AI-Driven Window Detection and Semantic Segmentation from Street View Imagery Using Grounding DINO and DeepLabV3 for Digital Twin Modeling

Sumeer Koirala[1], Prof. Dr. Xiaoxiang Zhu[2], Dr.-Ing. Yao Sun[3], Mr. Alejandro Rueda Segura[2],
mailtosumeer@gmail.com, xiaoxiang.zhu@tum.de, yao.sun@dlr.de, alejandro.rueda@tum.de
[1]Survey department, [2]Technical University of Munich, [3]German Aerospace Center,

KEYWORDS

Deep learning, Semantic segmentation, Grounding DINO, DeepLabV3, Digital twins

ABSTRACT

AI-driven automated generation of façade information using street view images can be a vital step toward large-scale urban digital twin generation. Traditional approaches rely on rule-based methods and manual annotation, which poses a significant time lag and is difficult on a large scale. This study focused on a state-of-the-art AI-based pipeline for window detection from street view images and semantic segmentation for windows parameter generation. The proposed workflow consists of image rectification(correcting perspective distortion in street view images). Secondly, window regions are detected using a zero-shot object detection model(GroundingDINO) followed by semantic segmentation using a fine-tuned DeepLabV3 model trained on the WinSyn dataset. Through systematic experimentation with different parameters and hyperparameters, the optimization of label classes from 11 to 3 classes significantly improved segmentation performance. The refined model achieved a mean Intersection over Union (mIoU) of 80.74%, representing an improvement of 44.31% compared to the baseline performance of 36.43% obtained using four classes. This class optimization reduced ambiguity among window components and improved segmentation consistency. Segmentation outputs are further refined using morphological operations to improve frame continuity and remove noise in window panes. Geometric parameters such as pane arrangement, frame thickness, and window layout are extracted from the refined masks and structured into a parametric representation. The proposed pipeline demonstrates the potential of combining zero-shot detection and semantic segmentation for automated façade analysis from street-view imagery. The extracted window information can support applications in urban digital twin generation, building energy modeling, and large-scale architectural analysis.

1. INTRODUCTION

Window detection and reconstruction from street-view images is very challenging. There is a lot of complexity in architecture, design, and structure, and real-world styles. Traditional window reconstruction techniques like Laser scanning, surveying, and photogrammetry are quite time-consuming and costly. On

the other hand, traditional object detection and segmentation techniques require a large volume of data and annotated labels for desired performance, and it could be time-consuming and costly for training data creation or training data collection. CityGML 3.0 is an essential standard for the generation of high-quality and realistic digital twins of urban environments. Window reconstruction is one of the critical components for CityGML 3.0 data. On the other hand, a city like Munich is still utilizing CityGML 2.0 and looking forward to an upgrade. Although the upgrade is critical, manual ways of window extraction is labor intensive process when dealing with large city areas. Furthermore, existing methods are constrained by the need for a large amount of training labels and limited ability to handle the complex and diverse nature of real-world data/ images.

1.1 Research gap

Despite having AI-driven techniques for façade detection and segmentation, several limitations persist:

Dependence on Synthetic Data: Several methods heavily rely on large volumes of high-quality synthetic or procedural datasets, limiting scalability and reliability in complex, real-world environments (Kelly, et al., 2024).

Lack of Domain-Specific Frameworks: General object detection models exist, but frameworks tailored for window detection and reconstruction are scarce, especially for unlabeled or complex real-world data (Pang & Biljecki, 2022).

Simplified 2D-to-3D Window Models: Existing 2D-to-3D approaches often oversimplify window structures, failing to capture detailed geometries for accurate digital twins (Hu et al., 2022).

Limited Use of Street-View Imagery: Low-resolution street-view images are underutilized for 3D reconstruction, leaving opportunities to develop image-to-mesh pipelines for real-world window geometries.

2. STATE OF THE ART

Object detection and semantic segmentation are two strongly correlated tasks for visual recognition. These tasks are strongly correlated, but they are resolved as separate tasks using entirely different techniques (Dong, et al. 2014). Object detection is the task of formulating a bounding box to enclose an object of interest (Felzenszwalb, et al. 2009), whereas semantic segmentation assigns a label to each pixel from predefined classes or predefined labels (Carreira, et al. 2012). In this research, I am interested in applying object detection methods for detecting window frames in street view images utilizing the detected windows, labeling each pixel with Windows classes, and generating 3d Windows. Semantic segmentation and object detection are highly correlated tasks and can be used for mutually beneficial tasks like windows detection from street view images, semantically semantic images, and reconstruction of geometry.

On one hand object detection provides prior knowledge to object (e.g., Windows), which can be later used for refining semantic segmentation. On the other hand, semantic segmentation is capable of providing both local and global semantic information from detected

object (e.g., windows detected from street view image) (Peng, Nan et al. 2020).

2.1. Zero-shot object detection

Object detection can be termed as computer vision technique, which identifies objects, labels items within images, videos, or even live footage. The main task of object detection is to identify and locate single or multiple objects with reference to predefined categories from images. (Tang, Feng et al. 2017).

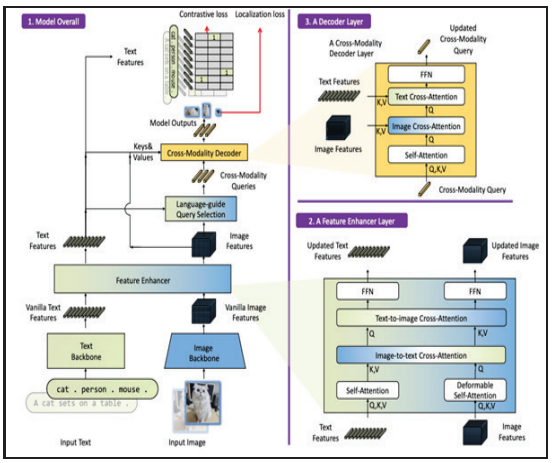

Figure 1: GroundingDINO architecture.

Zero shot learning (ZSL) is technique of identifying objects, even though training sets are not available. ZSL significantly tackles problem of data scarcity (Wang, Zhang et al. 2020). This approach mostly focuses on feature identification using concept of classification problem. Some of the significant challenges of ZSL are a) most of the ZSL benchmarks datasets mostly focuses on single dominant object in single image. Secondly, most of the ZSL are based on zero-shot classification method, which is not convenient to apply directly in the entire scene. Finally, ZSL fails to consider occlusion and clutters in real world scenarios (Li, Yao et al. 2019) Whereas Zero shot object detection (ZSoD) is techniques of simultaneously detection and localization of novel categories of object classes, even without the visual presence of those classes during the training phase. ZSoD can be utilized in variety of application ranging from ovel object localization, tracking and retrieval of object to determination of relationship of object with its environment only using either object name or natural language description (Rahman, et al. 2020).

2.2. You Only Look Once (YOLO) model

YOLO is an objection detection paradigm that works on the modality of utilizing object detection as regression problem, using spatially separated bounding boxes and associated class probabilities to detect objects. Whereas prior object detection model utilized repurpose classifiers to perform detection. YOLO model utilizes single neural network to predict bounding box around object of interest and predicts class probabilities directly using full image in single evaluation. Having the full detection pipelines using single network, hence it can be optimized directly on end-to end detection performance (Redmon 2016)

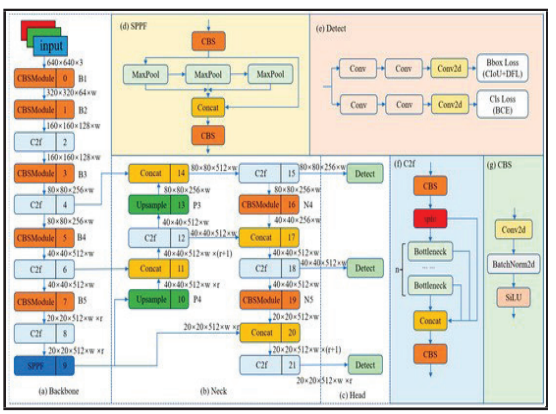

Figure 2: YOLOV8 model architecture. Source:(Wang, Chen et al. 2023)

2.3. Semantic segmentation

Semantic Segmentation (SS) can be defined as process of scene labelling, which is intended to assign a semantic label to each image pixel. One of the major challenges of SS is presence of multiple classes with higher degree of similarity between them (Yu, et al. 2018). SS provides category labels to each pixel, it is very beneficial for variety of tasks including self-driving, pedestrian detection, defect detection also geometry generation.(Hao, et al. 2020). SS was first termed in 1970 by (Ohta, et al. 1978) emphasizing on generation of semantically meaningful semantic regions. Lately, with advancement of deep learning techniques, research on SS has significantly advanced. Accuracy of segmentation has significantly increased with introduction of pioneering fully convolution network (FCN) by (Long, et al. 2015)

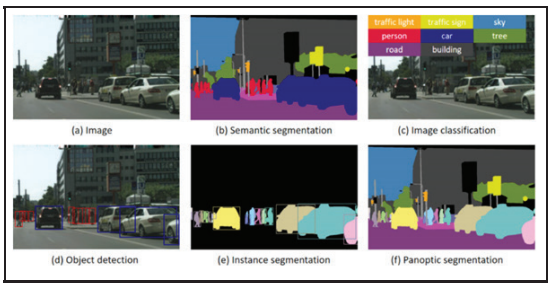

Figure 3: Different computer vision tanks. source : (Hao, et al. 2020)

2.4. Deeplabv3plus

Deeplabv3plus model is based on encoder-decoder method. This method uses different network forms and pyramid pooling modules to improve the accuracy of the network. Furthermore, representative algorithm is composed of Deeplab series(Liu, et al. 2024) and pyramid scene parsing network (PSPNet) (Sun and Wang 2018). Deeplab module perform spatial pyramid pooling at several grid scales or apply numerous atrous convolutions with varying rates. First component of deeplabv3plus network(Yang, et al. 2020) is backbone network. Backbone network extracts feature semantic information.(Wang, Li et al. 2019)

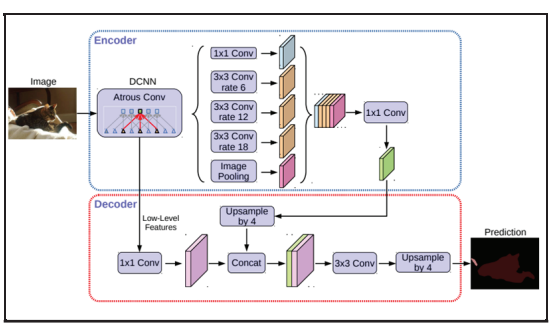

Figure 4: Deeplabv3plus model. Source:(Chen, Zhu et al. 2018)

3. METHODOLOGY

The proposed pipeline combines zero-shot object detection and semantic segmentation to automatically extract window information from street-view imagery. The workflow consists of image rectification, window detection, semantic segmentation, mask refinement, pane arrangement, and geometric parameter extraction. The resulting data can support urban digital twin generation, building energy modeling, and architectural analysis.

3.1. Window segmentation

3.1.1. Semantic segmentation

Semantic segmentation had been implemented using the DeepLabV3 model. This section provides in detail the process of training the DeepLabV3 model on the WinSyn dataset.

The window segmentation process consists of key parameters such as model architecture, class weights, learning rate, and the specific classes involved in the segmentation task. The

approach ensures that the model effectively learned the necessary features for detecting

and segmenting windows in different settings, and the choice of the most appropriate settings is intended at the end of this process. SS was carried out in the WinSyn datasets (Kelly, et al. 2024).

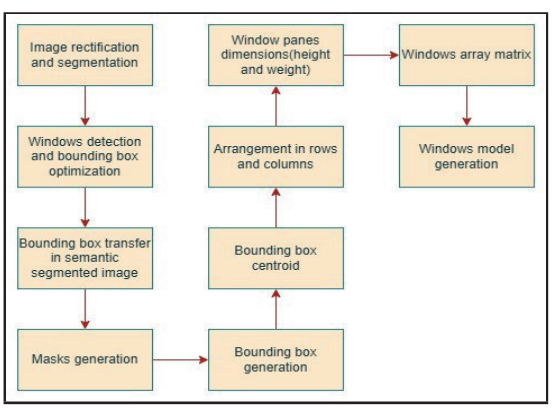

Figure 5: Overall methodology

Semantic segmentation was carried out using following subtasks:

3.1.2. Image Rectification

Street-view images from the WinSyn dataset were first rectified using a homographic transformation (Sasank, 2020) to correct perspective distortion and align window structures. Homography matrices were computed and applied to both segmented

masks and ground truth images to ensure consistency in subsequent analysis.

Bounding Box Generation and Transfer: Bounding boxes were generated around windows in rectified images to correct segmentation errors, with multiple boxes distinguished using color coding. These coordinates were then transferred to the segmented images, and pixels outside the boxes were reclassified as walls to improve segmentation accuracy.

3.1.3. Mask Generation and Refinement

HSV-based red and green masks were created to differentiate window panes and frames. Morphological operations, including opening with a 5×5 kernel, were applied to remove noise and connect fragmented components. Contours corresponding to individual panes were filtered by area, eliminating small or spurious detections.

Pane Arrangement: Bounding box centroids were calculated to arrange panes into rows and columns. Grouping thresholds of 15 pixels in both x and y directions were used, and dimensions within a 5-pixel tolerance were normalized to ensure consistent layout.

Window Matrix Construction and Frame Thickness: Panes were encoded into a 2D matrix representing spatial arrangement and inter-pane relationships. Frame thickness was calculated as the average Euclidean distance from five key points on the frame to the nearest pane edges. All extracted parameters—including row and column counts, pane dimensions, frame thickness, and spatial topology—were saved in JSON format for visualization, parametric analysis, and digital twin integration.

3.2. Windows detection

The window detection process utilized the GroundingDINO model. GroundingDINO model combines images and natural language processing to detect objects in an image and generate a bounding box. GroundingDINO model was utilized with the framework mentioned in figure1.

In the first step of window detection, street view images were fed into the GroundingDINO model along with the text prompt "Window" to specifically focus the model on detecting windows in the street view images. Secondly, GroundingDINO, a transformer-based object detection model, combines the image and text prompts together. It focuses on different segments of the image and correlates these segments with the text prompt "window". The model checks the shape, structure, and appearance of each region in the street view image, distinguishing windows from other structures like façades, doors, and rooftops. It then determines a confidence score for each detected segment. Thirdly, each detected window segment is encapsulated by a bounding box. This bounding box is generated using four coordinates: x-min, x-max, y-min, and y-max, which represent the top, bottom, left, and right edges of the detected image. Fourthly, visualizations of the generated images show that, in some cases, the bounding boxes narrowly capture the windows' edges in the street view images. To address this issue, a 10-pixel margin was added to each side of the detected image bounding box:

$xmin = xmin - 10$, $ymin = ymin - 10$, $xmax = xmax + 10$, $ymax = ymax + 10$.

Finally, after expanding the bounding boxes and creating a buffer around the detected windows, the windows are clipped.

Post-Training Evaluation:

Post training evaluation of the model was carried using standard segmentation metrics Intersection over Union (IoU) and pixel accuracy. These metrics depicted how well the model segmented the windows and its constituent structures. Once the model reached

satisfactory performance, inference was done on to real-world street view images, providing pixel-level classification of windows and their components, which was used for 3D

window reconstruction.

Accuracy Assessment of Segmentation:

The accuracy assessment of the semantic segmentation task was performed by comparing the segmentation results with the ground truth. The ground truth data used for this assessment came from the Windows labels in the Winsyn dataset, which contained a total of 452 images. These labels were divided into three classes: window panes, window frames, and walls.

Optimized Parameters for Semantic segmentation:

Semantic segmentation was performed using a fine-tuned DeepLabV3 model on the WinSyn dataset. Key optimization parameters included:

- **Image and Crop Size:** Input images were resized and cropped to ensure uniform processing and preserve window details.
- **Model Training:** A pre-trained backbone was fine-tuned with selective freezing of layers to leverage learned features while maintaining stability.
- **Iterations and Learning Rate:** Training iterations and learning rate were tuned for optimal convergence. Class weights were applied to mitigate class imbalance.
- **Class Selection:** The original WinSyn dataset contained 11 label classes, but using all classes resulted in suboptimal performance. Segmentation was optimized to focus on four key categories critical for window reconstruction:
    1. **Window Pane** – glass portions of the window
    2. **Window Frame** – surrounding structural elements
    3. **Wall** – areas surrounding the window
    4. **Background** – all remaining regions

This optimization from 11 to 4 classes reduced ambiguity, improved segmentation consistency, and ensured accurate extraction of window geometry for downstream analyses.

4. RESULTS AND DISCUSSION

4.1. Semantic segmentation

The impact of class selection on semantic segmentation performance is illustrated in Figure 6. Initial experiments using the full 11-class WinSyn dataset revealed significant challenges. Classes such as blinds, shutters, bars, and miscellaneous objects exhibited high visual similarity to panes and frames, leading to frequent misclassification and a maximum mIoU of only 36.43%. Segmentation masks were often noisy, with window components partially merged or misidentified, demonstrating that the high number of classes introduced unnecessary complexity for the model.

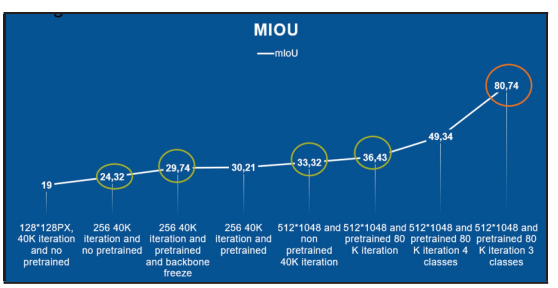

Figure 6: Segmentation optimization

To address this, the number of classes was reduced to four key categories: window pane, window frame, wall, and background. This simplification improved class separability and reduced ambiguity, raising the mIoU to 49.34%. While performance improved considerably, misclassification between blinds and panes persisted due to similar color and texture characteristics. These results indicate that even moderate reductions in class number can significantly improve segmentation accuracy but may still be insufficient when visually overlapping features exist.

The final optimization combined similar

classes, resulting in a three-class setup: window pane, window frame, and wall. This configuration prioritized the most critical components for window reconstruction and minimized distractions from redundant classes. With the same pretrained backbone and 80K training iterations, the model achieved a maximum mIoU of 80.74%. Segmentation masks under this configuration were markedly cleaner, with clear separation between panes, frames, and surrounding walls. Fine details such as narrow frame boundaries and individual panes were accurately captured, demonstrating the effectiveness of the combined strategy of class simplification, fine-tuning, and sufficient iterations.

Table 1: Segmentation accuracy.

Metric	**Value (%)**
Overall Accuracy	93.29 %
Precision (Wall)	96.89 %
Precision (Frame)	77.61 %
Precision (Panes)	90.29 %
Recall (Wall)	94.22 %
Recall (Frame)	81.33 %
Recall (Panes)	95.86 %
F1-Score (Wall)	95.18 %
F1-Score (Frame)	77.82 %
F1-Score (Panes)	92.14 %
Mean Intersection over Union (mIoU)	81.56 %
Metric	**Value (%)**

Table 1 complements these visual findings by providing quantitative performance metrics. The model achieved high overall accuracy (93.29%), with strong precision for walls (96.89%) and panes (90.29%) and slightly lower precision for frames (77.61%), reflecting the challenge of consistently detecting narrow frame structures. Recall values were similarly high for walls (94.22%) and panes (95.86%), while frames achieved 81.33%, indicating that most window components were correctly identified. The F1-scores further confirm consistent performance, with walls (95.18%) and panes (92.14%) outperforming frames (77.82%). The overall mIoU of 81.56% reflects robust segmentation quality across all classes and supports the pipeline's suitability for accurate window parameter extraction.

Overall, these results demonstrate that semantic segmentation performance is highly sensitive to both the number of classes and hyperparameter selection. Class simplification, combined with fine-tuning pretrained models and optimized training iterations, provides a reliable approach to segmenting complex window structures from street-view imagery. These optimized segmentation masks form the foundation for downstream processing, including bounding box extraction, pane arrangement, frame thickness calculation, and spatial encoding, enabling precise reconstruction of window geometry for digital twin modeling and large-scale architectural analysis. Figure 7 shows the semantic segmentation of images in their original form, which contains the geometric distortion. Homographic transformation was used to rectify the image depicted in Figure 8.

Figure 7: Semantic segmentation on original images

4.2. Windows detection

Window detection from street-view imagery was performed using two state-of-the-art object detection models: YOLOv8 and Grounding DINO. The goal was to identify the most suitable model for complex urban environments with diverse window architectures, varying lighting conditions, occlusions, and camera angles. Both models were evaluated based on precision, recall, mean average precision (mAP), bounding box alignment, and detection speed.

Figure 8: Semantic segmentation results(rectified)

4.2.1. YOLOv8 window detection

The YOLOv8 model, known for its speed and efficiency in single-stage object detection, was fine-tuned on annotated street-view images containing window locations. Under a 50% Intersection over Union (IoU) threshold, YOLOv8 achieved a precision of 91.4% and recall of 91.0%, indicating strong detection performance. Most missed detections were caused by occlusions, complex camera angles, or objects obstructing the view, such as trees and vehicles.

YOLOv8 produced reasonably accurate bounding boxes and maintained high inference speed, making it suitable for real-time applications and large-scale image inference. However, the model struggled with windows on slanted images or highly occluded façade s, limiting its reliability for complex urban scenes. Figures 6.12 and 6.13 illustrate training performance, loss reduction, and qualitative detection results, showing accurate predictions for vertical images but reduced performance for slanted or partially obscured windows.

Figure 9: Windows detection using YOLO V8 model

4.2.2. Grounding DINO window detection

Grounding DINO, a transformer-based object detection model integrating visual and textual prompts, was employed with the following parameters: TEXT_PROMPT = "window", BOX_THRESHOLD = 0.25, and TEXT_THRESHOLD = 0.25. The model demonstrated superior detection in challenging conditions, including small, misaligned, or occluded windows in complex façade s. Bounding box alignment was highly accurate,

and the model effectively handled diverse architectural styles, low-resolution images, and non-standard window geometries

The integration of textual prompts enabled Grounding DINO to detect windows missed by YOLOv8, providing a significant advantage in comprehensive detection tasks. While inference speed was slightly slower than YOLOv8, the model's robustness in challenging scenarios outweighed the speed trade-off.

Figure 10: Windows detection (Grounding DINO model)

4.2.3 Comparison and model selection

Table 2: Results comparison.

Model	Precision (%)	Recall (%)	mAP50 (%)	Bounding Box Accuracy	Inference Speed	Notes
YOLOv8	91.4	91.0	90.5	Good	High	Missed occluded/ slanted windows
Grounding DINO	93.2	94.1	92.8	Excellent	Moderate	Robust detection for complex/ occluded windows

Table 2 summarizes the comparative performance of YOLOv8 and Grounding DINO:

YOLOv8 excelled in inference speed and performed well for vertical, unobstructed images, but struggled under occlusions or non-standard angles. Grounding DINO, in contrast, achieved higher overall detection completeness and bounding box alignment for complex urban façades, making it more suitable for applications prioritizing detection accuracy over speed.

For this study, complete and reliable window detection was essential for downstream segmentation and parameter extraction. Therefore, Grounding DINO was selected as the preferred model, as it required no task-specific training and provided consistent detection across diverse street-view images.

4.3. Discussion

The results demonstrate that the proposed pipeline effectively integrates zero-shot window detection, optimized semantic segmentation, and parametric 3D modeling to reconstruct windows from street-view images. Several key observations can be drawn from the experiments:

Semantic Segmentation: Simplifying the label classes from 11 to three—window pane, window frame, and wall—was critical in reducing ambiguities between visually similar components. The optimized model achieved a mean Intersection over Union (mIoU) of 80.74%, with an overall accuracy of 93.29%, indicating robust delineation of window components. Fine-tuning pretrained models and increasing training iterations to 80K allowed the network to capture fine details that were previously lost with smaller crop sizes (128 px). Morphological operations further enhanced mask quality by connecting fragmented frames and removing noise, demonstrating the importance of post-processing in achieving high-quality segmentation.

Window Detection: Comparison between YOLOv8 and Grounding DINO highlighted the trade-offs between speed and detection robustness. While YOLOv8 achieved high precision (91.4%) and recall (91.0%) with fast inference, it struggled with occluded or slanted windows. Grounding DINO, using visual and textual prompts, was able to detect complex, misaligned, or partially occluded windows with higher completeness and bounding box accuracy, making it more suitable for urban façades with diverse architectural styles.

Challenges and Limitations: Despite promising results, the pipeline faces limitations when handling very low-resolution images, arched or non-rectangular windows, and extreme occlusions. The image rectification model depends on sufficient vanishing points; insufficient points can reduce rectification quality. YOLOv8, while fast, may miss occluded windows, limiting its applicability in comprehensive urban analyses.

Implications: The study shows that combining zero-shot detection with optimized semantic segmentation provides a scalable framework for façade analysis. It reduces reliance on large annotated datasets, supports diverse window styles, and enables accurate parameter extraction for 3D modeling. This modular approach is adaptable to various urban environments and can be extended for other façade components.

5. CONCLUSION

The proposed modular pipeline successfully integrates zero-shot object detection, semantic segmentation, and parametric 3D reconstruction to automate window modeling from street-view imagery. Key outcomes and contributions include:

1. **High-Accuracy Segmentation:** Optimizing class labels and fine-tuning pretrained models achieved a mean IoU of 80.74%, clearly distinguishing window panes, frames, and walls.
2. **Robust Detection with Grounding DINO:** Zero-shot detection outperformed YOLOv8 for occluded and complex façades, highlighting the effectiveness of vision-language models in architectural analysis.
3. **Enhanced Image Rectification and Mask Refinement:** Vanishing-point-based rectification and morphological operations ensured accurate geometry and spatial consistency.
4. **Parametric 3D Modeling:** Extracted parameters (rows, columns, pane dimensions, frame thickness) enabled scalable and accurate 3D reconstruction using Blender, preserving topological and spatial relationships.
5. **Modularity and Scalability:** The pipeline is adaptable to different window architectures and can be extended for additional façade elements or urban-scale digital twin generation.

Future Work: To further enhance applicability, future research should focus on low-resolution and non-rectangular windows, complex occlusions, lighting variations, and multi-view integration. Extending the pipeline to fully automate diverse façade modeling will increase its value for urban digital twin frameworks and architectural analysis.

In summary, this study demonstrates that combining zero-shot detection, optimized semantic segmentation, and parametric modeling provides a scalable and accurate solution for automated window reconstruction, bridging the gap between street-view imagery and urban digital twins.

REFERENCES

Carreira, J., Caseiro, R., Batista, J., & Sminchisescu, C. (2012). Semantic segmentation with second-order pooling. In *European conference on computer vision* (pp. 430-443). Berlin, Heidelberg: Springer Berlin Heidelberg.

Dong, J., Chen, Q., Yan, S., & Yuille, A. (2014). Towards unified object detection and semantic segmentation. In *European Conference on Computer Vision* (pp. 299-314). Cham: Springer International Publishing.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., & Ramanan, D. (2009). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9), 1627-1645..

Hao, S., Zhou, Y., & Guo, Y. (2020). A brief survey on semantic segmentation with deep learning. *Neurocomputing*, 406, 302-321..

Hu, H., Zhang, Y., & Zhu, Q. (2022). Efficient procedural modelling of building façades based on windows from sketches. *The Photogrammetric Record*, *37*(179), 272–295.

Kelly, T., Femiani, J., & Wonka, P. (2024). Winsyn: A high resolution testbed for synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 22456-22465).

Liu, Y., Bai, X., Wang, J., Li, G., Li, J., & Lv, Z. (2024). Image semantic segmentation approach based on DeepLabV3 plus network with an attention mechanism. *Engineering Applications of Artificial Intelligence*, 127, 107260.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).

Ohta, Y. I., Kanade, T., & Sakai, T. (1978, November). An analysis system for scenes containing objects with substructures. In *Proceedings of the Fourth International Joint Conference on Pattern Recognitions* (pp. 752-754). Piscataway: Institute of Electrical and Electronics Engineers Incorporated.

Pang, Y., & Biljecki, F. (2022). End-to-end building façade reconstruction from panoramic street view images. *ISPRS Journal of Photogrammetry and Remote Sensing,* 187, 146–163.

Rahman, S., Khan, S. H., & Porikli, F. (2020). Zero-shot object detection: Joint recognition and localization of novel concepts. *International Journal of Computer Vision*, 128(12), 2979-2999.

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Sasank, C., & Mittal, A. (2020). Rectification of building façades using a deep learning-based vanishing point detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision* (WACV).

Sasank, C., & Mittal, A. (2020). Rectification of building façades using a deep learning-based vanishing point detection. *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Sun, W., & Wang, R. (2018). Fully convolutional networks for semantic segmentation of very high resolution remotely sensed images combined with DSM. *IEEE Geoscience and Remote Sensing Letters*, 15(3), 474-478.

Wang, Y., Li, S., & Li, W. (2019). Deep learning-based backbone network for feature extraction in semantic segmentation. *IEEE Access*, 7, 12345-12356.

Yang, Z., Peng, X., & Yin, Z. (2020). Deeplab_v3_plus-net for image semantic segmentation with channel compression. In *2020 IEEE 20th International Conference on Communication Technology (ICCT)* (pp. 1320-1324). IEEE.

Yu, H., Yang, Z., Tan, L., Wang, Y., Sun, W., Sun, M., & Tang, Y. (2018). Methods and datasets on semantic segmentation: A review. *Neurocomputing*, 304, 82-103..

Author's Information

Name	:	Sumeer Koirala
Academic Qualification	:	MSc Geographic information system and science, MSc Land management and Geospatial sciences
Organization	:	Survey Department
Current Designation	:	Chief Survey Officer