Advanced Detection and Segmentation of Unused Lands in Saudi Arabian Cities for Urban Development

Thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of Science in Data Science

by

Emma Hovhannisyan

Under the supervision of

Gevorg Yeghikyan

AUA
American University *of* Armenia

DEPARTMENT OF DATA SCIENCE

AMERICAN UNIVERSITY OF ARMENIA

SPRING 2024

ABSTRACT

With the rapid urbanization and increasing demands in the real estate sector, the need for precise identification and utilization of urban land resources has become paramount. This paper delves into the precise detection of urban unused lands in Saudi Arabian cities, leveraging satellite imagery. While YOLO (You Only Look Once) models, notably YOLOv8, are instrumental tools for object detection and image segmentation tasks, this study places a significant focus on the crucial aspect of data collection. Through thorough data processing and deep learning techniques, our aim is to delineate these underutilized areas, offering valuable insights for urban planning and land management initiatives. The paper provides detailed insights into the process of data collection, the acquisition of high-resolution satellite imagery and the delineation of polygons representing unused land parcels. Special attention is devoted to the challenges inherent to urban data collection, highlighting the complexities and methodologies specific to this context. The research emphasizes the importance of rigorous data collection methods in accurately identifying urban land features.

Keywords: *YOLO*, *urban*, *segmentation*, *detection*, *evaluation metrics*

Contents

1 Introduction

With the continuous expansion and development of urban areas, the efficient management and utilization of land resources have become paramount. One significant aspect of urban development is the identification and classification of unused lands, which plays a crucial role in urban planning and real estate management. The ability to accurately detect and delineate these underutilized areas can provide valuable insights for policymakers, developers, and urban planners, enabling them to make informed decisions regarding land use and allocation.

The focus of this project is the precise detection of urban unused lands in Saudi Arabian cities, leveraging advanced techniques in satellite imagery analysis. While the utilization of satellite imagery for land classification is not a novel concept, our study places a significant emphasis on the development of robust methodologies for data collection and analysis, particularly focusing on the challenges inherent to urban environments.

The primary objective of the research is to develop an accurate and efficient approach for delineating underutilized lands, with the ultimate goal of supporting urban planning initiatives and addressing the growing demands of the real estate market. By leveraging state-of-the-art deep learning algorithms, such as YOLOv8, in conjunction with high-resolution satellite imagery, the aim is to create a comprehensive framework for identifying and classifying unused lands with a high degree of precision. Through thorough data processing and deep learning techniques, the study seeks to offer valuable insights into the identification and characterization of urban unused lands, providing stakeholders with actionable information for effective land management and development.

Th structure of the paper is the following: Section 2 provides a comprehensive review of existing literature related to urban land detection and classification. Section 3 outlines the methodology employed in our study, including data collection, processing, and the utilization of YOLOv8 for land classification. Finally, Section 4 presents the results of the project and Section 5 conclusions and future work in the field of urban land detection and management.

2 Literature Review

Urban development is a critical aspect of modern city planning, and the efficient utilization of land resources plays a pivotal role in sustainable urban growth. In recent years, there has been a growing interest in urban lands, particularly in regions experiencing rapid urbanization such as Saudi Arabia. This section provides a comprehensive theoretical foundation for the capstone project, serving as a backdrop for the proposed approach and research.

Various approaches have been proposed and applied for satellite imagery analysis, showcasing the effectiveness of remote sensing techniques, as demonstrated by studies such as the work by Babbar and Rathee [1]. Image segmentation, a crucial step in image processing, entails partitioning an image into homogeneous regions to enable subsequent analysis and classification. Techniques like region-based segmentation, which groups adjacent pixels with similar properties into homogeneous segments, and edge detection-based methods, which focus on identifying boundaries between different regions, play pivotal roles in extracting meaningful features from satellite images. Additionally, satellite image classification methods, particularly those based on pixel analysis, are essential for categorizing satellite images according to their spatial resolution, enabling detailed analysis and interpretation.

Zou et al. [2] proposed a satellite image segmentation system using RandLA-Net (ARandLA-Net), which is a widely used, fast, and efficient deep learning network designed for semantic segmentation of large-scale point clouds [3].

Convolutional neural networks are now wildely used in urban development. A convolutional neural network (CNN) is a type of network architecture used in deep learning algorithms, commonly applied in image classification problems such as object detection [4]. YOLO (You Only Look Once) is a popular CNN used for image segmentation and detection. YOLO frames object detection as a regression problem to spatially separated bounding boxes and associated class probabilities One significant advantage of YOLO (You Only Look Once) models is their ability to perform object detection in real-time, making them highly efficient for applications requiring fast processing speeds.[5].

The latest version of YOLO is YOLOv8, and the Figure1 represents the graph of the performance of the YOLOv8 compared with other versions of it [6] .

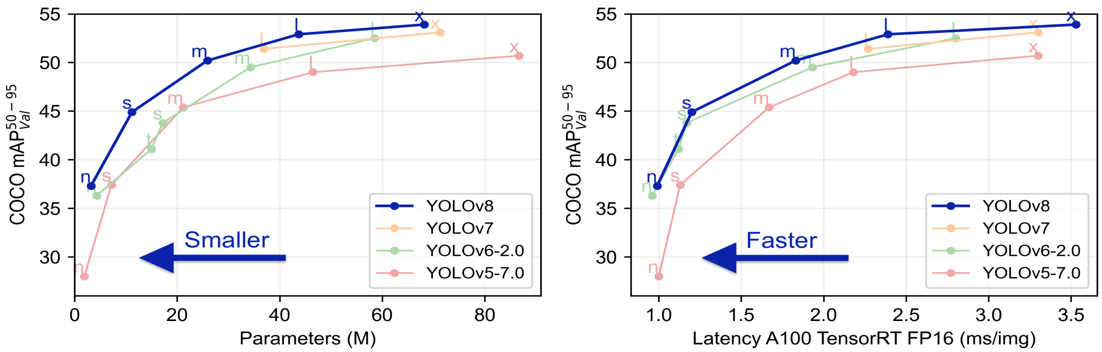

Figure 1: YOLOv8 performance comparison with other YOLO versions.

Grondin [7] proposed a video and image detection system using YOLOv5, another popular version of YOLO. Based on the research, YOLO demonstrates robust performance not only with image data but also when applied to video datasets.

Masayu Norman et al. [8] proposed urban building detection approach using oject-based image analysis (OBIA) [9] and several machine learning algorithms such as SVM which is a supervised machine learning algorithm that classifies data by finding an optimal line or hyperplane that maximizes the distance between each class in an N-dimensional space [10].

3 Methodology

3.1 Data Processing

Data Processing was one of the most important parts of the project as the new dataset was collected and labeled for this specific project. This subsection includes the process of Data Collection and Data Labeling for both detection and segmentation tasks.

3.1.1 Data Collection

The data processing phase of the project started with the software of advanced GIS techniques using QGIS software, an integral tool for spatial analysis and mapping. The objective was to delineate polygons representing areas of unused land within the satellite image of Jeddah, Saudi Arabia. 6000 polygons were drawn on the satellite image of Jeddah, each marking areas of land that seemed unused.

Throughout this process, the primary challenge lay in accurately identifying unused lands and drawing polygons as precisely as possible. Furthermore, all these polygons have completely different sizes and shapes, so they don't need to be limited to just squares or rectangles; they can include any shape of polygon.

After obtaining the polygons in GeoJSON format, the subsequent stage of data collection involved acquiring high-resolution images necessary for detection and segmentation tasks. Given the unavailability of suitable satellite imagery online for Saudi Arabian cities, the Tile+ plugin within QGIS was used to retrieve high-resolution images corresponding to the previously delineated polygons. These images on in Figure 2 were then acquired and divided into smaller segments measuring 1500x1500 pixels each. It was observed that any image size lower than 1500x1500 pixels resulted in significantly reduced resolution and quality, thus necessitating the use of this minimum threshold for optimal data quality. Same polygons and images were used for both doing detection and segmentation.

Figure 2: Example of Images from the Dataset

3.1.2 Data Labeling

Data Labeling is the second part of data processing of the project. Different types of labeling were used for detection and segmentation.

The detection task requires boundary box information for each drawn polygon for each specific image. In the initial phase of data labeling, the objective was to generate bounding box information for each polygon within the 1500x1500 pixel-sized images. This involved extracting the minimum and maximum coordinates (min_y, min_x, max_y, max_x) of each polygon. After overlapping the GeoJSON file with each image and obtaining the general bounding box information for each image, all bounding box information was normalized between 0 and 1 using the formulas below.

$$\text{Normalized}(X_{\min}) = \frac{X_{\min} + \frac{w}{2}}{\text{Image\_Width}},$$

$$\text{Normalized}(Y_{\min}) = \frac{Y_{\min} + \frac{h}{2}}{\text{Image\_Height}},$$

$$\text{Normalized}(w) = \frac{w}{\text{Image\_Width}},$$

$$\text{Normalized}(h) = \frac{h}{\text{Image\_Height}},$$

where ω is the width of the image and h is the height of the image.

Figure 3: Illustration of Normalized bounding boxes

Additionally, as our dataset contains only one class, a class value column containing 0s was added to all bounding box information files, which were saved with the .txt extension. Figure 3 represents the plot of the normalized bounding boxes of unused lands.

For the segmentation part, the labeling process differed from that of detection. While detection required bounding box information, segmentation required binary mask information delineating the exact polygon on each image. To achieve this, binary masks were generated by overlaying images with GeoJSON files corresponding to each image. Once the binary masks were successfully generated as images, they were saved as numerical data with a .txt extension. This resulted in each image having labels in the form of binary masks.

3.2 YOLOv8

YOLO pre-trained model was used to do both segmentation and detection of unused lands. YOLO (You Only Look Once) is very widely used architecture. It is famous of detection characteristics. The key feature of YOLO is that the algorithm predicts bounding boxes and probabilities from the input image with one forward pass which explains the name of the algorithm.It is better than famous architecture like R-CNN, SSD, Faster R-CNN with its accuracy, speed and efficiency. Unlike other methods that focus on specific regions of an image, YOLO divides the image into a grid and predicts bounding boxes for each grid cell [11].

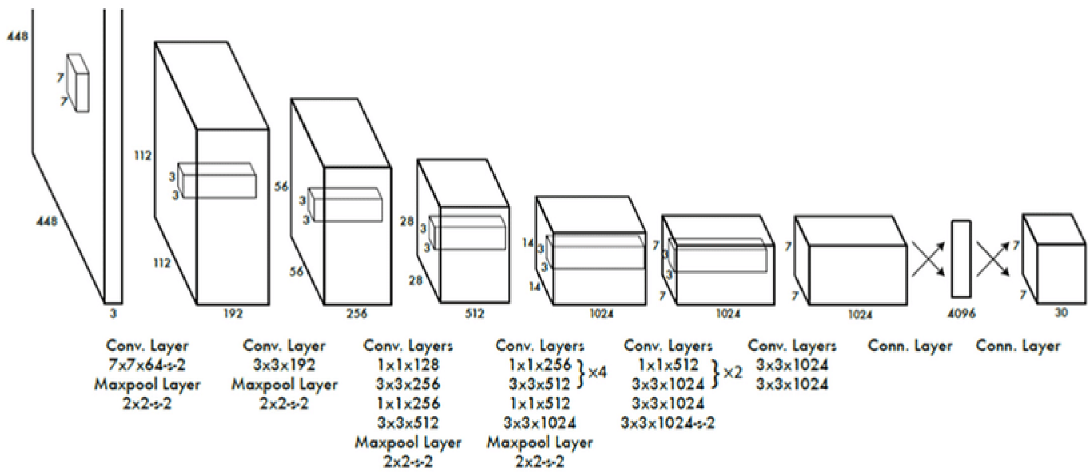

Figure 4: YOLO Architecture

Figure 4 represents [12] the general architecture of YOLO.The Architecture. The detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. The convolutional layers were pretrain on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

There are different versions of YOLO, each with its own improvements and optimizations. The latest version of the YOLO model, YOLOv8, introduced in January 2023 by the Ultralytics team, stands as a cutting-edge solution for classification, detection, and segmentation tasks. YOLOv8 introduces several significant enhancements over its predecessors, including an updated backbone network leveraging EfficientNet to capture high-level features more effectively. Additionally, a novel feature fusion module integrates features from different scales, enhancing object detection capabilities. Furthermore, YOLOv8 incorporates advanced data augmentation techniques, leading to notable improvements in accuracy and speed compared to earlier versions. YOLOv8 offers five different model sizes - nano, small, medium, large, and extra-large - each affecting mean average precision (mAP) and inference time differently. While larger models yield higher mAP but require more time for accurate object detection, smaller models offer faster inference times at the expense of slightly lower mAP. The nano version of YOLOv8 was used for detection and segmentation of unused lands.

3.2.1 YOLOv8 for Detection

Before discussing YOLOv8 detection, let's take a look at the general architecture of a detection model. The structure consists of a backbone, neck, and head components. The backbone, a pre-trained Convolutional Neural Network (CNN), extracts various levels of feature maps—low, medium, and high—from the input image. The neck integrates these feature maps using mechanisms like the Feature Pyramid Network (FPN). Subsequently, it transmits them to the head, which carries out object classification and bounding box prediction tasks. The head may adopt either one-stage or dense prediction models, exemplified by YOLO or Single-shot Detector (SSD), or opt for two-stage or sparse prediction methods like those seen in the R-CNN series.

In YOLO's detection model, the input image is divided into an S × S grid. For each grid cell, the model predicts B bounding boxes and confidence scores for those boxes. The confidence scores reflect the model's confidence that the box contains an object and the accuracy of the prediction. The confidence score is calculated as the product of the probability of an object being present in the cell and the Intersection over Union (IOU) between the predicted box and the ground truth. Each bounding box consists of predictions for the center coordinates (x, y) relative to the grid cell bounds, width (w), height (h) relative to the entire image, and confidence. Additionally, each grid cell predicts C conditional class probabilities, conditioned on the presence of an object. At test time, the conditional class probabilities and individual box confidence predictions are multiplied to obtain class-specific confidence scores for each box, encoding both the probability of the class appearing in the box and the fit of the predicted box to the object. At test time the conditional class probabilities and the individual box confidence predictions are multiplied together,

$$Pr(Class_i|Object) \times Pr(Object) \times IOU_{truth} = Pr(Class_i) \times IOU_{truth}$$

which gives us class-specific confidence scores for each box. These scores encode both the probability of that class appearing in the box and how well the predicted box fits the object. The Figure 5 represents the exact detection step of YOLO [11].

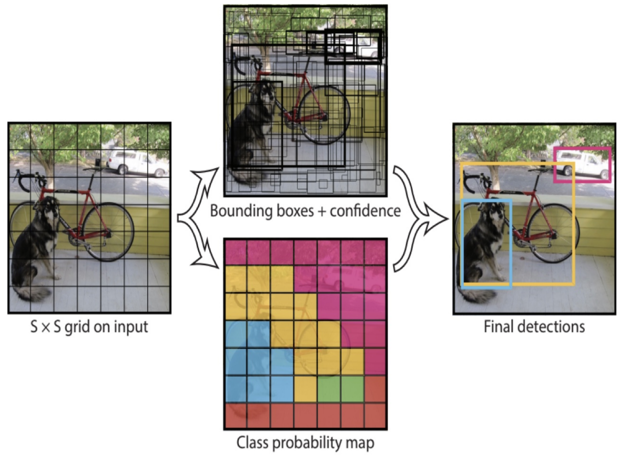

Figure 5: YOLO Object Detection Steps.

3.2.2 YOLOv8 for Segmentation

Image Segmentation is a computer vision challenge that entails recognizing and isolating individual objects within an image. This process involves detecting the boundaries of each object and assigning a distinct label to each one. The goal of image segmentation is to produce a pixel-wise segmentation map of the image, where each pixel is assigned to a specific object instance. While detection focuses on identifying the presence of objects within an image, segmentation is about precisely delineating the objects' boundaries.

In YOLOv8, the default task is object detection, which involves detecting and localizing objects in an image or video. Semantic segmentation, on the other hand, aims to assign a class label to every pixel in an image. While YOLOv8 can be employed for object detection, it is not specifically designed for semantic segmentation.

To train YOLOv8 for semantic segmentation, pixel-level annotations are typically required instead of object bounding box annotations. These pixel-level annotations provide the pixel-wise labels necessary for semantic segmentation. If only object bounding box annotations and corresponding class labels are available, it may not be suitable for directly training YOLOv8 for semantic segmentation.

YOLOv8-Seg shares the fundamental architecture of YOLOv8-Detect, with an additional output module in the head responsible for producing mask coefficients.

Additionally, an FCN layer called the Proto module is added, which outputs masks aiding in segmentation.

Within the Ultralytics package, two types of instance segmentation tracking are available [13]:

1. Instance Segmentation with Class Objects: Each class object is assigned a unique color, enabling clear visual separation.

2. Instance Segmentation with Object Tracks: Each track is represented by a distinct color, facilitating easy identification and tracking.

As the aim of the project is to identify only unused lands of the cities, there is only one class called unused. That is why Instance Segmentation with Class Objects was chosen.

3.2.3 Loss function and evaluation metrics

In YOLOv8, the loss function is typically composed of several components, including:

$$L_{loc} = \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 + (\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2]$$

$$L_{conf} = \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{obj} (C_i - \hat{C}_i)^2 + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{noobj} (C_i - \hat{C}_i)^2$$

$$L_{cls} = \lambda_{cls} \sum_{i=0}^{S^2} \mathbb{1}_{i}^{obj} \sum_{c \in classes} (p_i(c) - \hat{p}_i(c))^2$$

where: - L_{loc} is the localization loss, - L_{conf} is the confidence loss, and - L_{cls} is the classification loss.

Additionally, YOLOv8's performance is often evaluated using various metrics, including:

- **Mean Average Precision (mAP):** mAP measures the average precision across different object classes and IoU thresholds. It provides a comprehensive assessment of the model's detection performance.

$$mAP = \frac{1}{N_{class}} \sum_{c=1}^{N_{class}} AP_c$$

- **Precision (P) and Recall (R):**

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

  where TP denotes true positives, FP denotes false positives, and FN denotes false negatives.

- **Intersection over Union (IoU):** IoU measures the overlap between the predicted bounding boxes and the ground truth boxes:

$$IoU = \frac{Area_{overlap}}{Area_{union}}$$

- **F1 Score:** The F1 score is the harmonic mean of precision and recall:

$$F1 = 2 \times \frac{P \times R}{P + R}$$

These metrics collectively provide insights into the YOLOv8 model's ability to accurately detect and localize objects in images or videos.

4 Results

The final dataset used for this project consists of 1019 images with dimensions of 1500x1500 pixels each, saved in .tif format. Each image is labeled for both segmentation and detection tasks. The dataset was split into three parts: training, validation, and testing, with proportions of 80%, 10%, and 10% respectively. This section represents detailed description of the results both for Detection and Segmentation. The following metrics are part of the evaluation:

Precision (B) and Recall (B): These metrics evaluate correct detections (precision) and capturing all true positives (recall), offering a balanced view of accuracy and completeness.

mAP50 (B) and mAP50-95 (B): Mean Average Precision (mAP) summarizes detection accuracy across different thresholds, providing insights at varying confidence levels.

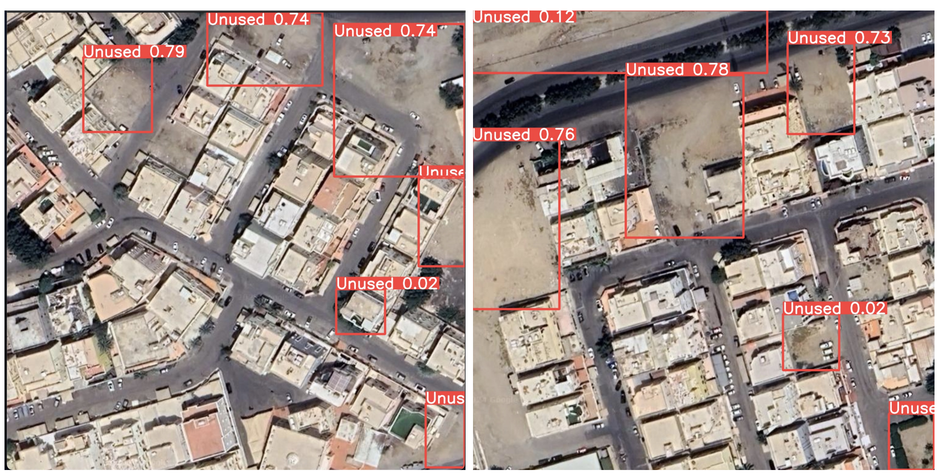

Figure 6: Detection of Unused Lands on Test Data

YOLOv8 pre-trained model was fine tuned for the detection. The number of classes was one and it was unused. The model was suppose to learn the unused lands of training dataset given the bounding boxes information and after that predict unused lands on unseen data. As you can see in Figure 6, the detection on unseen data is pretty precise and accurate which shows that the model was well learned very well on train data and cross validation was also good.

Below is Train and Validation detection loss graphs. The trend of loss graphs is mostly decreasing, however the model needs improvements.

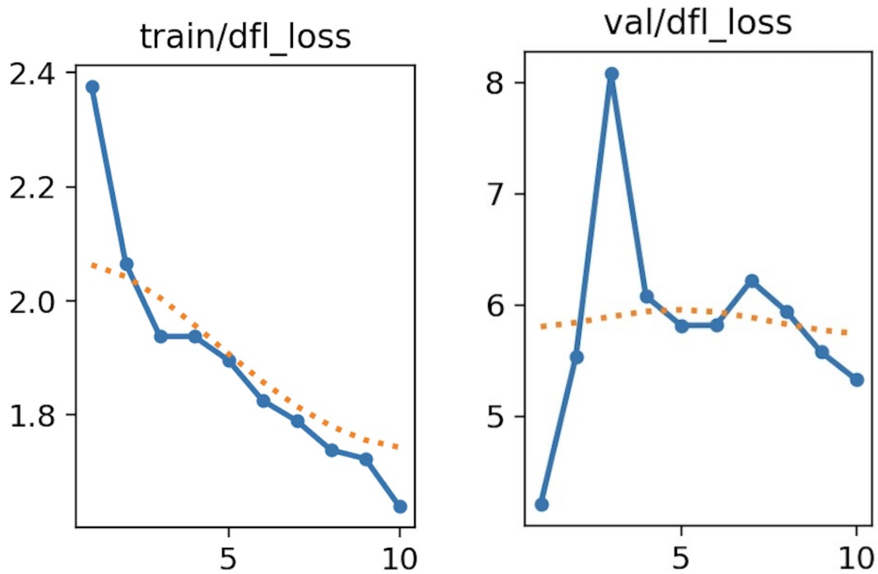

Figure 7: Losses of Detection for Train and Validation

YOLOv8 pretrained model was fine tuned for the segmentation. The number of classes was one and it was unused. The model was suppose to learn the unused lands of training dataset given the labels information of binary masks and after that segment unused lands on unseen data.

As it is shown in Figure 8, the segmentation on unseen data is pretty precise and accurate which shows that the model was well learned on train data.

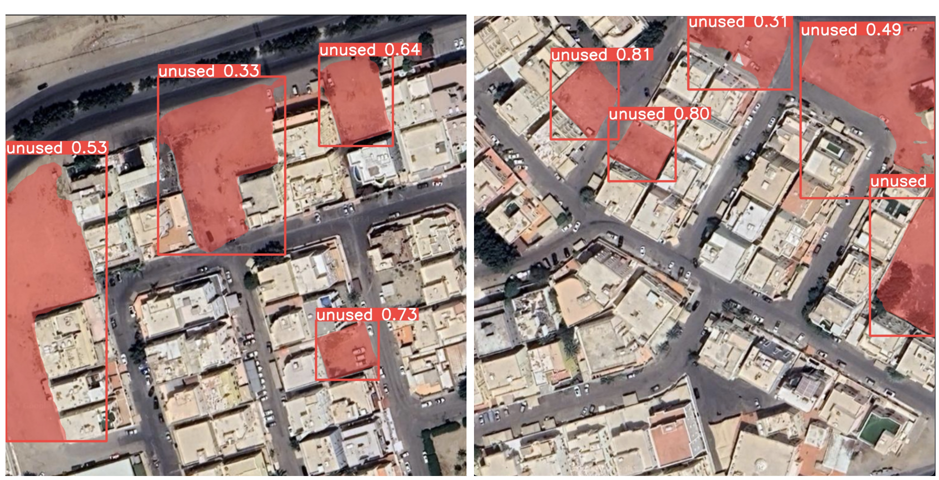

Figure 8: Segmentation of Unused Lands on Test Data

Concurrently, the evaluation metrics underscore the necessity of achieving ro-

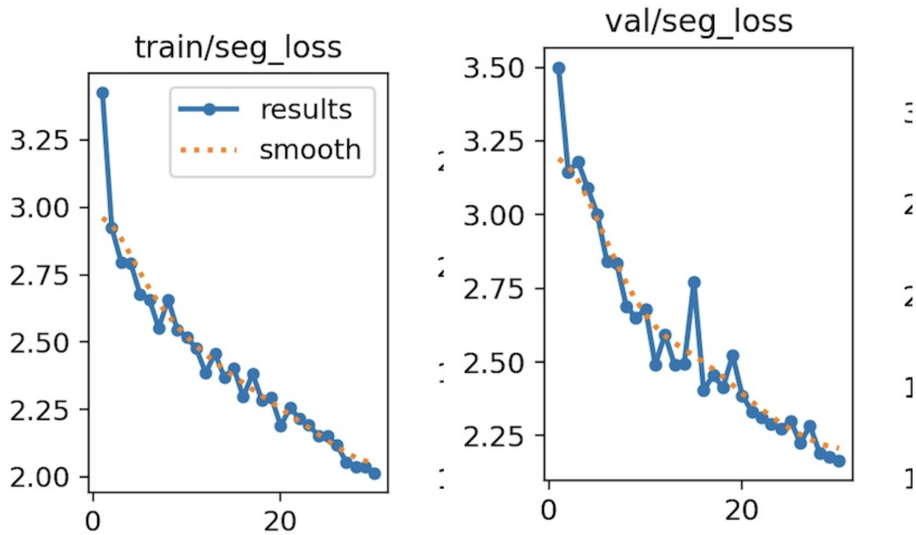

Figure 9: Losses of Segmentation for Train and Validation

bust model performance. Figure 9 shows the train and validation segmentation loss graphs. The trend both for train and Validation segmentation losses is decreasing. Over some epochs there are some increasing trends too, but overall the trend is decreasing. Although that is not a guarantee for anything, but it is important to have this trend to be sure that there is nothing wrong with our data in general.

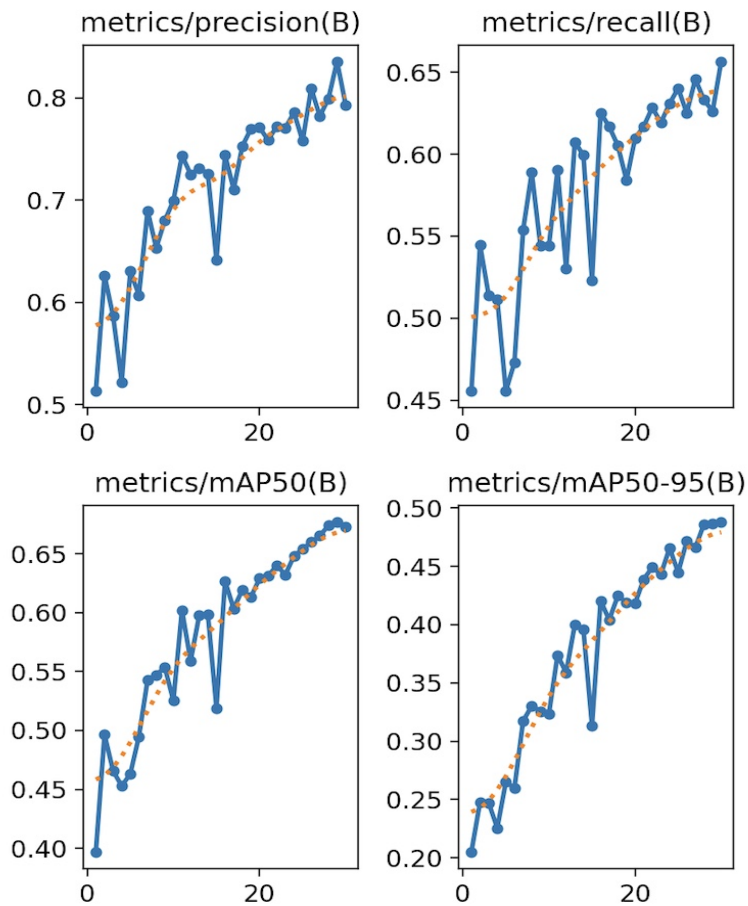

Figure 10: Evaluation Metrics

Figure 10 demonstrates the results of four key metrics mentioned in the beginning of this section. The evaluation results demonstrate consistently high or increasing performance across all four key metrics, including precision (B), recall (B), mAP50 (B), mAP50-95 (B), underscoring the robustness and effectiveness of the YOLOv8 model in object detection tasks.

5 Conclusion and Future Work

In conclusion, this project has presented a comprehensive exploration of object detection and segmentation using the YOLOv8 model, with a specific focus on its performance in detecting and segmenting unused lands within satellite imagery. YOLO, known for its prowess in object detection, proves to be a versatile tool not only for detection but also for segmentation tasks. Leveraging YOLO's capabilities, we have demonstrated the feasibility of utilizing it for precise segmentation of unused lands within satellite images.

Segmentation, offers a more precise and practical approach to identifying unused lands within images. By delineating the boundaries of different areas directly, segmentation provides a clearer understanding of the location and extent of unused land. In contrast, traditional detection methods may lack the granularity required for such tasks.

By applying advanced methodologies like segmentation within the YOLO framework, the study offers a precise means of identifying unused lands within urban areas. This capability not only informs strategic urban planning initiatives but also guides real estate decision-making by providing insights into land utilization patterns. With the ability to pinpoint unused lands accurately, stakeholders in urban development and real estate sectors can optimize land use, identify redevelopment opportunities, and foster sustainable growth in urban environments. Overall, this research serves as a practical tool for enhancing efficiency and sustainability in urban development and real estate practices.

For future work, enhancing the methodology's precision and practical applications is key. Utilizing more powerful GPUs can enable training on larger datasets, leading to a more accurate model. Additionally, integrating area computation and approximation functionalities can facilitate land valuation and property assessment in the real estate sector. Exploring data fusion techniques, such as incorporating aerial imagery, can further improve the model's performance and applicability. These advancements promise to empower stakeholders with valuable tools for sustainable land management and urban planning.

References

[1] Jatin Babbar and Neeru Rathee. Satellite image analysis: A review. In *2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, pages 1–6, 2019.

[2] Y. Zou, H. Weinacker, and B. Koch. Towards urban scene semantic segmentation with deep learning from lidar point clouds: A case study in baden-württemberg, germany. *Remote Sensing*, 13(16):3220, 2021.

[3] MATLAB documentation: randlanet function. https://www.mathworks.com/help/lidar/ref/randlanet.html#, Accessed 2024.

[4] R. Awati. Convolutional neural network (cnn). 2023. Enterprise AI.

[5] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *University of Washington, Allen Institute for AI, Facebook AI Research*, 2016.

[6] Chinmay Bhalerao. YOLO v8! The real state-of-the-art? My experience & experiment related to YOLO v8, January 2023. Medium.

[7] Jean-Sébastien Grondin. Detecting objects in urban scenes using YOLOv5. *Towards Data Science*, 2021.

[8] Masayu Norman et al. Urban building detection using object-based image analysis (obia) and machine learning (ml) algorithms. *IOP Conf. Ser.: Earth Environ. Sci.*, 620:012010, 2021.

[9] GISGeography. Obia – object-based image analysis (geobia), 2024.

[10] IBM. What are svms?, 2023.

[11] Alžběta Turečková, Tomas Holik, and Zuzana Oplatkova. Dog face detection using yolo network. *MENDEL*, 26:17–22, 12 2020.

[12] Pratheesh Shivaprasad. A comprehensive guide to object detection using yolo framework — part i. *Towards Data Science*, January 2019.

[13] Ultralytics. Instance segmentation and tracking using ultralytics yolov8. https://docs.ultralytics.com/guides/instance-segmentation-and-tracking/, Month YYYY.