SahiKid: An Efficient SAHI-based Framework for Small Kidney Stone Detection in CT Scans

Tien-Dao Luu[1], Thanh Ma[1], Phu-An Thai[1], Phat-Dat Trinh[1], Van-Kha Phan[1], Chi-Thien Nguyen[2], Huu-Hoa Nguyen[1]
[1] College of Information and Communication Technology, Can Tho University, Vietnam
[2] Vietnam Posts and Telecommunications Group, Vietnam
Correspondence: Huu-Hoa Nguyen, nhhoa@ctu.edu.vn
Communication: received 30 June 2025, revised 28 August 2025, accepted 01 October 2025
Digital Object Identifier: https://doi.org/10.32913/mic-ict-research.v2025.n3.1398

Abstract: Kidney stone disease represents a prevalent urological condition, where timely detection—particularly of small calculi in computed tomography (CT) scans—is essential for effective clinical intervention and the mitigation of subsequent complications. Nevertheless, the reliable identification of such minute structures remains a considerable challenge owing to their diminutive size and the intricate surrounding anatomical context. To address this issue, we present a computationally efficient detection framework that integrates a lightweight object detection (OD) model with the Slicing Aided Hyper Inference (SAHI) methodology. Furthermore, we propose a novel algorithm, termed B-SAHI, designed to substantially accelerate inference. The framework is systematically evaluated on a publicly available CT dataset across multiple configurations, including full-image inference (FI), Slicing Aided Fine-tuning (SF), patch overlapping (PO), and kidney-specific region localization (KD). The optimal configuration, combining OD with SF, SAHI, PO, and KD, achieved a mean Average Precision at IoU threshold 0.5 (mAP50) of 78.66%, significantly surpassing the baseline of 0.05%. In terms of computational efficiency, B-SAHI reduced inference latency from 0.1553 to 0.0329 seconds, thereby enabling real-time applicability. These findings highlight the efficacy of slice-based inference and anatomically guided localization in advancing the detection of small-scale structures within medical imaging.

Keywords: *Kidney stones, Small object detection, CT images, SAHI, Medical image analysis, Deep learning*

I. Introduction

Nephrolithiasis or kidney stone disease poses a significant and growing global health burden, characterized by a high recurrence rate [1]. The condition involves the formation of mineral crystalloids in the urinary tract, potentially leading to severe complications including renal colic, obstruction, and irreversible renal impairment [1]. Non-contrast Computed Tomography (NCCT) is the established gold standard for diagnosis due to its high sensitivity and specificity in detecting calcifications [2].

A pivotal challenge is the detection of small nephroliths (≤ 5mm). Although sometimes asymptomatic, these stones are clinically significant as nidi for future growth and can precipitate excruciating renal colic. Their accurate identification is crucial for guiding clinical management, from active surveillance to minimally invasive procedures [3]. However, manual detection on NCCT scans is a demanding, time-consuming task fraught with potential for error due to reader fatigue, inter-observer variability, and the subtle presentation of small stones [3].

To mitigate these issues, Computer-Aided Diagnosis (CAD) systems leveraging deep learning have shown considerable promise [4]. While standard object detection architectures such as Faster R-CNN [5], YOLO [6], and SSD [7] have been applied to this problem [8], they exhibit a critical limitation: suboptimal performance on small objects [9]. This stems from the hierarchical nature of Convolutional Neural Networks (CNNs), where successive down-sampling operations reduce spatial resolution, thereby losing the fine-grained information necessary to identify small targets [10].

To address this specific and critical gap, we propose a novel, integrated deep learning framework designed explicitly to enhance the detection of small kidney stones. Our approach leverages the YOLOv11 architecture [11], a state-of-the-art model known for its exceptional balance of speed and accuracy. The core innovation of our work is the integration of OD with the Slicing Aided Hyper Inference framework (named SAHI)[12]. Namely, this framework is a generic methodology that improves *small object detection* [9] by partitioning high-resolution images into smaller, overlapping patches. Inference is performed on each patch, where small objects are now relatively larger and more discernible, before the results are stitched back together. This technique has shown remarkable success in other domains like satellite and aerial imagery, but its application to medical imaging, specifically for nephrolithiasis, remains underexplored. Our system further refines this process through a two-stage approach: an initial model first localizes the kidney region, thereby constraining the search space and reducing the likelihood of false positives. In particular, our main contributions of this paper are therefore threefold:

- We propose a novel two-stage framework that integrates the OD model with the SAHI technique for the specific and challenging task of small kidney stone detection in CT images.
- An algorithm named B-SAHI are introduced to reduce the inference time.
- We conduct a systematic evaluation that demonstrates the profound impact of our approach, showing a dramatic increase in mean Average Precision from a baseline of 0.05% to 78.66%.
- We provide empirical evidence that the combination of region-of-interest localization and slicing-aided inference is a highly effective strategy for overcoming the pervasive problem of small object detection in medical imaging analysis.

The remainder of this paper is structured as follows: We briefly present a fundamental background in Section 2. Next, Section 3 describes the proposed framework. Then, Section 4 introduces an B-SAHI algorithm. Section 5 presents the experiment and the results of the summary models. Finally, Section 6 shows the conclusions and future works.

II. Background

This section establishes the theoretical foundations upon which our proposed methodology is built. We first provide a formal overview of deep learning-based object detection, with a focus on the one-stage YOLO architecture. We then delve into the specific challenges of SOD (Small Object Detection) [13] and detail the SAHI framework designed to address these challenges. Subsequently, we formalize several ancillary techniques used to refine and enhance the SAHI pipeline. Finally, we introduce a proposed enhancement to accelerate SAHI's inference speed through batch processing.

1. Detecting the Small Object

Object detection is a computer vision task aiming to identify and localize objects in an image. Given an image I, the goal is to generate a set of predictions $\mathcal{B} = \{(c_i, b_i, s_i)\}_{i=1}^{N}$, where c_i is the class label, $b_i = [x, y, w, h]$ is the bounding box, and s_i is the confidence score for each detected object i. Deep Learning, especially with Convolutional Neural Networks (CNNs) that automatically learn hierarchical features, has become the dominant approach. Modern detectors are primarily categorized into two families: two-stage methods (e.g., Faster R-CNN) that first propose regions and then classify them, prioritizing accuracy, and single-stage methods that treat detection as a direct regression problem, prioritizing speed.

A persistent challenge is SOD, where an object occupies a very small fraction f of the image area (e.g., $f < 1\%$) [13]. Detecting small kidney stones in high-resolution CT scans is a quintessential example of this problem. The difficulty stems from two core issues inherent to standard CNN architectures [14]:

- **Loss of Spatial Resolution**: Successive pooling and striding operations in CNNs progressively reduce the feature map size, effectively erasing the fine-grained information required to identify small objects.
- **Weak Feature Representation**: With very few pixels, small objects produce features that are inherently less distinct, making them difficult to differentiate from background noise or other non-target structures.

2. Slicing Aided Hyper Inference

The Slicing Aided Hyper Inference framework is a powerful, model-agnostic technique designed to mitigate the challenges of SOD in high-resolution imagery. Its core principle is to divide the detection task on a large image into smaller, more manageable sub-problems on image patches, where small objects are relatively larger and more discernible. The framework comprises two key stages: fine-tuning and inference.

a) Slicing Aided Fine-Tuning:

To adapt a pre-trained model for SOD, SF is employed as a data augmentation strategy. Given a training dataset $\mathcal{D} = \{(I_i, B_i)\}_{i=1}^{M}$, where I_i is a high-resolution image and B_i is its set of ground-truth bounding boxes: (1) Each image I_i is partitioned into a set of smaller, potentially overlapping patches $\{P_{ij}\}_{j=1}^{K_i}$; The ground-truth annotations B_i are mapped to each patch, creating new annotation sets B'_{ij}; A new, augmented fine-tuning dataset $\mathcal{D}_{SF}$ is created, consisting of these patches and their annotations:

$$\mathcal{D}_{SF} = \bigcup_{i=1}^{M} \{(P_{ij}, B'_{ij})\}_{j=1}^{K_i} \tag{1}$$

Fine-tuning the model on $\mathcal{D}_{SF}$ forces it to learn features from objects at a higher effective resolution, significantly improving its capability to detect them.

b) Slicing Aided Inference:

During inference, SAHI applies a similar slicing strategy to a given query image I_{query}. The standard process is as follows: (1) *Slicing*: The image I_{query} is divided into a set of N patches $\{P_1, P_2, \ldots, P_N\}$; *Inference*: A detection model $\mathcal{M}$ is independently applied to each patch to obtain a set of predictions $\mathcal{B}_j = \mathcal{M}(P_j)$; *Shifting*: The coordinates of the predictions in each $\mathcal{B}_j$ are translated from the patch's local coordinate system to the original image's global coordinate system; The coordinates of the predictions in each $\mathcal{B}_j$ are translated from the patch's local coordinate system to the original image's global coordinate system. (3) *Merging*: All shifted predictions are aggregated. A post-processing step, typically Non-Maximum Suppression (NMS) [15], is applied to merge redundant bounding boxes, yielding the final prediction set $\mathcal{B}_{final}$.

This baseline procedure can be significantly enhanced with several ancillary techniques, as detailed in the following section.

3. Refinements for Sliced-Based Detection

The standard SAHI pipeline can be augmented with several techniques to improve both accuracy and efficiency. Our work incorporates three key refinements: Kidney Detection for ROI focusing, Overlapping Patches for robustness, and Full-Image Inference for large object retention.

a) Kidney Detection (KD) for ROI Focusing:

In medical imaging, objects of interest often reside within specific anatomical structures [16]. Instead of slicing the entire high-resolution image, a two-stage approach can be adopted where a preliminary model first localizes the broader Region of Interest. Let $\mathcal{M}_{KD}$ be a detector trained specifically to identify kidneys. For an input CT image I, this model produces a set of bounding boxes for the kidney regions: $\mathcal{B}_{KD} = \mathcal{M}_{KD}(I) = \{\mathbf{b}_{k_1}, \mathbf{b}_{k_2}, \ldots\}$ The subsequent SAHI slicing and stone detection processes are then applied only within the cropped image areas defined by each $\mathbf{b}_k \in \mathcal{B}_{KD}$. This focusing strategy dramatically reduces the search space, which not only accelerates computation but also significantly mitigates the risk of false positives arising from confounding structures in the background.

b) Overlapping Patches (PO):

To prevent objects located at the boundaries of slices from being missed or poorly detected, overlapping patches are utilized during both fine-tuning and inference. Let an image have dimensions $W \times H$, and the desired patch size be $w_p \times h_p$. Let r_w and $r_h \in [0, 1)$ be the overlap ratios for width and height, respectively. The step size, or stride, for generating subsequent patches is then defined as: $S_w = w_p \cdot (1 - r_w), S_h = h_p \cdot (1 - r_h)$ The set of patches $\{P_{ij}\}$ is generated by extracting regions of size $w_p \times h_p$ starting at coordinates $(i \cdot S_w, j \cdot S_h)$. This ensures complete coverage and provides multiple views of objects near patch edges, enhancing detection robustness.

c) Full-Image Inference (FI):

While slicing is highly effective for small objects, it can compromise the detection of large objects that may be bisected across multiple patches. To counter this, SAHI can optionally incorporate a parallel inference pass on the original, unsliced image. Let $\mathcal{B}_{SAHI}$ be the set of predictions from all sliced patches (after coordinate shifting) and $\mathcal{B}_{FI}$ be the predictions from the full image. The sets are combined before the final merging step:

$$\mathcal{B}_{combined} = \mathcal{B}_{SAHI} \cup \mathcal{B}_{FI} \tag{2}$$

The final, merged set of predictions is then obtained by applying NMS to this combined set:

$$\mathcal{B}_{final} = \text{NMS}(\mathcal{B}_{combined}) \tag{3}$$

This hybrid approach aims to leverage the strengths of both methods, preserving high accuracy for both small and large objects.

4. Proposed Batched Inference

A practical limitation of the standard SAHI process is its sequential nature. Applying the detection model $\mathcal{M}$ to each of the N slices one-by-one can be a performance bottleneck. To address this, we enhance the pipeline by incorporating *batched inference*.

This optimization modifies the inference step to leverage the parallel processing capabilities of modern GPUs.

1) **Slicing**: The input image I_{query} is divided into N patches $\{P_1, P_2, \ldots, P_N\}$.
2) **Batching**: The N patches are grouped into batches of a predefined size S. The k-th batch is denoted as $\mathbb{P}_k = \{P_{(k-1)S+1}, \ldots, P_{\min(kS,N)}\}$.
3) **Batched Inference**: Instead of sequential processing, the detection model performs inference on an entire batch of patches in a single forward pass.

$$\mathbb{B}_k = \mathcal{M}_{\text{batch}}(\mathbb{P}_k) \tag{4}$$

This significantly reduces computational overhead by maximizing GPU utilization.
4) **Merging**: The predictions from all batches are collected, shifted, and merged using NMS as in the standard approach.

By processing slices in batches, the total inference time is substantially reduced, making the highly accurate SAHI framework more practical for applications where both precision and speed are critical.

III. Proposal Framework

To address the significant challenges of SOD in medical imaging—namely, the loss of spatial resolution and weak feature representation inherent in standard models—we introduce a novel, two-stage framework. This framework is specifically engineered to enhance the detection of small kidney stones in high-resolution CT scans. Our approach establishes a hierarchical pipeline that first localizes the broader anatomical region of interest before deploying a specialized, slicing-based detection model. This strategy aims to maximize detection accuracy for minute targets by focusing computational resources and mitigating false positives. The architecture of our proposed framework is systematically illustrated in Figure 1 and detailed in the subsequent sections.

Model Preparation and Fine-Tuning: This initial, offline stage is dedicated to preparing the specialized deep learning models that form the core of our inference pipeline. The primary goal is to create models that are highly adapted to the specific nuances of the kidney stone detection task.

1) **Kidney Region Detector ($\mathcal{M}_{KD}$):** We begin by training a dedicated object detector, denoted as $\mathcal{M}_{KD}$, using the YOLOv11n architecture. This model is trained on a dataset manually annotated with bounding boxes encompassing the entire kidney regions. Its sole objective is to accurately localize the kidneys within a full CT scan. Upon successful training, for any given input image I, this model produces a set of bounding boxes $\mathcal{B}_{KD} = \mathcal{M}_{KD}(I)$ that define the spatial boundaries of the kidneys. This model serves as the critical first step in our inference pipeline to constrain the search space.
2) **Slicing Aided Fine-Tuning for the Stone Detector ($\mathcal{M}_{Stone}$):** Recognizing that standard object detectors pre-trained on general dataset are ill-suited for detecting minuscule kidney stones, we employ Slicing Aided Fine-Tuning. This process transforms our original training dataset of CT images, $\mathcal{D}$, into an augmented fine-tuning dataset, $\mathcal{D}_{SF}$, as formulated in Equation (1) of the background section. Specifically, each high-resolution training image is partitioned into a set of smaller, overlapping patches. The ground-truth annotations for kidney stones are then mapped to these new, smaller images. A second YOLOv11n model, which we denote as the stone detector $\mathcal{M}_{Stone}$, is then fine-tuned on this $\mathcal{D}_{SF}$ dataset. This procedure forces the model to learn the discriminative features of small stones at a much higher effective resolution, conditioning it to become an expert in identifying these challenging targets.

The Two-Stage Inference Pipeline: This stage represents the operational, online application of our framework, where the models prepared in Stage 1 are deployed to analyze a new, unseen CT scan. The pipeline proceeds sequentially as follows:

1) **Kidney Localization:** The input CT scan I is first processed by the pre-trained kidney detector $\mathcal{M}_{KD}$. This step yields the set of kidney region bounding boxes, $\mathcal{B}_{KD}$. This ROI-focusing strategy is paramount; it drastically reduces the search area, allowing the subsequent, more computationally intensive steps to operate only on relevant anatomical regions. This significantly enhances efficiency and reduces the likelihood of false positives arising from confounding structures (e.g., bones, calcified vessels) outside the kidneys.
2) **SAHI on Regions of Interest:** Instead of applying slicing to the entire image, our framework applies the SAHI protocol exclusively within each cropped kidney region defined by the bounding boxes in $\mathcal{B}_{KD}$. Each kidney region is partitioned into a set of N overlapping patches $\{P_1, P_2, ..., P_N\}$, employing the PO technique described in the background section to ensure robust coverage at patch boundaries.
3) **Batched Stone Detection:** The specialized stone detector, $\mathcal{M}_{Stone}$, performs inference on the generated patches. To optimize for speed and leverage modern hardware, we implement the **Batched Inference** enhancement. The N patches are grouped into batches $\mathbb{P}_k$, and the model performs inference on each batch in a single forward pass: $\mathbb{B}_k = \mathcal{M}_{batch}(\mathbb{P}_k)$, as defined in Equation (4) of the background section. This parallel processing significantly reduces the latency bottleneck associated with sequential inference.
4) **Prediction Merging and Final Output:** The predictions from all batches are collected. The coordinates of the detected bounding boxes within each patch are translated from their local coordinate systems back to the global coordinate system of the original input image I. Finally, all shifted predictions are aggregated. The final, consolidated set of stone detections $\mathcal{B}_{final}$ is obtained by applying NMS to this combined set, as formulated in Equation (3) of the background section, to merge redundant detections and eliminate low-confidence predictions. The framework's output is a precise set of bounding boxes identifying the location of small kidney stones on the original CT scan.

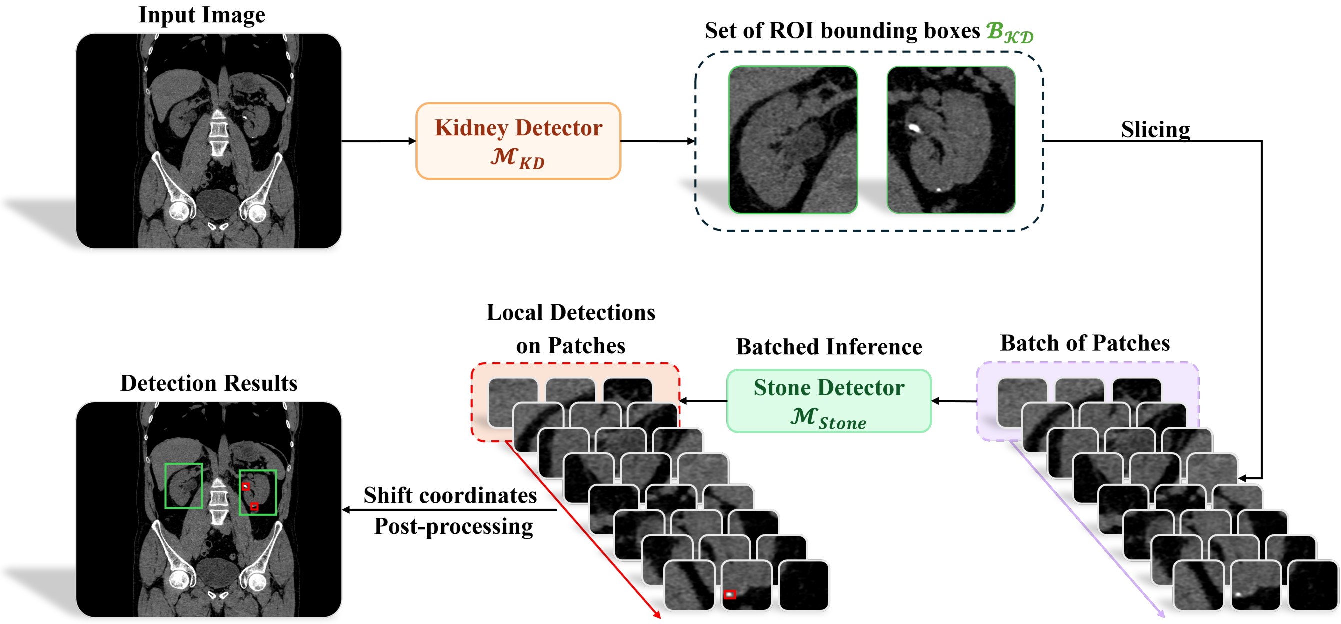

Figure 1: Inference pipeline diagram of the SahiKid framework for small kidney stone detection.

IV. B-SAHI Algorithm

The Batched Slicing Aided Hyper Inference (B-SAHI) algorithm, presented in Algorithm 1, provides a structured and efficient pipeline for detecting objects, particularly small ones, in high-resolution images. It enhances the standard SAHI inference process by incorporating batched processing of image slices, a critical optimization for leveraging the parallel processing capabilities of modern GPUs. The following sections provide a detailed procedural explanation and a computational complexity analysis of the algorithm.

1. Procedural Description

The B-SAHI algorithm is defined by a set of parameters governing slicing, detection, and post-processing.

Inputs and Outputs:: The algorithm requires an input image I, a pre-trained model $\mathcal{M}$, slicing parameters (h_{slice}, w_{slice}) with overlap ratios (r_h, r_w), a batch size S, a post-processing function Ψ_{NMS} (e.g., NMS), and a boolean flag $perform\_FI$. It outputs a final set of object predictions $\mathcal{B}_{final}$, with coordinates mapped to the original image space.

Our algorithm is as follows: (1) *Image Slicing:* The input image I is partitioned into a set of patches $\mathcal{P}$ and their corresponding top-left coordinates C using the getSlicesAndCoords$(\cdot)$ function. The overlap ratios, r_h and r_w, ensure that objects spanning slice boundaries are fully captured in at least one patch. (2) *Batch Formation and Iteration:* The patches are grouped into $N_{batches} = \lceil |\mathcal{P}|/S \rceil$ mini-batches. The algorithm iterates through each mini-batch of patches $\mathbb{P}_k$ and their associated coordinates $\mathbb{C}_k$. (3) *Batched Inference and Coordinate Shifting:* The model $\mathcal{M}$ performs batched inference, $\mathcal{M}_{batch}(\mathbb{P}_k)$, to generate local predictions $\mathcal{B}_{local}$. These predictions are then mapped to the global image coordinates using the shiftCoords$(\cdot)$ function and aggregated into a master list, $\mathcal{B}_{aggr}$; (4) *Optional Full-Image Inference:* If $perform\_FI$ is true, an additional inference pass is performed on the entire image I to detect large objects. The resulting predictions, $\mathcal{B}_{FI}$, are appended to $\mathcal{B}_{aggr}$; (5) *Post-processing and Final Output:* The post-processing function Ψ_{NMS} is applied to the aggregated predictions $\mathcal{B}_{aggr}$ to resolve duplicate detections from overlapping regions and the optional full-image pass. This produces the final, refined set of predictions, $\mathcal{B}_{final}$.

2. Computational Complexity Analysis

The time complexity of B-SAHI depends on the image dimensions $(H \times W)$, slice dimensions $(h \times w)$, model inference cost $C(\mathcal{M})$, and the number of post-processing predictions.

- **Slicing:** The number of slices $|\mathcal{P}|$ is approximately $O(\frac{H \cdot W}{h \cdot w(1-r_h)(1-r_w)})$, simplifying to $O(\frac{H \cdot W}{h \cdot w})$ as overlap ratios are constants. The slicing operation itself has a negligible complexity of $O(|\mathcal{P}|)$.
- **Batched Inference:** This is the primary computational stage. The total complexity is $O(N_{batches} \cdot C(\mathcal{M})) = O(\frac{|\mathcal{P}|}{S} \cdot C(\mathcal{M}))$. The batch size S improves practical

Algorithm 1: B-SAHI (Batched Slicing Aided Hyper Inference)

Input : I: Input Image, $\mathcal{M}$: Pre-trained Detection Model, h_{slice}, w_{slice}: Slice height and width, r_h, r_w: Overlap ratios for height and width, S: Batch size for sliced patches, Ψ_{NMS}: Post-processing function (e.g., NMS, NMM), $perform\_FI$: Boolean flag for Full-Image Inference

Output: $\mathcal{B}_{final}$: Final set of object predictions

begin
- $\mathcal{B}_{aggr} \longleftarrow \emptyset$
- $\mathcal{P}, C \longleftarrow$ getSlicesAndCoords$(I, h_{slice}, w_{slice}, r_h, r_w)$
- $N_{batches} \longleftarrow \lceil |\mathcal{P}|/S \rceil$
- **for** $k \longleftarrow 1$ **to** $N_{batches}$ **do**
  - $\mathbb{P}_k \longleftarrow \mathcal{P}[(k-1)S$ **to** $\min(kS-1, |\mathcal{P}|)]$
  - $\mathbb{C}_k \longleftarrow C[(k-1)S$ **to** $\min(kS-1, |C|)]$
  - $\mathcal{B}_{local} \longleftarrow \mathcal{M}_{batch}(\mathbb{P}_k)$
  - $\mathcal{B}_{shifted} \longleftarrow$ shiftCoords$(\mathcal{B}_{local}, \mathbb{C}_k)$
  - $\mathcal{B}_{aggr} \longleftarrow \mathcal{B}_{aggr} \cup \mathcal{B}_{shifted}$
- **end**
- **if** $perform\_FI$ *is* ***true*** **then**
  - $\mathcal{B}_{FI} \longleftarrow \mathcal{M}(I)$
  - $\mathcal{B}_{aggr} \longleftarrow \mathcal{B}_{aggr} \cup \mathcal{B}_{FI}$
- **end**
- $\mathcal{B}_{final} \longleftarrow \Psi_{NMS}(\mathcal{B}_{aggr})$
- **return** $\mathcal{B}_{final}$

end

throughput by amortizing fixed overheads (e.g., data transfer, kernel launch) across multiple inferences.

- **Full-Image Inference:** The optional full-image pass has a complexity of $O(C(\mathcal{M}))$, assuming the image is resized to a standard model input size.
- **Post-processing:** A standard NMS algorithm exhibits a quadratic time complexity relative to the number of aggregated predictions, i.e., $O(|\mathcal{B}_{aggr}|^2)$.

The overall time complexity is dominated by the batched inference and post-processing stages:

$$O\left(\frac{|\mathcal{P}|}{S} \cdot C(\mathcal{M}) + |\mathcal{B}_{aggr}|^2\right)$$

The algorithm's efficiency stems from balancing the increased number of inference units ($|\mathcal{P}|$) with parallel processing via batching. The NMS step can become a bottleneck if $|\mathcal{B}_{aggr}|$ is very large.

V. Dataset and Experimental Result

This section delineates the comprehensive experimental framework designed to evaluate our proposed two-stage detection system. We first introduce the datasets, followed by a detailed description of the implementation and training protocols. Finally, we present a thorough analysis of the results, culminating in an ablation study that highlights the contribution of each component to the final model's performance, especially in the challenging task of detecting diminutive kidney stones.

1. Datasets and Preprocessing

Our experiments leverage a publicly available dataset, "CT KIDNEY DATASET: Normal-Cyst-Tumor and Stone"[1] which was manually annotated using Roboflow[2]. To support our two-stage pipeline, we curated two distinct datasets for kidney region detection and small kidney stone detection, respectively.

A series of systematic preprocessing steps were applied. All images were first standardized to a resolution of 640×640 pixels to ensure dimensional consistency for the model inputs. This resolution was selected as a balance between preserving image detail and managing computational load. Data augmentation strategies were applied exclusively to the training sets to enhance model generalization, while the validation and test sets remained unaltered to provide an unbiased evaluation.

a) Kidney Region Detection Dataset

The objective of this dataset is to precisely localize the kidney regions, thereby defining a constrained Region of Interest (ROI) for the subsequent detection task. The dataset follows an 80/10/10 split for training, validation, and testing. The augmentation pipeline included:

- **Horizontal Flipping**: Simulates variations in anatomical orientation, ensuring the model is invariant to the left-right positioning of the kidneys.
- **Random Cropping**: By applying random crops with a maximum zoom of 20%, we mimic different fields of view and scales, compelling the model to identify kidneys even when partially occluded or varying in apparent size.

b) Small Kidney Stone Detection Dataset

The primary challenge is detecting stones that occupy a minuscule area of the full-resolution CT image. To rigorously evaluate our approach, the data was prepared in two formats: a standard dataset for baseline comparison and a specialized, patch-based dataset for our proposed method. A distinct set of augmentations was applied to this dataset to address the challenges of detecting minute targets:

- **Horizontal Flipping**: Ensures orientational invariance.

[1]Kidney CT Dataset
[2]Roboflow

TABLE I: Dataset statistics for the kidney region detection model.

Split	Annotations	Original Images	Augmented Images
Train	2066	1,088	3,264
Validation	265	136	136
Test	251	136	136

- **Random Rotation**: Images were randomly rotated within a range of $-13°$ to $+13°$ to simulate slight variations in patient positioning and improve the model's ability to recognize stones from different angles.
- **Noise Injection**: A small amount of noise (up to 0.06% of pixels) was introduced to help the model become more resilient to inherent artifacts in medical imaging.

The **Standard Dataset** (Table II, left) comprises original, full-frame images with annotated kidney stones. It serves to establish a baseline performance, highlighting the limitations of conventional detectors on such small objects.

The **SAHI-oriented Dataset** (Table II, right) was created to facilitate Slicing Aided Fine-Tuning. By partitioning high-resolution images into smaller, overlapping patches, we effectively increase the apparent size of the stones relative to the detector's input dimensions. This slicing process dramatically expands the dataset and provides the model with focused, high-resolution views of the target objects, which is critical for learning their distinguishing features.

2. Implementation and Training Protocol

To ensure reproducibility, all experiments were conducted within a unified environment. We selected YOLOv11n as our base architecture, with training hyper-parameters specified in Table III.

3. Results and Ablation Study

TABLE IV: Performance comparison of YOLO variants on the kidney stone detection task.

Model	mAP_{50}	mAP_{75}	mAP_{50-95}
YOLOv8n	99.5	99.1	85.0
YOLOv9t	99.5	99.2	83.9
YOLOv11n	**99.5**	**99.2**	**85.3**

We evaluate the performance of our proposed kidney stone detection system using the mean Average Precision (mAP) metric at Intersection over Union (IoU) thresholds of 50% (mAP_{50}) and 50-95% (mAP_{50-95}). To determine the optimal base model, we compare three YOLO variants: YOLOv8n, YOLOv9t, and YOLOv11n. As detailed in Table IV, YOLOv11n exhibits a slight performance advantage, achieving an mAP_{50-95} of 85.3% compared to 85.0% for YOLOv8n and 83.9% for YOLOv9t. Based on this, YOLOv11n is selected for further analysis.

The ablation study, central to our evaluation, investigates small kidney stone detection—a task complicated by object size. Table V systematically evaluates the contribution of each pipeline component. It is structured to first demonstrate the limitations of a single-stage approach *without* Kidney Detection and then to validate the effectiveness of our proposed two-stage framework *with* Kidney Detection.

Without kidney detection pre-processing, the baseline configuration (YOLO + FI) performs poorly, with an mAP_{50} of 0.05%, highlighting the challenge of small object detection. The primary challenge addressed in this study is the detection of objects that are disproportionately small relative to the overall image dimensions. In standard object detection models, the hierarchical down-sampling operations within the network's backbone progressively reduce spatial resolution. While effective for learning high-level semantic features of large objects, this process often leads to the loss of fine-grained information essential for identifying small targets like kidney stones. To counteract this, we employed SF and SAHI. By partitioning the high-resolution training images into smaller patches, we effectively increase the apparent size of the stones relative to the model's input field of view. This forces the model to learn discriminative features at a higher effective resolution, conditioning it to become more adept at identifying these minute targets. Incorporating SF+SAHI markedly improves performance to 48.74% mAP_{50}. Adding PO yields a modest gain to 49.76% mAP_{50}, while reintroducing FI slightly reduces it to 47.61% mAP_{50}. The modest performance of these configurations underscores a fundamental limitation: without a focused region of interest, the model must contend with a vast search space containing confounding anatomical structures (e.g., vertebrae, calcified vasculature) that can be easily mistaken for small stones. This often leads to a high rate of false positives and dilutes the model's ability to learn the subtle features of true kidney stones. Therefore, we hypothesized that constraining the search area to only the kidney regions would be a critical step toward improving detection accuracy.

Activating the first stage of our framework—kidney de-

TABLE II: Statistics for the standard and SAHI-oriented kidney stone datasets.

	Standard Dataset			SAHI-Oriented Dataset (Patches)		
Split	**Annotations**	**Original**	**Augmented**	**Total**	**Background**	**With Objects**
Train	1199	734	2,202	5,888	3,984	1,904
Validation	301	160	160	624	416	208
Test	258	160	160	648	420	228

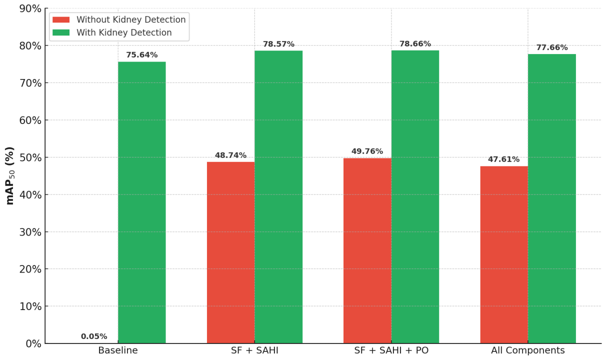

Figure 2: Comparison of mAP50 performance across experimental configurations. The chart shows the significant impact of each component, particularly the kidney localization preprocessing stage (KD).

TABLE III: Training Environment and Key Hyperparameters.

(a) Environment Specification

Component	Specification
Platform	Google Colab
GPU	NVIDIA Tesla T4
CUDA Version	12.4
Framework	Ultralytics YOLO 8.3.99
Backend	PyTorch 2.6.0

(b) Training Parameters

Parameter	Value
Base Model	yolov11n.pt
Epochs	500
Batch Size	32
Image Size	640x640
Patience	20

tection pre-processing (KD)—immediately showcases the benefit of a focused search space. The baseline configuration within this two-stage setup (YOLO + FI + KD) achieves a substantial 75.64% mAP_{50} confirming our hypothesis that reducing the search area is paramount. Adding SF+SAHI increases this to 78.57% mAP_{50}, and incorporating PO results in the highest performance of 78.66% mAP_{50}. However, including FI again slightly lowers it to 77.66% mAP_{50}. These findings underscore the pivotal role of SF+SAHI in small object detection, with KD and PO providing complementary enhancements. The optimal configuration (YOLO + SF + SAHI + PO + KD) achieves the best performance, reflecting the synergy of these components.

We also assess the computational efficiency of inference methods. Table VI compares SAHI and B-SAHI across multiple datasets. B-SAHI consistently outperforms SAHI, reducing the average time per image (AvgTPI) significantly—e.g., from 0.1553 seconds to 0.0329 seconds on the Small Kidney Stone dataset. This efficiency is vital for real-time medical applications.

4. Application and Demonstration

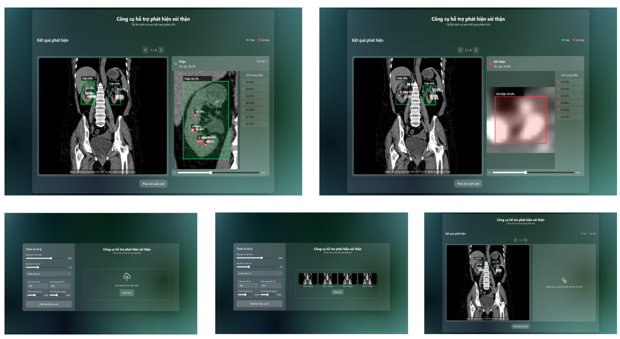

Figure 3: The user interface for our kidney stone detection application, NephroScan, showing the image upload panel, visualization of detection results, and interactive controls.

TABLE V: Ablation study of YOLOv11n on the small kidney stone test set. Performance is measured in mAP (%).

Experimental Configuration	**mAP**$_{50}$	**mAP**$_{75}$	**mAP**$_{50-95}$
Without Kidney Detection Pre-processing			
YOLO + FI (Baseline)	0.05	0.00	0.00
YOLO + SF + SAHI	48.74	12.27	20.02
YOLO + SF + SAHI + PO	49.76	13.27	20.89
YOLO + SF + SAHI + PO + FI	47.61	11.57	19.13
With Kidney Detection Pre-processing			
YOLO + FI + KD	75.64	31.33	37.56
YOLO + SF + SAHI + KD	78.57	31.66	38.89
YOLO + SF + SAHI + PO + KD	**78.66**	**32.42**	**39.42**
YOLO + SF + SAHI + PO + KD + FI	77.66	32.42	38.10

TABLE VI: Performance comparison of SAHI and B-SAHI inference methods across datasets. Total time is in seconds and AvgTPI is in seconds per image.

Dataset	**Total Test Images**	**SAHI**		**B-SAHI**	
		Total Time	**AvgTPI**	**Total Time**	**AvgTPI**
DOTAv2.0 [17]	10542	6157.727	0.5841	1162.843	0.1103
SODA-A [18]	2512	5065.59	2.0166	960.2099	0.3822
VisDrone [19]	8628	2098.96	0.2433	358.2767	0.0415
Small Kidney Stone	100	15.5296	0.1553	3.2963	0.0329

To demonstrate the practical utility of our framework, we developed NephroScan, an interactive web-based application. Designed for clinicians and researchers, the tool allows users to upload CT scan images (e.g., PNG, JPEG) and receive real-time detection results from our two-stage system. The application first localizes the kidney regions and then precisely identifies small stones within them. The user interface, shown in Figure 3, provides an intuitive visualization by overlaying bounding boxes for detected kidneys and stones directly onto the source image. Each detection is accompanied by a confidence score to aid in clinical assessment. Users can interactively analyze the results using integrated tools for zooming and panning.

A video demonstrating the application's workflow—from image upload to result visualization is available on YouTube[3], showcasing features such as parameter adjustment and the display of confidence percentages. For transparency, reproducibility, and community development, the complete source code is publicly accessible on GitHub[4].

VI. Conclusion

This paper introduced a novel two-stage framework that integrates YOLOv11 with the SAHI technique, effectively addressing the critical challenge of small kidney stone detection in CT scans. Our method demonstrates a transformative performance increase, elevating the mean Average Precision from a near-failure baseline of 0.05% with conventional methods to a remarkable 78.66%. This improvement is achieved through a synergistic strategy: an initial kidney localization stage constrains the search space to reduce false positives, while subsequent slicing-aided inference provides the necessary high-resolution views to discern minute stones often missed by standard models.

The implications of our work extend beyond nephrolithiasis, presenting a potent methodology for other challenging small object detection tasks in medical imaging, such as identifying micro-calcifications in mammography or small pulmonary nodules. The proposed B-SAHI algorithm further ensures computational efficiency, enhancing its viability for clinical deployment.

We acknowledge several limitations that suggest avenues for future research. Firstly, the evaluation relies on a single, publicly available dataset. To better assess the model's robustness and generalizability across different scanners and patient populations, multi-institutional validation would be an important next step. Secondly, the current framework performs binary detection (stone or no stone) and does not yet include stone characterization capabilities (e.g., size, density, composition), which are valuable for clinical decision-making.

Furthermore, the benchmark was focused on YOLO variants and internal ablations. Future work could include a

[3]Demonstration Video - https://www.youtube.com/watch?v=scU84FkiTKc
[4]Source Code - https://github.com/p4wndev/nephro-scan

comparative analysis against other medical image segmentation architectures, such as U-Net or nnU-Net, to better position the framework's performance within the broader medical imaging domain. Finally, while the annotation process included internal verification, the absence of direct validation by certified radiologists is a limitation; incorporating expert-in-the-loop validation would further enhance the dataset's quality. Integrating these improvements, along with developing uncertainty quantification mechanisms, would be beneficial for improving the framework's clinical utility and readiness for deployment.

In conclusion, our work establishes a new and effective paradigm for small object detection in medical imaging by strategically combining region localization with slicing-based inference. This framework not only offers substantial improvements in diagnostic accuracy and workflow efficiency but also provides a robust foundation for tackling similar challenges across a diverse range of medical imaging applications, ultimately contributing to better patient outcomes.

REFERENCES

[1] William NP Kwateng. Kidney stones (nephrolithiasis) statistics: Global and us overview, types, trends, and complications.

[2] Naeem Bhojani, Jessica E Paonessa, Marawan M El Tayeb, James C Williams Jr, Tariq A Hameed, and James E Lingeman. Sensitivity of noncontrast computed tomography for small renal calculi with endoscopy as the gold standard. *Urology*, 117:36–40, 2018.

[3] Wayne Brisbane, Michael R Bailey, and Mathew D Sorensen. An overview of kidney stone imaging techniques. *Nature Reviews Urology*, 13(11):654–662, 2016.

[4] Maryellen L Giger and Kenji Suzuki. Computer-aided diagnosis. In *Biomedical information technology*, pages 359–XXII. Elsevier, 2008.

[5] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[6] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pages 21–37. Springer, 2016.

[8] Banoth Ramesh, G Srinivas, P Ram Praneeth Reddy, MD Huraib Rasool, Divya Rawat, and Madhulita Sundaray. Feasible prediction of multiple diseases using machine learning. In *E3S Web of Conferences*, volume 430, page 01051. EDP Sciences, 2023.

[9] Yang Liu, Peng Sun, Nickolas Wergeles, and Yi Shang. A survey and performance evaluation of deep learning methods for small object detection. *Expert Systems with Applications*, 172:114602, 2021.

[10] Syed Muhammad Anwar, Muhammad Majid, Adnan Qayyum, Muhammad Awais, Majdi Alnowami, and Muhammad Khurram Khan. Medical image analysis using convolutional neural networks: a review. *Journal of medical systems*, 42:1–13, 2018.

[11] Areeg Fahad Rasheed and M Zarkoosh. Yolov11 optimization for efficient resource utilization. *arXiv preprint arXiv:2412.14790*, 2024.

[12] Fatih Cagatay Akyon, Sinan Onur Altinuc, and Alptekin Temizel. Slicing aided hyper inference and fine-tuning for small object detection. In *2022 IEEE international conference on image processing (ICIP)*, pages 966–970. IEEE, 2022.

[13] Behzad Mirzaei, Hossein Nezamabadi-Pour, Amir Raoof, and Reza Derakhshani. Small object detection and tracking: a comprehensive review. *Sensors*, 23(15):6887, 2023.

[14] Wei Wei, Yu Cheng, Jiafeng He, and Xiyue Zhu. A review of small object detection based on deep learning. *Neural Computing and Applications*, 36(12):6283–6303, 2024.

[15] Alexander Neubeck and Luc Van Gool. Efficient non-maximum suppression. In *18th international conference on pattern recognition (ICPR'06)*, volume 3, pages 850–855. IEEE, 2006.

[16] Edward L Chaney and Stephen M Pizer. Defining anatomical structures from medical images. In *Seminars in Radiation Oncology*, volume 2, pages 215–225. Elsevier, 1992.

[17] Jian Ding, Nan Xue, Gui-Song Xia, Xiang Bai, Wen Yang, Michael Yang, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Object detection in aerial images: A large-scale benchmark and challenges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

[18] Gong Cheng, Xiang Yuan, Xiwen Yao, Kebing Yan, Qinghua Zeng, Xingxing Xie, and Junwei Han. Towards large-scale small object detection: Survey and benchmarks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(11):13467–13488, 2023.

[19] Pengfei Zhu, Longyin Wen, Dawei Du, Xiao Bian,

Heng Fan, Qinghua Hu, and Haibin Ling. Detection and tracking meet drones challenge. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021.

Truong-Thanh Ma received his a Master's degree in Information Systems from Can Tho University, Vietnam, in 2014. In 2022, he was awarded a Ph.D. in Computer Science from Artois University, France. He is currently a Senior Lecturer at the Faculty of Computer Science, College of Information and Communication Technology, Can Tho University, Vietnam. His research interests span Machine Learning, Deep Learning, Computer Vision, KR&R, Belief Merging, and Ontology. He has actively contributed to several international research projects funded by highly competitive programs, including the Horizon 2020 European Commission and the French National Research Agency (ANR).
Email: mtthanh@ctu.edu.vn

Tien-Dao Luu received his Bachelor's degree in Computer Science from Can Tho University, Vietnam, in 2001, followed by a Master's degree in Computer Science from Université Paul Sabatier, Toulouse, France, in 2006. In 2012, he was awarded a Ph.D. in Bioinformatics from the University of Strasbourg, France. He is currently a Lecturer at the Faculty of Computer Science, College of Information and Communication Technology, Can Tho University, Vietnam. His research interests include Bioinformatics, Deep Learning, Knowledge Discovery, Artificial Intelligence in Healthcare & Medical Field.
Email: ltdao@ctu.edu.vn

Huu-Hoa Nguyen received his Engineering Degree in Computer Science from Can-Tho University, Vietnam. He earned his MSc in Information Systems from HAN University, the Netherlands, and his PhD in Informatics from Lyon University, France. Dr. Nguyen is currently a senior lecturer at the College of Information and Communication Technology, Can Tho University, Vietnam. His research interests span a wide range of computer science topics, including artificial intelligence, machine learning, data mining, knowledge management systems, computer networks, and cybersecurity. He has led multiple international research projects, receiving funding from prestigious sources such as the Horizon-2020 European Commission and the Newton Fund
Email: nhhoa@ctu.edu.vn

Phu-An Thai is a student of the 48th class, majoring in Computer Science at the College of Information and Communication Technology, Can Tho University, Vietnam. His areas of interest include Deep Learning, Natural Language Processing, Computer Vision and Reasoning.
Email: contact.thaiphuan@gmail.com

Phat-Dat Trinh is a student of the 48th class, majoring in Computer Science at the College of Information and Communication Technology, Can Tho University, Vietnam. His areas of interest include Machine Learning, Deep Learning, Computer Vision, Natural Language Processing.
Email: datb2207517@gmail.com

Van-Pha Phan is a student of the 48th class of Information Technology Education, College of Education, Can Tho University. His research interests include Deep Learning, Computer Vision, Educational Technology, Intelligent Tutoring Systems, and the Application of Artificial Intelligence in Education.
Email: phanvankha06122004@gmail.com

Chi-Thien nguyen received a Bachelor of Engineering in Software Engineering from Can Tho University, Vietnam, in 2016, and a Master's degree in Information Systems from the University of Science, Vietnam National University, Ho Chi Minh City, in 2024. He is currently a Project Manager at VNPT Information Technology Company, Vietnam. His research interests include Information Systems for e-Government, Machine Learning, and Computer Vision. He has been involved in the development of digital citizen applications and public service systems in Vietnam.
Email: thiennc.tgg@vnpt.vn