highest box mAP50 of 0.934, which is an absolute improvement of 0.016 over EfficientDet-D3 (0.918), 0.018 over Mask-RCNN (0.916), and 0.052 over DETR (0.882). Similarly, in the metric Box mAP50:95, YOLOv11+SAHI+BAR attained 0.718, outperforming EfficientDet-D3 (0.689) by 0.019, Mask-RCNN (0.691) by 0.027, and DETR (0.689) by 0.029. These gains indicate the stronger capability of the proposed model to maintain detection precision across varying object scales. In addition, the confidence score was increased to 0.90, 0.85, and 0.78 with the proposed IoU thresholds.

For segmentation performance, the proposed YOLO+SAHI+BAR model yielded a Mask mAP50 of 0.936, showing improvements of 0.016 and 0.042 compared to Mask-RCNN and Mask-DETR, respectively. As EfficientDet-D3 supports only bounding box segmentation, mask-related metrics cannot be performed. Similarly, Mask mAP50:95 improved to 0.651, exceeding Mask-RCNN by 0.018 and DETR by 0.042. These consistent improvements across both box- and mask-level metrics indicate that the YOLOv11+SAHI+BAR model attained more refined segmentation boundaries and maintained superior accuracy even under higher IoU thresholds. In terms of recall, YOLOv11+SAHI+BAR attained a Recall@0.5 of 0.938, marking improvements of 0.018 over EfficientDet-D3 (0.92), 0.024 over Mask R-CNN (0.914), and 0.038 over DETR (0.90). These results highlight the model's sensitivity and reduced false-negative rate in identifying all relevant regions.

TABLE IV. PERFORMANCE COMPARISON OF THE PROPOSED WITH OTHER ADVANCED MODELS

Method	Box mAP50	Box mAP50:95	Mask mAP50	Mask mAP50:95	Recall @0.5
Mask R-CNN	0.916	0.691	0.92	0.633	0.914
Mask DETR	0.882	0.658	0.894	0.609	0.900
EfficientDet -D3	0.918	0.689	-	-	0.920
Proposed	**0.934**	**0.718**	**0.936**	**0.651**	**0.938**

Fig. 3. Predicted classes from the baseline and the proposed model.

Figure 3 shows how SAHI and BAR improve the confidence scores of some classes, such as Dose_strength and Duration, with the YOLOv11 baseline. The green color bounding boxes represent detected classes from the baseline YOLOv11, and the red color bounding boxes represent the detected classes from YOLOv11+SAHI+BAR, along with confidence scores. The confidence scores for Dose_strength and Duration are 0.84 (in Block1) and Duration is 0.73 (in Block2) from the baseline model, but increase to 0.90, 0.85, and 0.78, respectively, with the proposed model.

C. Efficiency

Table V reports Parameters, GFLOPs, and FPS for the different models on a T4 NVIDIA GPU. YOLOv8m-seg and YOLOv11m-seg exhibit remarkable efficiency with relatively fewer parameters, 27.3 M and 22.4 M, respectively, compared to Mask-RCNN (44.2 M) and DETR (41.3 M). Although YOLOv11m-seg shows slightly higher computational demands (113.2 GFLOPs) than YOLOv8m-seg (110.2 GFLOPs), the increase is justified by enhanced architectural refinements to improve segmentation accuracy.

The inference throughput of YOLOv11m-seg achieves substantially higher performance, with 2.14× the FPS of Mask-RCNN while engaging 27% fewer GFLOPs, and 2.73× the FPS of DETR with 22% fewer GFLOPs. These results demonstrate the efficiency of the YOLOv11m design, which offers improved feature extraction and optimized decoding mechanisms. While EfficientDet-D3 shows the lowest parameter count (12 M) and reduced GFLOPs (25), it achieves relatively lower FPS (24), showing that theoretical reductions in FLOPs do not always correspond to practical runtime gains. This difference reinforces the influence of hardware-specific factors such as memory throughput, kernel optimization, and post-processing latency.

The YOLOv11m-seg provides an optimal trade-off between computational complexity and inference speed. Its balance of low parameter counts, moderate GFLOPs, and high FPS renders it highly efficient for real-time medical prescription parameter segmentation tasks, where both accuracy and latency are crucial for practical deployment.

TABLE V. EFFICIENCY COMPARISON OF DIFFERENT MODELS

Method	Params (M)	GFLOPs	FPS
YOLOv8m-seg	27.3	110.2	3.6
YOLOv11m-seg	22.4	113.2	3.5
Mask R-CNN	44.2	134.2	5
DETR	41.3	86	108
EfficientDet-D3	12	25	24

D. Error Analysis

Validation errors were systematically examined and categorized into four primary types, as shown in Table VI. The most frequent concern observed was small-font misses in dense regions, which could be reduced by increasing the local resolution, although a few residual misses persist in extreme clutter. SAHI improves the detection of each word by processing it into a high-resolution slice and preserving fine handwritten details that are lost in downsampling. In addition, preserved pixel and stroke information reduces missed detections in densely written and small-font regions. Slice overlaps ensure that the words near the boundary are captured at least in one slice, mitigating the boundary induced false negatives. Tile-boundary fragments or duplicates were arising primarily from overlapping artifacts and were effectively mitigated using a block-aware routing mechanism. Class confusion errors occasionally appear due to contiguous block strings, leading to misclassification among visually similar components. Block misassignment errors were identified when a valid region was erroneously assigned to an incorrect block, often resulting from spatial proximity or alignment ambiguity. The detections produced from overlapping slices were merged into a global coordinate space without layout level constraint. The slice-based inference lacks global contextual cues like inter-block spacing, resulting in ambiguous block labeling when words lie close to structural borders. This inconsistent prediction across adjacent slices causes incorrect detection associated with a neighboring block in the merging stage. To address this, the block-aware post-hoc employs region-specific non-maximum suppression with Hungarian matching to resolve ambiguities and reduce cross-block confusion. Collectively, these error findings highlight the dominant sources of schema-compliance and recall deviations in the validation phase.

TABLE VI. VALIDATION ERROR BREAKDOWN

Category	Observations
Small-font miss in dense regions	Reduces by increasing local resolution; residual misses persist in extreme clutter.
Tile-boundary duplicate/fragment	The block-aware router suppresses the overlap artifacts.
Class confusion	Contiguous block strings
Block misassignment	Valid fields attached to wrong blocks.

IV. DISCUSSION

The experiments aimed to enhance the segmentation of structured prescription components from prescription images by augmenting a strong YOLO-based segmentation baseline with inference-time tiling (SAHI), a lightweight routing mechanism (BAR), and the integration of a more recent YOLOv11 backbone.

For fields such as Instructions and Duration, where small fonts and crowded layouts are common, tiling with GreedyNMM merging produced recall gains of +5.7 pp and +1.3 pp, respectively. This confirms the hypothesis that fixed-resolution global inference struggles with fine-grained tokens, while tiling increases the effective resolution seen by the detector. However, tiling alone occasionally introduced duplicates, particularly near tile boundaries.

The routing stage (best-of-both) preserved the cleaner outputs of the baseline by enforcing a one-entity-per-class-per-block constraint. By prioritizing the highest-utility candidate per class, routing mitigated duplication while retaining the recall benefits of tiling. This was especially effective for Instructions and Quantity, where multiple overlapping predictions would otherwise reduce mAP.

Compared to YOLOv8, the YOLOv11 backbone introduces lightweight attention mechanisms and structural refinements, leading to improved mAP50:95 and more stable training dynamics. However, inference-time tiling and routing remained the dominant contributors to recall improvements, suggesting that architectural advances alone do not fully resolve small-object challenges. Tiling increases the inference time approximately 2-3 times compared to single-pass baseline inference, as each document requires multiple overlapping slices. Although routing adds negligible cost, deployment in latency-sensitive environments may require adaptive tiling strategies.

Box-level performance remained consistently stronger than mask-level metrics. This is expected given the complexity of polygon annotations for thin, elongated fields (e.g., Instructions), where minor boundary deviations cause larger IoU penalties. Future work may integrate refinement modules (e.g., CRF or SAM-based boundary correction) to improve mask fidelity. Improvements were concentrated in classes with small or dense instances. For larger, well-separated classes (e.g., Med_Type, Block_id), the baseline already performed strongly, and tiled inference offered limited gains. This imbalance points to opportunities for class-specific thresholds or adaptive routing strategies.

V. CONCLUSION

Medical prescription documents remain challenging for automated information extraction due to dense layouts, small handwritten text, and highly varied field structures, underscoring the need for robust and block-aware segmentation strategies. Existing methods rely on architectural modifications and multi-stage detector-based models to improve the performance at the cost of increased training complexity and deployment overhead. This work presents a modular and inference-efficient framework for structured segmentation of medicine and its parameters from handwritten prescription documents by using YOLO-based segmentation with SAHI tiling that consistently improves recall in dense and small-font regions without compromising the precision, with BAR further resolving the cross-tile duplication and block misassignment, producing stable one-entity-per-class output that has not been explicitly explored in prior prescription parsing or document segmentation studies, being better suited for downstream OCR and archival workflows.

The proposed model was quantitatively compared against various established object detection and instance segmentation models, outperforming existing methods by achieving Box mAP50 of 0.934, Mask mAP50 of 0.936, and Recall of 0.938. Overall, this approach demonstrates a practical balance between accuracy and deployment efficiency, showing that inference-side strategies, combined with light-weight routing logic, can substantially enhance structured document parsing pipelines in real-world scenarios.

REFERENCES

[1] J. Memon, M. Sami, R. A. Khan, and M. Uddin, "Handwritten Optical Character Recognition (OCR): A Comprehensive Systematic Literature Review (SLR)," *IEEE Access*, vol. 8, pp. 142642–142668, 2020, https://doi.org/10.1109/ACCESS.2020.3012542.

[2] J. Rausch, O. Martinez, F. Bissig, C. Zhang, and S. Feuerriegel, "DocParser: Hierarchical Document Structure Parsing from Renderings," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 35, no. 5, pp. 4328–4338, Vancouver, Canada, May 2021, https://doi.org/10.1609/aaai.v35i5.16558.

[3] I. Sanasam, P. Choudhary, and K. M. Singh, "Line and word segmentation of handwritten text document by mid-point detection and gap trailing," *Multimedia Tools and Applications*, vol. 79, no. 41, pp. 30135–30150, Nov. 2020, https://doi.org/10.1007/s11042-020-09416-1.

[4] Y. Qian, E. Santus, Z. Jin, J. Guo, and R. Barzilay, "GraphIE: A Graph-Based Framework for Information Extraction," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, MN, USA, Mar. 2019, pp. 751–761, https://doi.org/10.18653/v1/N19-1082.

[5] M. Javed, P. Nagabhushan, and B. B. Chaudhuri, "A direct approach for word and character segmentation in run-length compressed documents with an application to word spotting," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, Tunis, Tunisia, Dec. 2015, pp. 216–220, https://doi.org/10.1109/ICDAR.2015.7333755.

[6] B. Kada, A. Mohammed, and B. Abdelmajid, "An Optimized Approach for Handwritten Arabic Character Recognition based on the SVM Classifier," *Engineering, Technology & Applied Science Research*, vol. 15, no. 2, pp. 22232–22238, Apr. 2025, https://doi.org/10.48084/etasr.9292.

[7] M. Das and M. Panda, "Seam carving, horizontal projection profile and contour tracing for line and word segmentation of language independent

handwritten documents," *Results in Engineering*, vol. 18, June 2023, Art. no. 101110, https://doi.org/10.1016/j.rineng.2023.101110.

[8] S. Kaur, S. Bawa, and R. Kumar, "Heuristic-based text segmentation of bilingual handwritten documents for Gurumukhi-Latin scripts," *Multimedia Tools and Applications*, vol. 83, no. 7, pp. 18667–18697, Feb. 2024, https://doi.org/10.1007/s11042-023-15335-8.

[9] C. Vinotheni and S. L. Pandian, "Fast Recurrent Neural Network with Bi-LSTM for Handwritten Tamil Text Segmentation in NLP," *ACM Transactions on Asian Low-Resource Language Information Processing*, vol. 23, no. 5, Feb. 2024, Art. no. 68, https://doi.org/10.1145/3643808.

[10] M. Liao, Z. Zou, Z. Wan, C. Yao, and X. Bai, "Real-Time Scene Text Detection With Differentiable Binarization and Adaptive Scale Fusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 919–931, Jan. 2023, https://doi.org/10.1109/TPAMI.2022.3155612.

[11] W. Wang *et al.*, "Shape Robust Text Detection With Progressive Scale Expansion Network," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA, June 2019, pp. 9328–9337, https://doi.org/10.1109/CVPR.2019.00956.

[12] W. Wang *et al.*, "Efficient and Accurate Arbitrary-Shaped Text Detection With Pixel Aggregation Network," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Korea (South), July 2019, pp. 8439–8448, https://doi.org/10.1109/ICCV.2019.00853.

[13] D. Deng, H. Liu, X. Li, and D. Cai, "PixelLink: Detecting Scene Text via Instance Segmentation," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018, https://doi.org/10.1609/aaai.v32i1.12269.

[14] P. Lyu, M. Liao, C. Yao, W. Wu, and X. Bai, "Mask TextSpotter: An End-to-End Trainable Neural Network for Spotting Text with Arbitrary Shapes." *in Proceedings European Conference Computer Vision (ECCV)*, Munich, Germany, 2018, pp. 67–83.

[15] A. Mukhejee, A. Halder, S. Nath, and S. K. Sarkar, "A New Approach to Information Retrieval Based on Keyword Spotting from Handwritten Medical Prescriptions," *Advances In Industrial Engineering And Management*, vol. 6, no. 2, 2017.

[16] E. Hassan, H. Tarek, M. Hazem, S. Bahnacy, L. Shaheen, and W. H. Elashmwai, "Medical Prescription Recognition using Machine Learning," in *2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*, Jan. 2021, pp. 0973–0979, https://doi.org/10.1109/CCWC51732.2021.9376141.

[17] J. Zia, U. Habib, and M. A. Naeem, "Extraction and Classification of Medicines from Handwritten Medical Prescriptions," in *2023 18th International Conference on Emerging Technologies (ICET)*, Peshawar, Pakistan, Aug. 2023, pp. 104–109, https://doi.org/10.1109/ICET59753.2023.10374771.

[18] G. Jocher, J. Qiu, and A. Chaurasia, "Ultralytics YOLO." Jan. 2023, Available: https://github.com/ultralytics/ultralytics.

[19] F. C. Akyon, S. O. Altinuc, and A. Temizel, "Slicing Aided Hyper Inference and Fine-Tuning for Small Object Detection," in *2022 IEEE International Conference on Image Processing (ICIP)*, Bordeaux, France, July 2022, pp. 966–970, https://doi.org/10.1109/ICIP46576.2022.9897990.

[20] H. Zhang, C. Hao, W. Song, B. Jiang, and B. Li, "Adaptive Slicing-Aided Hyper Inference for Small Object Detection in High-Resolution Remote Sensing Images," *Remote Sensing*, vol. 15, no. 5, Feb. 2023, https://doi.org/10.3390/rs15051249.

[21] G. A. Reina, R. Panchumarthy, S. P. Thakur, A. Bastidas, and S. Bakas, "Systematic Evaluation of Image Tiling Adverse Effects on Deep Learning Semantic Segmentation," *Frontiers in Neuroscience*, vol. 14, Feb. 2020, https://doi.org/10.3389/fnins.2020.00065.

[22] M. Gong, D. Wang, X. Zhao, H. Guo, D. Luo, and M. Song, "A review of non-maximum suppression algorithms for deep learning target detection," in *Seventh Symposium on Novel Photoelectronic Detection Technology and Applications*, Kunming, China, Mar. 2021, vol. 11763, pp. 821–828, https://doi.org/10.1117/12.2586477.

[23] R. Solovyev, W. Wang, and T. Gabruseva, "Weighted boxes fusion: Ensembling boxes from different object detection models," *Image and Vision Computing*, vol. 107, Mar. 2021, Art. no. 104117, https://doi.org/10.1016/j.imavis.2021.104117.

[24] A. Banerjee, S. Biswas, J. Lladós, and U. Pal, "SwinDocSegmenter: An End-to-End Unified Domain Adaptive Transformer for Document Instance Segmentation," in *Document Analysis and Recognition - ICDAR 2023*, San José, CA, USA, 2023, pp. 307–325, https://doi.org/10.1007/978-3-031-41676-7_18.

[25] G. R. Rekha and S. Siddesha, "Categorization and Content Extraction in Medical Prescription Using YOLOv8," in *Emerging Electronics and Automation, Volume 1*, vol. 1455, M. Kankanhalli, S. Bhartiya, and P. S. Pravin, Eds. Springer Nature Singapore, 2026, pp. 419–428, https://link.springer.com/chapter/10.1007/978-981-96-9554-6_33.

[26] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proceedings of the IEEE International Conference on Computer Vision*, Venice, Italy, 2017, pp. 2961–2969.

[27] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-End Object Detection with Transformers," in *Computer Vision – ECCV 2020*, Glasgow, UK, 2020, pp. 213–229, https://doi.org/10.1007/978-3-030-58452-8_13.

[28] M. Tan, R. Pang, and Q. V. Le, "EfficientDet: Scalable and Efficient Object Detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Seattle, WA, USA, 2020, pp. 10781–10790.