Chair of Computational Modeling and Simulation
TUM School of Engineering and Design
Technical University of Munich

Innovative Approaches to Semantic Segmentation in Construction Sites:

Combining New Dataset with Semi-Supervised and Zero-Shot Learning

Scientific work to obtain the degree

Master of Science (M.Sc.)

at the TUM School of Engineering and Design
of the Technical University of Munich.

Supervised by Prof. Dr.-Ing. André Borrmann
Miguel Arturo Vega Torres M.Sc.
Chair of Computational Modeling and Simulation

Submitted by Shaowen Qi [redacted]

Submitted on June 1, 2024

Abstract

This thesis addresses the challenge of enabling automatic on-site data acquisition in Building Information Modeling (BIM) by developing new datasets and exploring efficient scene understanding algorithms. The primary objectives are divided into two main directions: creating a dataset specific to construction environments and exploring semi-supervised learning algorithms to enhance scene understanding.

The research begins by identifying the types of data necessary for accurately interpreting construction site scenes and streamlining the creation of high-quality segmentation data. A new dataset is generated using RGB images from the ConSLAM Sequence 2, annotated with segments of construction-related objects. Additionally, a semi-supervised learning workflow, RTMDet-SAM, is proposed to generate pseudo labels, enhancing model training without extensive manual labeling.

Experiments demonstrate the effectiveness of the proposed workflows, with the pseudo labels generated by RTMDet-SAM enabling superior recall and generalization performance for Mask R-CNN compared to Mask R-CNN trained without pseudo labels. The Average Recall (AR) increases 2.5%. Besides, the confidence scores of the inferred segments are improved up to 55% in some cases. The zero-shot approach, leveraging Grounding DINO, shows promise in generating pseudo labels with minimal manual intervention, although it requires further optimization.

The contributions of this research include the development of a new annotated dataset, the introduction of a semi-supervised learning workflow, and insights into the potential of zero-shot learning for scene understanding in construction environments. These advancements pave the way for more efficient and automated BIM practices, reducing the labor and costs associated with manual data collection for updating BIM.

By integrating advanced computer vision algorithms with BIM, this thesis aims to enhance the automation of the on-site data acquisition processes in construction projects, ultimately contributing to the broader adoption and development of BIM technologies.

Zusammenfassung

Diese Dissertation befasst sich mit der Herausforderung, die automatische Datenerfassung vor Ort im BIM zu ermöglichen, indem neue Datensätze entwickelt und effiziente Algorithmen zur Szenenverständnis erforscht werden. Die Hauptziele sind in zwei Hauptbereiche unterteilt: die Erstellung eines spezifischen Datensatzes für Bauumgebungen und die Untersuchung semi-supervisierter Lernalgorithmen zur Verbesserung des Szenenverständnisses.
Die Forschung beginnt mit der Identifizierung der Datentypen, die für die genaue Interpretation von Baustellenszenen erforderlich sind, und der Optimierung der Erstellung hochwertiger Segmentierungsdaten. Ein neuer Datensatz wird unter Verwendung von RGB-Bildern aus der ConSLAM Sequenz 2 erstellt, die mit Segmenten von baubezogenen Objekten annotiert sind. Zusätzlich wird ein semi-supervisierter Lernworkflow, RTMDet-SAM, vorgeschlagen, um Pseudo-Labels zu generieren und das Modelltraining ohne umfangreiche manuelle Kennzeichnung zu verbessern.
Experimente zeigen die Wirksamkeit der vorgeschlagenen Workflows, wobei die durch RTMDet-SAM generierten Pseudo-Labels eine überlegene Recall- und Generalisierungsleistung für Mask R-CNN ermöglichen, verglichen mit Mask R-CNN, das ohne Pseudo-Labels trainiert wurde. Die AR steigt um 2,5%. Zudem verbessern sich die Vertrauenswerte der abgeleiteten Segmente in einigen Fällen um bis zu 55%. Der Zero-Shot-Ansatz, der Grounding DINO nutzt, zeigt Potenzial bei der Generierung von Pseudo-Labels mit minimalem manuellen Eingriff, erfordert jedoch weitere Optimierung.
Die Beiträge dieser Forschung umfassen die Entwicklung eines neuen annotierten Datensatzes, die Einführung eines semi-supervisierten Lernworkflows und Einblicke in das Potenzial des Zero-Shot-Lernens für das Szenenverständnis in Bauumgebungen. Diese Fortschritte ebnen den Weg für effizientere und automatisierte BIM-Praktiken, wodurch der Arbeitsaufwand und die Kosten für die manuelle Datenerfassung zur Aktualisierung von BIM reduziert werden.
Durch die Integration fortschrittlicher Computer-Vision-Algorithmen mit BIM zielt diese Dissertation darauf ab, die Automatisierung der Datenerfassungsprozesse vor Ort in Bauprojekten zu verbessern und letztlich zur breiteren Akzeptanz und Weiterentwicklung von BIM-Technologien beizutragen.

Contents

List of Figures

List of Tables

List of Acronyms

AEC	Architecture, Engineering and Construction
AR	Average Recall
BIM	Building Information Modeling
CNN	Convolutional Neural Network
CVAT	Computer Vision Annotation Tool
DCT	Discrete Cosine Transform
DPM	Deformable Part Model
DT	Digital Twin
FCN	Fully Convolutional Network
HOG	Histogram of Oriented Gradients
IoU	Intersection over Union
LiDAR	Light Detection and Ranging
mAP	Mean Average Precision
MSE	Mean Squared Error
NIR	Near-Infrared
NLP	Natural Language Processing
NMS	Non-maximum Suppression
OVD	Open-Vocabulary Object Detection
R-CNN	Region-based Convolutional Neural Network
RFID	Radio Frequency Identification
RLE	Run-Length Encoding
RoI	Region of Interest
RPN	Region Proposal Network
SAM	Segment Anything Model
SGD	Stochastic Gradient Descent
SHM	Structural Health Management

SLAM Simultaneous Localization And Mapping

VRAM Video Random Access Memory

Chapter 1

Introduction

1.1 Motivation of the Research

In contemporary construction projects, BIM has been widely adopted. However, due to various complex reasons, such as the lack of expertise, standardization, and protocols, practitioners often use BIM merely as a static 3D model without further exploration (Hamma-adama et al., 2020). The underdevelopment of BIM applications can be partially attributed to the lack of automatic data acquisition methods, which are essential components of BIM practice protocols.

Enabling 4D-BIM necessitates the continuous collection of the on-site field data, a process that is both labor-intensive and costly if performed manually. To address this, robots could be employed to collect data on structural components, register, and update the status of corresponding elements in BIM. However, for this to be feasible, a robust mechanism for robots to recognize their environment and the target objects is essential.

In recent years, researchers in computer vision and machine learning have proposed a variety of algorithms aimed at accurate localization and classification. Machine learning algorithms, in particular, have demonstrated superior performance over conventional geometry-based algorithms, excelling across a wide range of multimodal data. This success suggests significant potential for creating new workflows for automatic data acquisition on construction sites using robots.

Despite the promising performance, these novel algorithms also pose challenges. Deep learning algorithms are typically validated on datasets designed for generic contexts, which may not be suitable for Architecture, Engineering and Construction (AEC)-specific scenarios. While these models can easily distinguish between a truck and a car, they may struggle to differentiate between a pile of fine aggregates and cement without custom adjustments.

Therefore, developing a self-trained, specialized model is crucial. Such a model requires specific data, and there is a significant shortage of datasets related to structures captured at various phases of construction compared to the abundance of algorithms. Moreover, well-annotated datasets that can be applied to this context are even rarer.

To address the problem caused by the lack of data, two dimensions of methods would be helpful. On the one hand, creating more well-annotated datasets specific to the AEC industry is essential. The performance of machine learning algorithms largely depends on high-quality datasets that can cover a broad range of the input space. Inadequate training data typically results in an underfitting model, while data that is unevenly distributed

or lacks variability can lead to an overfitting model. Both outcomes are detrimental to developing a robust machine learning model.

On the other hand, the usage of datasets specific to the AEC industry, especially visual data from construction environments, is more limited compared to datasets captured in more generic scenes. There are relatively fewer researchers focusing on AEC-specific topics, and companies are less motivated to invest in computer vision research that targets this particular field. These factors collectively contribute to the scarcity of datasets available for developing robust automatic on-site data acquisition workflows. Given the shortage of data, it becomes increasingly important to optimize scene understanding algorithms to fully leverage the valuable datasets that are available. Although modeling the relationship between visual data and semantic information is primarily a supervised learning task, the intensive dependency on labeled data can be alleviated by semi-supervised learning. Algorithms such as consistency regularization, pseudo-labeling, and self-learning can leverage unlabeled data to improve model performance. These methods hold great potential for facilitating the development of an ideal mechanism for automatic on-site data acquisition, even with limited training data.

Hence, this exploration will present two directions: new data and new methods. It is hoped that this work will contribute to the advancement of a more optimal workflow for the utilization of on-site data in construction projects.

1.2 Objectives of the Research

As mentioned above, the main objective is divided into two directions: creating a new dataset and exploring new algorithms.

For Creating a New Dataset:

- Identifying the specific types of data necessary to accurately interpret and understand scenes in construction environments.
- Exploring methods and tools to streamline the creation of high-quality segmentation data, minimizing time and effort.
- Generating more data of the same quality as ground truth without manual intervention.

For Exploring New Algorithms:

- Evaluating the performance of the segmentation model under conditions with limited labeled data.
- Examining the impact of pseudo labels generated through the semi-supervised method on the ability of the segmentation model to generalize and enhance accuracy.
- Exploring techniques to further automate the generation of pseudo labels, reducing the need for manual intervention.

1.3 Reading Guide

- chapter 2 explains key concepts in machine learning and computer vision.
- chapter 3 reviews existing datasets relevant to construction sites and algorithms for scene understanding.
- chapter 4 details the creation of a new dataset and the development of two experimental workflows.
- chapter 5 describes the experimental setup and procedures, including hardware, software, and data processing techniques.
- chapter 6 presents the results and analysis by evaluating the performance of the proposed workflows.
- chapter 7 discusses the findings, contributions, limitations, and future research directions, concluding the study.

Chapter 2

Theoretical Basics

In this chapter, the fundamental ideas involved in the research of the thesis will be presented. Firstly, the characteristics of different types of machine learning are compared with semi-supervised learning prevailing in the context of this thesis. Then two computer vision tasks involved in the proposed workflows are explained.

2.1 Supervised, Unsupervised & Semi-supervised Learning

Machine learning is nowadays one of the most ubiquitous technologies in data science. It covers a wide range of algorithms that focus on revealing the intrinsic characteristics of the data. According to the objectives of the specific machine learning tasks, and the data types required, the machine learning algorithms can be divided into mainly two categories: supervised learning and unsupervised learning.

Supervised learning is characterized by the use of labeled data. the overall objective is to model the relationship between the input and output data, i.e., construct a mapping method from the features to the target (Alloghani et al., 2020).

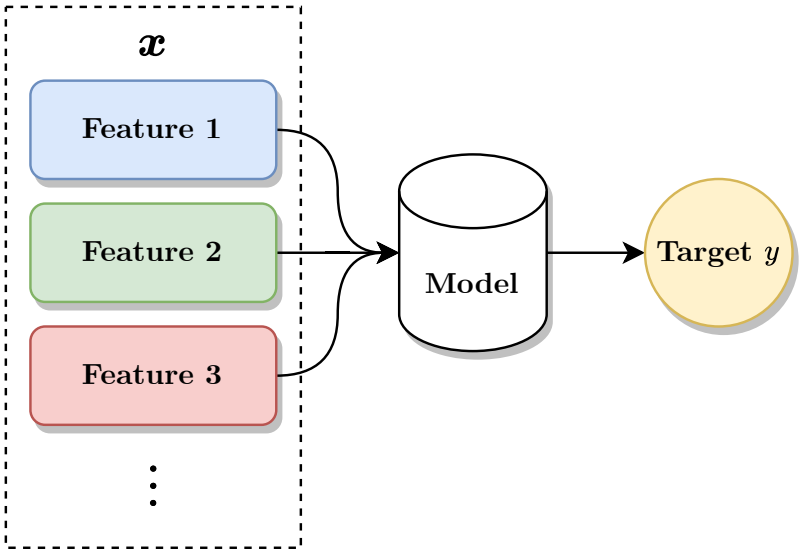


Figure 2.1: Mechanism of Supervised Learning

In the dataset used for capturing this relationship, both the input data $\boldsymbol{x} \in \mathcal{X}$, where $\mathcal{X}$ is the input space, and the output data y are provided. The predetermined outputs are applied as guidance in the learning process. In most cases, the parameters of the model can not be determined with analytical solutions. Hence, the numerical solutions are acquired

by optimization. The parameters are consecutively updated according to the feedback from the calculated error term. The estimated parameters are eventually determined until the global minimum of the error term is reached.

With this general idea, the implementations of the model are manifold. From the most basic linear regression with one parameter for each feature to the neural network containing millions of parameters, all those algorithms are regarded as supervised learning.

Unsupervised learning is contrary to supervised learning, it requires no additional information other than the input features. This set of algorithms is dedicated to discovering the intrinsic patterns and structures of the input features. The typical applications of unsupervised learning include clustering, anomaly detection, and dimensionality reduction. These tasks are solely based on the input features and can even create label data for further supervised learning workflow (Alloghani et al., 2020).

Due to the absence of label data, the objective of the task is less defined, i.e., the output of the algorithms may not match the demand. This requires a larger effort in tuning the hyperparameters of the algorithms. But this character also means the burden of data labeling, which is usually laborious for large datasets, can be drastically lessened.

Semi-supervised learning is a combined method of both the idea of supervised and unsupervised learning. In this method, a relatively small portion of the dataset is labeled, and the objective of the machine learning task is defined and restricted by the labeled subset. Besides, the unlabeled subset makes up the majority and augments the learning process (van Engelen & Hoos, 2020). The most classic implementation of this augmentation is pseudo-training. In general, this implementation uses the labeled data to train a "base" model, and the unlabeled data is subsequently fed to the model to generate the pseudo labels. Finally, the combined dataset containing true and pseudo labels is used to train another model. This process can be done iteratively, and the performance would be enhanced after each iteration (Sharma, 2023).

Semi-supervised learning is especially effective in computer vision tasks, in which the raw data could be abundant while labeled data is scarce. Thus, the performance of image classification, object detection and semantic segmentation could benefit from the full utilization of the dataset.

2.2 Object Detection Task

2.2.1 Objective of Object Detection

Object detection is a longstanding and vital topic in the research of computer vision. Its primary task is to localize and identify ideally all the instances in the images or video frames. The instances are predefined with several classes (Amit et al., 2021).

This task originated at a time when machine learning algorithms were not yet popular. The initial research focused mainly on identifying the geometric features of the objects. Those methods were largely dependent on good feature extraction and sensitive to variation. These factors result in poor generalization. The thriving machine learning methods in later years patched the drawbacks and quickly became the common practice in the object detection task.

2.2.2 Dataset Format of Object Detection

In most cases, the object detection task is performed with supervised learning methods. That is to say, the required data has two parts, the input feature and the output label. For the object detection, the input data are 24-bit RGB images. This is the most common format for colored images, which contains 3 channels representing red, green and blue. Each channel has a color depth of 8 bits, i.e. each channel is capable of displaying $2^8 = 256$ distinct color values. In summary, a single input data point is a tensor with the shape of $C \times H \times W$, in which the number of channels $C = 3$, H and W represent the height and width of the image. The value of the elements in the tensor could take integers from 0 to 255. This provides sufficient feature space to contain the information used for inferring the target output.

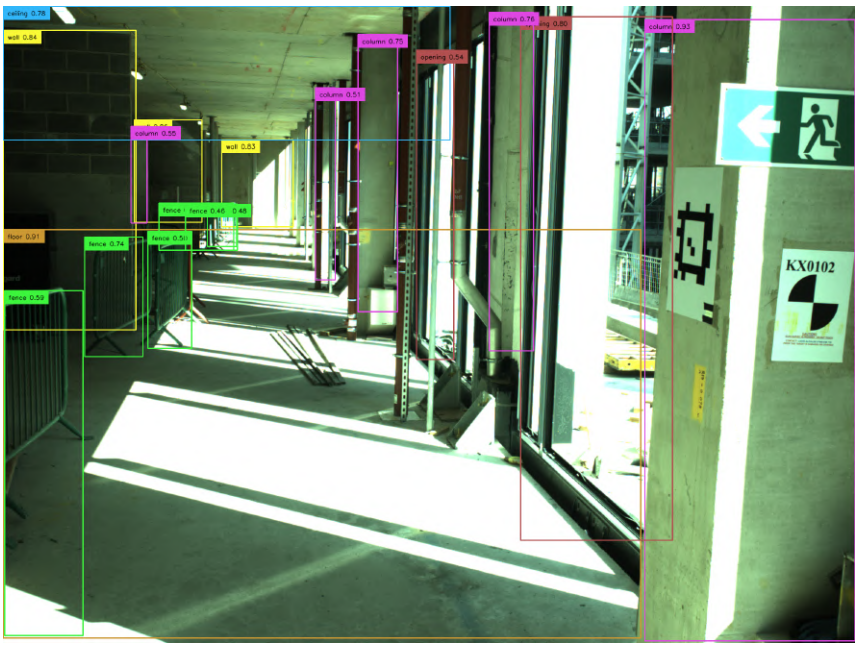


Figure 2.2: Example of Image with Bounding Boxes

The output data are the labels of objects and the location information of the corresponding objects. The labels of the objects are stored in the form of the unique integers assigned to each class. During the learning process of the algorithms, the labels are usually converted into ont-hot coding, which fits better with the shape of the output of algorithms like neural networks. The location of the objects is defined with bounding boxes. These are the rectangular frames prompting both the location and scale of the objects. In common

practice, the bounding boxes are defined with 4 values, which indicate two coordinates of a specific vertex, height and width of the bounding boxes. Figure 2.2 shows a sample image with bounding boxes.

2.2.3 Metrics for Evaluation

2.2.3.1 Loss Functions

The loss function is a universal approach for measuring the misalignment between the prediction from the current model and the true target of the training data. This metric reflects the converging progress of a supervised learning model. For different sorts of tasks, a variety of loss functions are applied, e.g., Mean Squared Error (MSE), cross-entropy, and ℓ_1 loss. Although the error of the algorithm is retrieved by performing calculations with the designated loss function, additional information used for monitoring the learning process is still necessary. The loss calculated in the training phase indicates only the ability of the model to fit the training data, the ability of generalization needs to be assessed with fresh data. This method could prevent overfitting of the model. Besides the loss functions of the localization and classification stages, some extra task-specific metrics are introduced to monitor whether the objectives of the task are properly achieved.

2.2.3.2 Intersection over Union

Intersection over Union (IoU) is an intuitive method for measuring the performance of object detection algorithms. It is also known as the Jaccard index and is commonly used for comparing the similarity of two arbitrary shapes. It is defined by the fraction of the area of the intersection and union of two shapes, as illustrated in Figure 2.3. This metric is normalized to eliminate the influence of geometric attributes of the shape. Hence, it focuses solely on the degree of overlapping of two shapes without the consideration of scale and location of the shapes (Rezatofighi et al., 2019).

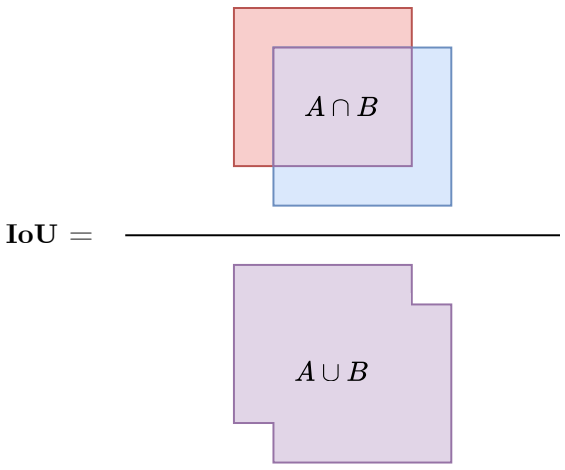


Figure 2.3: Definition of IoU

The larger value of this fraction hints at higher accuracy of the prediction. Due to this correlation and the scale-invariant characteristics, IoU is made a suitable criterion for discriminating true positives and false positives. This is fundamental for further calculations of other metrics, because IoU plays the role of a threshold for judging the correctness of a single prediction.

2.2.3.3 Mean Average Precision

Mean Average Precision (mAP) is one of the key concepts for measuring the performance of the model. *mean* and *average* may appear to be duplicated, it actually means 2 times averaging happened in the calculating process. The precision of predictions is defined in Equation 2.1.

$$\text{Precision} = \frac{|\text{True Positives}|}{|\text{True Positives}| + |\text{False Positives}|} \tag{2.1}$$

During the evaluation stage, a certain IoU value is selected. Under this condition, the precision and recall are calculated pairwise according to different probability thresholds of the predictions. Higher precision usually means lower recall. This trade-off results in the correlation of precision and recall, which is described by precision-recall curve (Henderson & Ferrari, 2017; Zhu, 2004), as illustrated in Figure 2.4. It is noteworthy that different choices of IoU value incur different precision-recall curves.

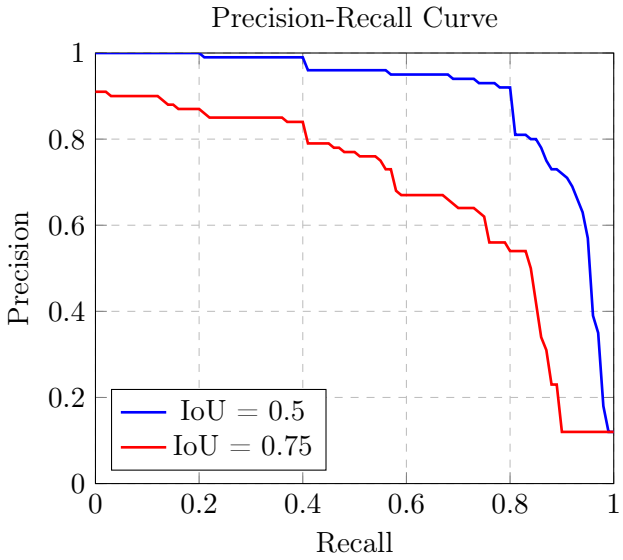


Figure 2.4: Demonstration of Precision-Recall Curves under Different IoU

In order to consider both the precision and recall of the predictions, the average precision is acquired by weighted averaging the precision under different recall values as described in Equation 2.2, where p stands for precision and r stands for recall. Object detection includes performing multi-class classification, thus multiple average precision values are acquired from the individual calculation for each class. Finally, the overall performance of

the model can be represented by a metric—mAP, which is easily computed by averaging amongst all classes (Henderson & Ferrari, 2017), as shown in Equation 2.3, where K refers to the total number of classes (Padilla et al., 2020).

$$\text{AP} = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p(r_{i+1}) \tag{2.2}$$

$$\text{mAP} = \frac{1}{K} \sum_{k=1}^{K} \text{AP}_k \tag{2.3}$$

2.2.3.4 Average Recall

Similar to the concept of mAP, AR is a metric emphasizing the other side of the performance. While mAP mainly measures the correctness of the predictions from the model, AR focuses on assessing the completeness of the predictions, i.e., the ratio of correctly detected objects in the total number of objects in the ground truth. This is an essential metric for models concerning rare events, such as detecting structural defects. These tasks generally require high recall performance. The recall is a ratio as defined in Equation 2.4. Similarly, a specific IoU value is selected before computing the recall value.

$$\text{Recall} = \frac{|\text{True Positives}|}{|\text{True Positives}| + |\text{False Negatives}|} \tag{2.4}$$

This value is calculated on the basis of a single class and a fixed IoU. To obtain the evaluation of overall performance, the recall values across all the classes and multiple IoU values are averaged, as demonstrated in Equation 2.5, in which K stands for the number of classes, and M stands for the number of IoU values involved in the calculation. It is noteworthy that the maximum number of predictions per image is fixed during the calculation (Padilla et al., 2020).

$$\text{AR} = \frac{1}{K \cdot M} \sum_{k}^{K} \sum_{m}^{M} Recall(k, \text{IoU}_m) \tag{2.5}$$

2.3 Semantic Segmentation Task

2.3.1 Objective of Semantic Segmentation

As another category of scene understanding algorithms, semantic segmentation goes deeper in mining the semantic information of the images. While object detection requires only the envelop bounding box, which provides 4 values per instance, semantic segmentation generates a label for each pixel. Figure 2.5 compares the output of object detection and semantic segmentation on the same input image.

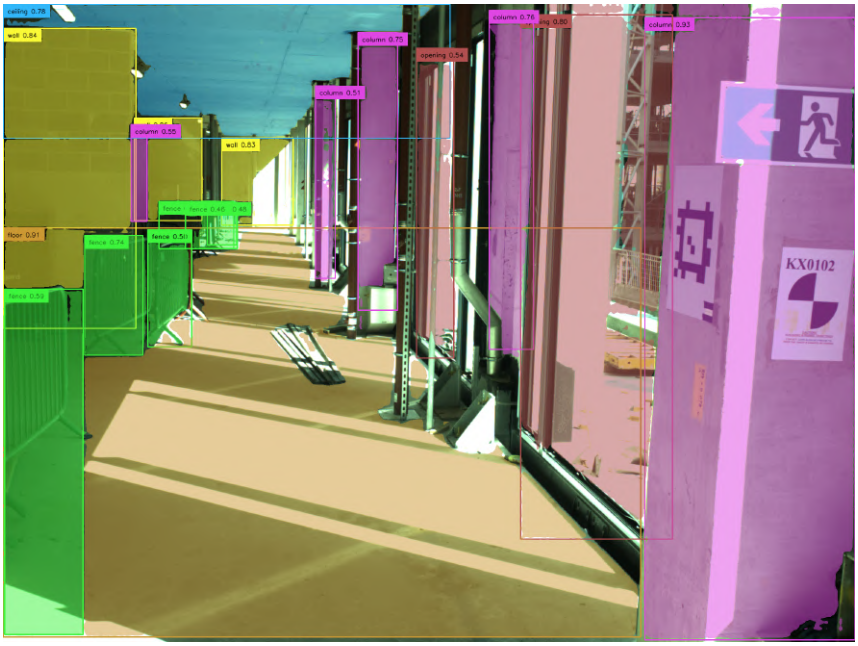


Figure 2.5: Comparison Between the Outputs

The semantic segmentation brings not only richer information about the objects in the scenes, but also enormous challenges to the algorithms and data preparation (Feng et al., 2020; Shotton & Kohli, 2020). First of all, the ambiguity of the boundaries of the detected objects is much lower compared to object detection. The output of the model contains geometric information besides the location and scale. This enables better scene understanding, which could be useful in many scenarios, such as precise registration for various structural components. Secondly, the complexity of data labeling for semantic segmentation would be drastically higher than it is for object detection. The pixel labeling is laborious and prone to error, while the bounding box labeling requires only 4 values for each object. Thirdly, the shapes of the output differ greatly. Object detection creates 4 values for a bounding box, but the semantic segmentation needs to create a binary mask with an identical shape to the input image. This feature poses a larger computational burden on the semantic segmentation model. Despite the challenges, the semantic segmentation algorithms still thrive in the research of autopilot, robotics, medical imaging, etc. (Datagen, 2023) It also has great potential in the implementation of 4D-BIM for providing semantically rich data used for model update.

2.3.2 Dataset Format of Semantic Segmentation

As two similar tasks target the same type of data, the semantic segmentation algorithms require the same input data as object detection. However, the label data needs to indicate not only the location and scale of certain objects, but also describe the geometry of it (Shotton & Kohli, 2020). Several methods are developed to efficiently encode and store the geometric information of the objects. The most primal and intuitive approach is to create color masks. By this approach, the objects in each image are denoted by segments with colors assigned to the corresponding class. This format matches the typical output of

a semantic segmentation model after post-processing. It is readable for humans, and easy to troubleshoot when errors are introduced during the data preparation. This format is further developed into a more concise representation. The RGB channels are substituted by a single channel, storing the class ID of the pixels. This method is widely used by the datasets published on Huggingface, which provides a convenient API for retrieving this format and loading to the model (Lhoest et al., 2021). The binary masks split the multi-class masks into multiple masks where the single mask distinguishes only objects of a certain class and the background. To further reduce the redundancy, Run-Length Encoding (RLE) is introduced to losslessly compress the sparse matrix like binary masks (Birajdar et al., 2019). After this processing, the geometric information is saved in an array of values, which can be conveniently stored in text files (Lin et al., 2014). In some cases, the segments are stored in the form of polygons with labels, the per-pixel labels are generated on the fly while processing.

2.3.3 Metrics for Evaluation

IoU as Standalone Metric is utilized for the semantic segmentation. The metrics used for evaluating object detection tasks are also applicable with some nuances. the concept of IoU is now defined with the intersection and union areas of the overlapping segments from the prediction and ground truth. The IoU serves not only as the threshold for distinguishing true and false prediction, but also as a metric that directly indicates the accuracy of the model. In the evaluation defined by PASCAL VOC Challenge, the IoU is defined as Equation 2.6 shown (Everingham et al., 2015), in which $|\cdot|$ refers to the number of pixels matching the condition.

$$\text{IoU} = \frac{|\text{TP}|}{|\text{TP}| + |\text{FP}| + |\text{FN}|} \tag{2.6}$$

The CityScape Dataset improved this criterion to overcome the bias of unweighted IoU, in which the successful prediction of smaller segments is not additionally rewarded. The IoU value is amplified by multiplying the ratio of the average instance size to the size of the respective ground truth. This instance-level IoU is defined in Equation 2.7. It's noteworthy that the false positives are not penalized for not genuinely associated with any class. (Cordts et al., 2016).

$$\text{iIoU} = \frac{|\text{iTP}|}{|\text{iTP}| + |\text{FP}| + |\text{iFN}|} \tag{2.7}$$

Loss, Precision & Recall metrics applied during the evaluation of the object detection are still meaningful while training the semantic segmentation model. Jaccard loss defined based on IoU, which is easily calculated by $\mathcal{L} = 1 - \text{IoU}$ is additionally available for semantic segmentation task (Duque-Arias et al., 2021). The precision and recall performance of the model is similarly assessed by metrics defined in Equations 2.2 to 2.5

Chapter 3

Related Work

3.1 Dataset Concerning Construction Sites

3.1.1 Available Data Sources

3.1.1.1 Read-Time Data

The implementation of BIM brings various new application scenarios in every phase of the project, such as construction coordination, clash detection, schedule management, facility management, energy analysis, etc. Those applications are enabled by the data derived from the integrated multimodal digital assets from BIM (Sacks et al., 2018). Thus, it has proposed significantly higher requirements for the collection and storage of data.

Research carried out by Davtalab (2017) indicates the necessity of adopting real-time data in the implementation of BIM in the facility management process. The maintenance and daily operation are made more efficient by saving up to 80% of the time. Besides the post-construction stage, J. Wang et al. (2015) also find that real-time data could play a critical role in the quality control process of the construction project. The potential structural defects can be efficiently identified by real-time data-aided BIM.

3.1.1.2 Data from Sensors

Despite the potential of real-time data facilitating the refined management with BIM, acquiring data with high granularity poses a tremendous challenge to stakeholders. A typical source of such data is the sensors embedded in structural elements during the construction phase shown in Figure 3.1. These sensors include strain gauges, displacement sensors, inclinometers, and corrosion sensors, which provide stable data flow in the regular time interval, serving Structural Health Management (SHM). However, the drawbacks are also prominent. Sensors are installed on the structural components spreading around the building, these sophisticated sensors require regular calibration to maintain accuracy. This causes a higher workload and needs a higher skill level of maintainers. Besides, inclinometers and corrosion sensors are intrusive for structural components. Their installation can only be scheduled in the construction phase or needs to risk deteriorating the components. These sensors mainly focus on the direct measurement of metrics about structural dynamics and chemical states. They can provide accurate data reflecting structural health but with rather high costs.

(a) (b) (c)

Figure 3.1: Sensors generating real-time data: (a) strain gauges (Geotechnical Observations, 2024b), (b) inclinometers (Geotechnical Observations, 2024a), (c) corrosion sensors (Ramón et al., 2022)

3.1.1.3 Data from GPS & RFID

To compensate for the deficiency of those sensors, more sources of data are introduced as supplements. To fulfill the requirements of indirect measurement of structural health metrics, the auxiliary data, such as the location of objects or personnel is necessary. These data also enrich the parameters of elements in BIM, which is beneficial to other tasks besides SHM during the life cycle of the building.

The research on GPS-based SHM systems has become increasingly popular in recent years. This technology facilitates the monitoring of structure dynamics (Kaloop et al., 2017). And the potential of applying Radio Frequency Identification (RFID) in construction sites is also explored by researchers in several aspects. The GPS and RFID are mutually complementary and form a robust system for the localization of personnel and equipment. They improve the efficiency of project management and lower the cost during the construction phase (Andoh et al., 2012). More notably, by tracking the real-time location of personnel and equipment, more accidents can be prevented, and the safety of workers is thus optimized (H.-S. Lee et al., 2012). Besides the earlier phases of projects, this technology can also be helpful in the maintenance phase. It enhances the automation level of facility management, saving up to 80% of the operation time and 50% of the manpower (Valero et al., 2015).

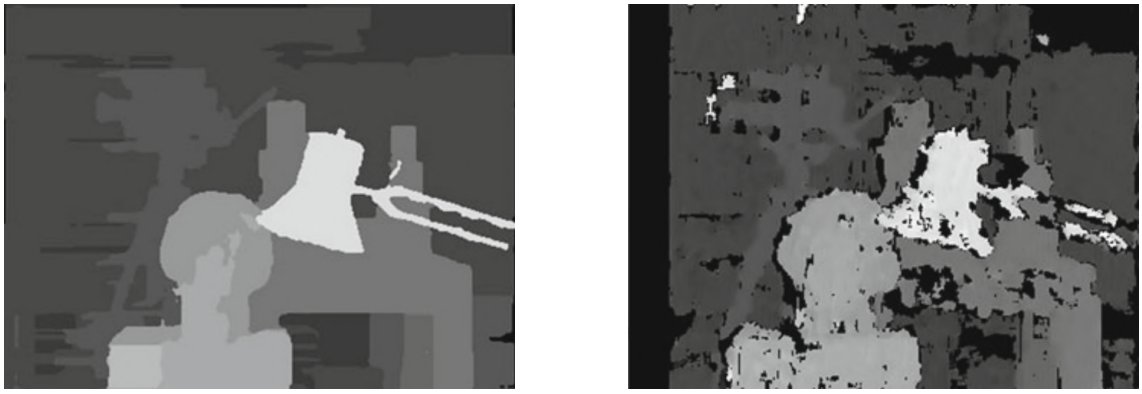
3.1.1.4 Visual Data

Besides, visual data is another essential data source. To achieve the concept of Digital Twin (DT), the elements in BIM should be accurately registered to the corresponding real-world components. The first step of it is to achieve a reliable Simultaneous Localization And Mapping (SLAM) workflow. This requires a large quantity of spatial data. Conventionally, the point cloud data captured by laser scanners is employed. Laser scanners as a sort of surveying equipment, can generate accurate point cloud data by scanning the surrounding environment with laser beams. This hardware usually needs to be firmly installed and well-calibrated. As shown in Figure 3.2, the laser scanner is installed on a tripod to ensure stability. Although the accuracy brought by surveying laser scanners is unparalleled, the cost of this solution could be intolerably high for many companies. As a novel technology, it requires highly trained operators. This could also increase the cost of the project (Ellis,

2023). Besides, the environment of the construction sites could be complex. The device is cumbersome and can hardly be installed in some narrow spaces. And its strict requirements also hinder the automation of the SLAM workflow. Thus, the acquisition of data using laser scanners is an unbearable solution for stakeholders.

Figure 3.2: ScanStation from Leica (Leica Geosystems, 2024)

In addition to the laser scanners, the researchers from the field of computer vision provide an alternative. The depth cameras capture color information along with the depth information of pixels, i.e., the images with 4 channels: RGBD. The RGB information and depth information are simultaneously captured by depth cameras. Depth cameras include mainly 3 categories according to their mechanism: *a)* structured light and coded light, *b)* stereo depth, and *c)* Light Detection and Ranging (LiDAR) (Intel RealSense, 2020). The structured light method provides high fidelity but is sensitive to the environment and only feasible on small objects and short distances (Polyga, 2024). The stereo depth method could be used in more scenarios, but the generated depth map contains a large portion of artifacts as shown in Figure 3.3. Besides, these artifacts would be amplified by accumulation error. As a result, the reconstructed point cloud data from it would be unreliable (Kadambi et al., 2014). The LiDAR uses the same technology as surveying laser scanners. Despite the ability to be mounted on drones and the unparalleled accuracy, the disadvantages of excessive cost persist.



(a) Ground Truth (b) Stereo Depth

Figure 3.3: Comparison between ground truth and stereo depth map (Kadambi et al., 2014)

The fallback to the conventional RGB images is another viable solution. The RGB images can benefit from the abundant research on semantic scene understanding. As the easiest visual data to collect, the RGB dataset is more common on construction sites than the RGBD dataset, which can be a huge advantage for developing algorithms dedicated to construction site scene understanding. Thus, it is a robust and reliable source of real-time data, which can be used for construction site scene understanding, and thereby implemented in BIM as a data acquisition method.

3.1.2 ConSLAM Dataset

ConSLAM dataset is currently one of the most well-constructed visual datasets in the AEC industry. This dataset contains regular RGB images and Near-Infrared (NIR) images, point cloud data from hand-held LiDAR scan, inertial data, and even point cloud data from professional devices as ground truth. This dataset is not only rich in data variety, but also covers the construction phase by periodically collecting data. Although, it is a dataset intentionally used for measuring the performance of SLAM algorithms (Trzeciak et al., 2023), the image data derived from it can still be utilized in the scene understanding tasks. The corresponding synchronized point cloud data could also facilitate subsequent research based on the scenes with semantic information augmented.

The whole dataset contains 4 subsets with the same structure, as shown in Figure 3.4. Taking sequence 2 as an example, the aforementioned RGB images, NIR images, and LiDAR scan are organized in `/rgb`, `/nir`, and `/lidar` folders. Additionally, the device pose of each scan is provided in `/poses` and corresponds to the scan data in `/lidar` (Trzeciak et al., 2023).

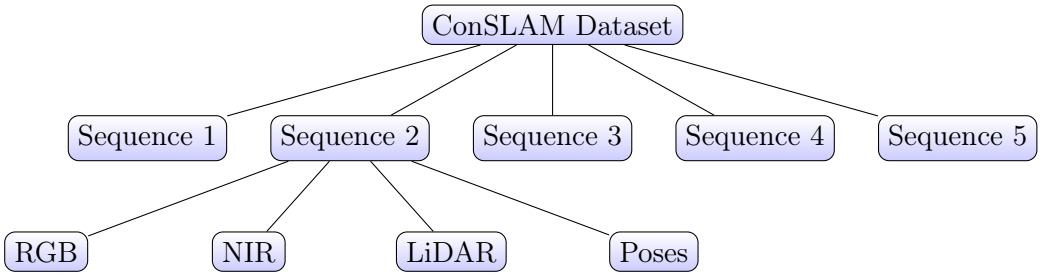


Figure 3.4: Data Structure of ConSLAM

As shown in Figure 3.6c, the hand-held LiDAR scan is fragmented data and only capable of reflecting a certain field of view. Thus, it requires the pose data to construct a whole point cloud. This necessary process is recognized as mapping. However, the data generated by the hand-held LiDAR scanner is unstable and constantly contains artifacts which could severely deteriorate the output point cloud, if the mapping algorithm is inferior. In order to evaluate the performance of the mapping algorithms, the ground truth data is necessary. The ground truth scan is generated by the land surveyor and provides maximal precision and integrity, as Figure 3.5 demonstrates.

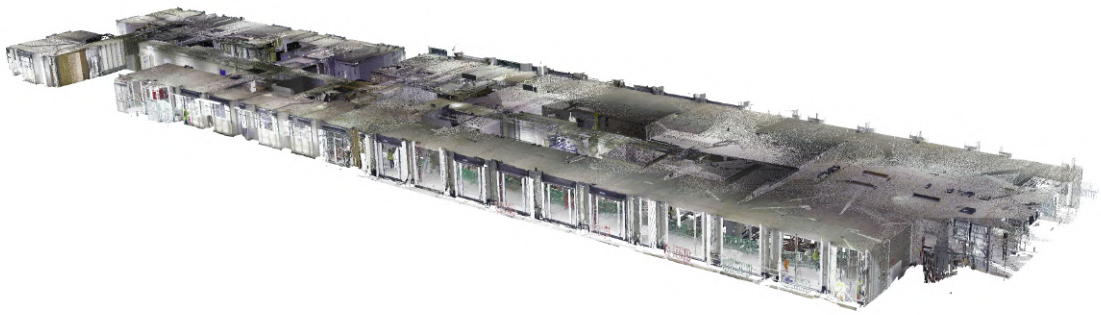


Figure 3.5: Ground truth scan of ConSLAM sequence 2 (Trzeciak et al., 2023)

The RGB images and NIR images are important parts of the dataset. These represent the most intuitive visual data. They carry rich information about the surrounding environment, while being the easiest to acquire. The size of both RGB and NIR images is 2064×1544. The RGB images have three 8-bit channels. As for the NIR images, the output of the NIR sensor is mapped to 8-bit greyscale, due to the invisibility of NIR. Figure 3.6 demonstrates the RGB image and NIR image captured while the corresponding LiDAR scan.

(a) RGB Image (b) NIR Image (c) LiDAR Scan

Figure 3.6: Variant data types in ConSLAM dataset (Trzeciak et al., 2023)

The RGB images from the ConSLAM dataset genuinely show the indoor scene of a construction site and would be utilized as the main data source in this thesis. Without loss of generality, the sequence 2 is selected for the thesis.

3.1.3 Construction Site Safety Image Dataset

Safety management is a critical topic of the construction project. Conventional safety management typically involves safety planning, manual inspection, and paper-based documentation. These methods not only make the management labor-intensive, but also archive the data on paper, making it hard to utilize. This practice is regarded inefficient (Afzal & Shafiq, 2021). Thus, the innovation of safety management is one of the most urgent parts to be achieved by BIM. In order to replace the primitive manual inspection, an automatic method for recognizing the elements on the construction sites is necessary.

The development of scene understanding algorithms relies heavily on the data concentrating on the visual elements on the construction sites. In this case, the safety-specific attributes of the visual elements should be considered. The Construction Site Safety Image Dataset from Roboflow is an excellent example of this use case. This dataset contains 717 RGB images with non-uniform resolutions. These images are collected from various sources, which is beneficial for the generalization of algorithms. Apart from the raw images, the data is well annotated with bounding boxes for the object detection task. The visual elements in the images are classified into 24 classes and mainly cover the personnel (including gloves, masks, helmets, and safety vests), and equipment (including vehicles, machinery, and fire-fighting equipment) (Roboflow Universe Projects, 2023). The annotations of the dataset are safety-oriented.

Besides this dataset, Roboflow also assembles plenty of other visual datasets dedicated to the construction site scenes, which can be easily retrieved from their platform.

3.1.4 ZInD: Zillow Indoor Dataset

ZInD is a dataset created by Zillow, a real-estate marketplace company. Benefitting from its own business, Zillow managed to acquire plenty of data regarding the indoor space. ZInD consists of 71474 panoramas from 1524 unfurnished homes. It is also a well-annotated

dataset, which contains the annotations of 3D room layouts, 2D and 3D floor plans, panorama location in the floor plan, and locations of windows, doors, and openings (Cruz et al., 2021).

BIM as a technology for building management, is meant to be implemented throughout the whole life cycle of the buildings, including the maintenance phase. However, scene understanding algorithms, especially those based on machine learning, developed on data from the construction phase, can hardly be helpful to the automation of the maintenance phase. The as-built visual data provided by datasets like ZInD can enhance the performance of the scene understanding algorithms used in this phase, and thus further complete the workflow of BIM.

3.2 Algorithms for Scene Understanding

As mentioned before, the scene understanding task is a vital prerequisite for utilizing real-time visual data in BIM. Scene understanding is the summarized overall objective of a series of tasks. This mainly consists of 3 tasks: *a)* scene classification, *b)* object detection, and *c)* semantic segmentation. The scene classification algorithms are to help to understand the subject of the image at the level of a single frame. These algorithms could enhance the ability of drones to accurately classify their surrounding environment. However, in order to fulfill the requirements of the automation in BIM, the algorithms that generate results with higher granularity are more desirable. In the workflow of this thesis, the object detection and semantic segmentation tasks would be in the scope for further discussion.

3.2.1 Object Detection Algorithms

As introduced in chapter 2, object detection achieves both localization and classification of the objects of interest in the images. Different genres of algorithms are developed. The object detection task can be split into two main objectives: localization and classification, thus some algorithms use two stages to fulfill these two objectives separately, while the others extract and classify the objects in one step (Pang & Cao, 2019). Recently, with the popularity of attention mechanism, encoder-decoder structure, and the transformer architecture derived from them, some hybrid algorithms combining traditional Convolutional Neural Network (CNN) with the new architecture emerge (Shehzadi et al., 2023).

3.2.1.1 Two-Stage Algorithms: R-CNN

The representative network for the two-stage method is Region-based Convolutional Neural Network (R-CNN), which is the initial attempt to apply convolutional layers to object detection. This architecture achieved better performance than the traditional Deformable Part Model (DPM), which uses statistical features such as Histogram of Oriented Gradients

(HOG). More optimized algorithms are later derived from the R-CNN architecture, e.g., Fast R-CNN, Faster R-CNN. (Pang & Cao, 2019).

Faster R-CNN is the one in this series that achieved the fastest performance in both training and inferring. By jointly training the Region Proposal Network (RPN) and R-CNN, about 25%-50% of the training time is reduced. During the inference, the speed of the model is drastically higher after substituting the selective search method with RPN. The frame rate of detection increases from $0.5fps$ to $5fps$ (Ren et al., 2015).

The network is composed of 4 key components:

1. CNN Backbone Layers: VGG-16 (Simonyan & Zisserman, 2014)
2. RPN Layers
3. RoI Pooling Layers
4. Fully Connected Layers

As an end-to-end object detection model, the input image can be fed into the model without preprocessing. The first step is the CNN backbone layers, which serve as the feature extraction component. In Faster R-CNN, VGG-16 is used as the backbone by default. Assuming the size of the input image is $H \times W \times 3$, the output feature map has the shape of $\frac{H}{16} \times \frac{W}{16} \times 512$. Then the RPN operates on the feature map generated by the backbone. In this process, the possible regions that contain objects are proposed. For each proposal, k different regions with nuance in shape and size are generated. This results in the output shape of $\frac{H}{16} \times \frac{W}{16} \times 2k$ for the classification layer and $\frac{H}{16} \times \frac{W}{16} \times 4k$ for the regression layer. These outputs indicate the probability of the image containing the Region of Interest (RoI), and if so, the location and size of the RoI. The redundancy in the proposals is reduced by the Non-maximum Suppression (NMS) layer. Finally, the reduced proposals are processed by the RoI pooling layer to unify the dimension of the feature map. This feature map with RoI serves as the input in the original Fast R-CNN network, producing the final localization and classification output (Ren et al., 2015). The overall diagram of the network architecture is shown in Figure 3.7.

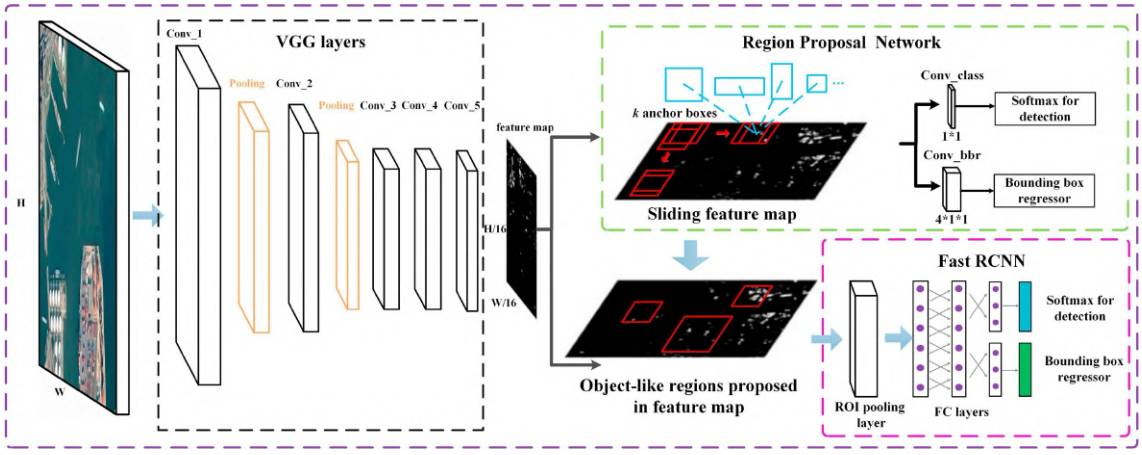


Figure 3.7: Network Architecture of Faster R-CNN (Z. Deng et al., 2018)

Although Faster R-CNN achieved high performance with relatively fast speed, the characteristic of performing two stages of detection still causes the costly computation by the per-region operations in the second stage (Z. Deng et al., 2018).

3.2.1.2 One-Stage Algorithms: YOLO

In addition to the development of algorithms for two-stage detection, algorithms with reduced computational complexity have also been created, including SSD and YOLO. Among these algorithms, YOLO has received the greatest attention. The acronym YOLO, which stands for *You Look Only Once*, indicates that localization and classification occur in a single stage of computation. This approach reduces the burden of excessive computation, but it also entails certain compromises in model performance, particularly in terms of mAP (Z. Deng et al., 2018).

The YOLO algorithm has undergone significant evolution over the past seven years, since its initial proposal in 2015. The original architecture was relatively straightforward. The YOLOv1 algorithm first divides the image into grids with a shape of $S \times S \times 3$, then generates B bounding boxes for each grid. The classification occurs at the level of each grid, generating C probability values for each grid, indicating the probability of an object in the grid belonging to a specific class (Redmon et al., 2016).

The 7th iteration of the YOLO algorithm, designated YOLOv7, integrates a multitude of novel network blocks, which facilitate enhanced feature extraction, including ELAN and SPPFCSP. Its network architecture is illustrated in Figure 3.8. YOLOv7 outperforms all real-time models ranging from $5fps$ to $120fps$. It achieves a superior performance with the mAP of 56.8%, while experimenting with MS COCO Dataset (C.-Y. Wang et al., 2023).

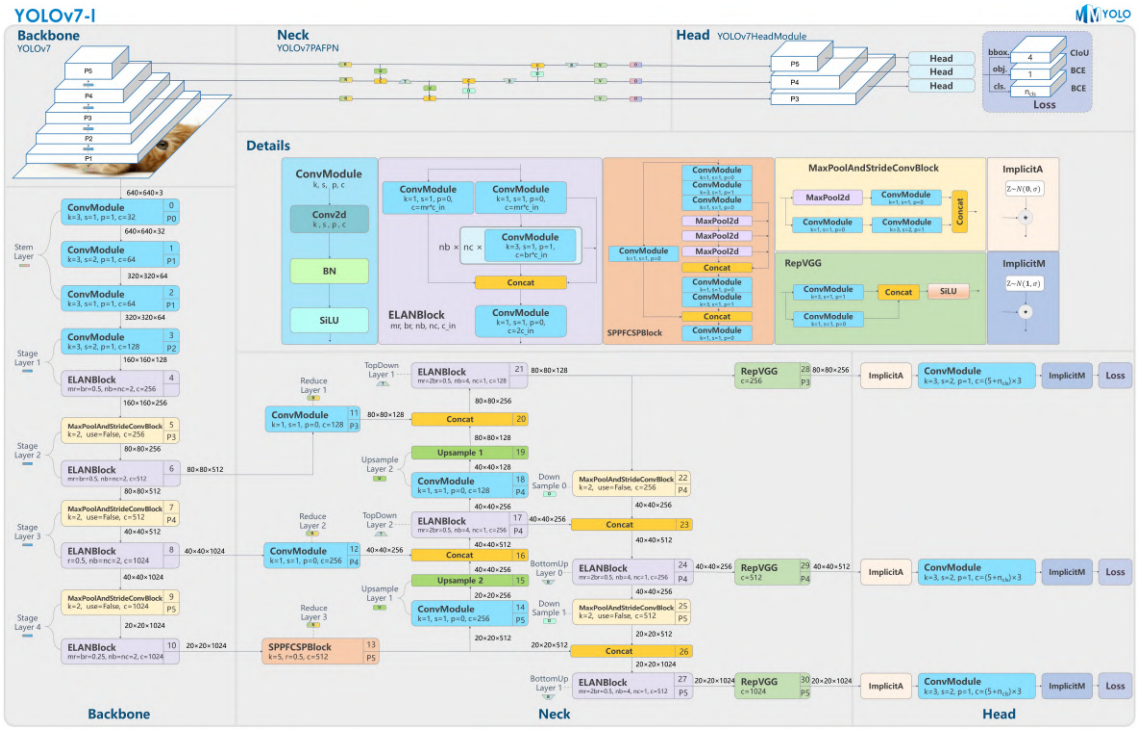


Figure 3.8: Network Architecture of YOLOv7 (MMYOLO Contributors, 2022)

To enhance the real-time performance of one-stage object detection models, Lyu et al. (2022) proposes a novel architecture that employs large-kernel depth-wise convolutional basic blocks, which are designed to optimize the computational efficiency of the model and the ability to capture global context. Additionally, it incorporates CSP blocks, which are similar to those utilized in the YOLO series. This integration enables the inheritance of the exceptional accuracy performance observed in the YOLO series. This architecture is named *RTMDet*, which stands for *Real-Time Models for Object Detection*.

3.2.2 Semantic Segmentation Algorithms

Semantic segmentation algorithms have undergone significant evolution since the advent of deep learning. Prior to the deep learning era, the algorithms required manual design to recognize local appearance and consistency in images. The representative algorithm of this era was conditional random fields. However, the aforementioned algorithms lack flexibility, which results in difficulty in generalization and requires a significant amount of computational power (Csurka et al., 2022).

In the context of the deep learning era, several new genres of algorithms have emerged. These include the implementation of CNN and the more recent introduction of transformer architectures, which have become popular in recent times. The performance of these algorithms in terms of accuracy and generalization has evolved significantly.

3.2.2.1 Fully Convolutional Network

The proposed architecture of the Fully Convolutional Network (FCN) employs a minimal number of convolutional layers and fully connected layers as shown in Figure 3.9. The network exhibits a straightforward structure, with an end-to-end design that is capable of accommodating an arbitrary input size. The prediction process occurs concurrently with the convolution operations, and the resulting predictions are subsequently upsampled to match the input size. These predictions are then utilized for augmenting the final prediction of the FCN. (Long et al., 2015). While subsequent algorithms have since surpassed it in terms of performance, the fundamental FCN structure of extracting and utilizing feature maps at different depths has been adopted by other later proposed complex networks (Csurka et al., 2022).

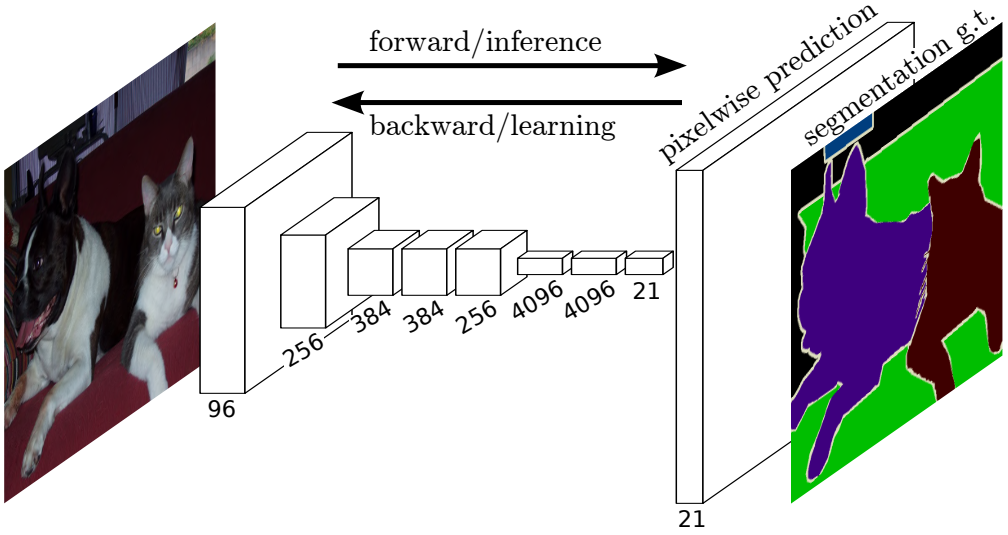


Figure 3.9: Network Architecture of FCN (Long et al., 2015)

3.2.2.2 Encoder-Decoder Architecture

While FCN encodes the input image into a more concise representation with feature maps generated by multiple convolutional layers, the inverse operation, deconvolution, is proposed to decompress the concise feature maps to resemble the input data. Thus, a novel network architecture is developed to convert the input image to the corresponding segmentation mask via intermediate concise representations. This network is called the Encoder-Decoder architecture (Csurka et al., 2022).

The representative implementations of this method include SegNet and UNet. SegNet is one of the simplest networks to utilize the encoder-decoder architecture. Its encoder comprises multiple convolutional layers and max pooling layers, which generate low-dimensional but deep representations of the features. The decoder has the same hierarchy as the encoder, but is inversely constructed. Its convolutional layers (trainable decoder filter bank) and max unpooling layers resume the feature maps to the original height and width of the input,

thus producing the segmentation mask with a softmax layer (Badrinarayanan et al., 2017).

The encoder-decoder architecture helps SegNet to generate segmentation masks with a relatively simple network, and having good training and inferring speed, benefitting from its simplicity. However, the details extracted in the intermediate layers of the encoder are partially lost, which negatively impacts the performance of the algorithm. UNet introduces a novel modified architecture that preserves the detailed information generated on the path of the encoder. In the decoder, the corresponding preserved feature maps are combined with the upsampled output to enhance the learning process (Ronneberger et al., 2015). The UNet does incur a slight increase in computational cost, but this is more than compensated by the significant improvement in model performance, particularly when only limited data is available for training.

3.2.2.3 Transformer in Semantic Segmentation

Since the introduction of the transformer architecture by Vaswani et al. (2017), which employs a self-attention mechanism, this architecture has become a focal point of research in the field of machine learning. The concept of self-attention places emphasis on the connections between tokens within a single input data. This addresses the challenge of losing correlation between distant tokens, a persistent issue in conventional CNNs.

The transformer architecture's ability to memorize long-distance correlations and its suitability for processing tokenized data make it an ideal candidate for natural language processing tasks. This novel architecture has become the de facto standard for natural language processing, facilitating the development of mature large language models, such as GPT.

The transformer's fruitful achievements have also attracted researchers from other fields to attempt to implement this method on other tasks. The initial and most straightforward implementation of the transformer in the semantic segmentation task is SETR, proposed by Zheng et al. (2021). This network eliminates the convolutional layers in the decoder section. Instead, it applies the transformer directly to the decoder. The image is divided into patches, which emulates the tokenization of Natural Language Processing (NLP) tasks, making it compatible with the transformer. This model rapidly achieved state-of-the-art performance in the ADE20K dataset. Nevertheless, this implementation is deficient in its lack of refinement, resulting in a significant increase in computational cost. This, in turn, necessitates the use of larger computational resources and a larger dataset (Thisanke et al., 2023).

A more sophisticated implementation was proposed by Cheng et al. (2022), called Mask2Former (Masked-attention Mask Transformer). Their scheme employs the transformer in the decoder part, while limiting the cross-attention to the predicted foreground area, i.e., masked-attention. This approach also allows for greater flexibility in the choice of encoder. The encoder can be implemented using either conventional CNNs or transformer structures for feature extraction. In comparison to the transformer encoder, the CNN

encoder is less computationally demanding. This model also unifies instance segmentation, semantic segmentation, and panoptic segmentation tasks, while achieving state-of-the-art performance in all these tasks (Thisanke et al., 2023).

Segment Anything Model (SAM) proposed by Kirillov et al. (2023) is a sophisticated segmentation model that leverages prompts, such as bounding boxes, to extract precise segmentation masks for given objects. Distinguished by its ability to adapt to various input prompts, SAM demonstrates remarkable versatility and accuracy in segmentation tasks. This model's approach contrasts notably with other transformer-based models. While Mask2Former integrates a transformer-based encoder-decoder structure to directly predict masks, SAM emphasizes a prompt-based interaction mechanism, allowing for more dynamic and user-driven segmentation processes. This distinction underscores the unique capability of SAM in handling diverse and interactive segmentation requirements, even data from the field, on which the model is not fully trained. This feature sets it apart from other transformer-based models in semantic segmentation.

3.3 Semi-supervised Learning: Pseudo Labeling

The application of supervised learning to scene understanding tasks typically requires a substantial quantity of well-annotated data, as previously stated in chapter 2. However, the acquisition of such data is challenging due to the high demand for human labor. For instance, the data labeling procedure of the MS COCO dataset is divided into multiple stages to create label information from image-level to pixel-level. The most labor-intensive stage of this process is instance segmentation. Statistical analysis revealed that the labeling process produced 2.5 million segments, with each segment requiring 22 worker hours. Moreover, these are only coarse instance outlines, and need further refinement, necessitating additional time and effort (Lin et al., 2014).

The lack of specificity in datasets designed for generic contexts presents a further challenge. Neural networks trained on these datasets may not perform optimally when applied to scenarios within the AEC industry. However, a well-annotated dataset for scene understanding tasks in the AEC industry is relatively costly and scarce. Therefore, the technique of leveraging unlabeled data with pseudo-labeling in semi-supervised learning can be introduced in order to circumvent the negative impact of scarce labeled data.

In the early stages of research on applying deep learning to computer vision tasks, researchers such as D.-H. Lee (2013) have already proposed the method of using pseudo labeling. In their research, the unlabeled handwritten numbers from the MNIST dataset (L. Deng, 2012) demonstrated a significant reduction in classification error for numbers, with a 26% improvement observed when only 100 labeled data were utilized. Yan et al. (2019) also developed a method for object detection by generating pseudo labels for unlabeled video frames. This approach achieved state-of-the-art performance on the VOS, DAVIS, and FBMS datasets at the time of publication. In the PseudoSeg framework proposed by Zou et al. (2020), a novel algorithm is employed for the segmentation task. Based on

the consistency constraint commonly used in the field of semi-supervised learning, the refined pseudo label is obtained by fusing the decoder prediction of weakly augmented data with the self-attention Grad-CAM (SGC). In the experiment with the COCO and VOC12 datasets, the method proposed in PseudoSeg demonstrated a significant improvement in mIoU, reaching up to 10% when only 1/256 of the training data had pixel-level labels, compared to fully supervised learning with the DeepLabv3+ network.

The pseudo-labeling method exhibited remarkable effectiveness in addressing the paucity of labeled data. For the AEC industry, where visual data is abundant but corresponding semantic data is comparatively scarce, this approach is especially advantageous.

Chapter 4

Methodology

4.1 Overview

As mentioned in chapter 1, this thesis research will focus on developing new datasets and methods. Both directions are crucial for developing a robust and precise workflow for automatic on-site data acquisition.

For the new dataset, it would be redundant to capture more raw data from construction projects, given that the crux of the dataset scarcity specific to the AEC industry lies in the shortage of comprehensive labels rather than raw data. Therefore, the creation of the new dataset aims to generate high-quality annotations for existing datasets. Considering that the task of consecutive workflow is scene understanding, the annotation to be created is the segmentation masks of the images in the original dataset. To best reflect the genuine scenarios of ongoing construction sites, the RGB images from the second sequence of ConSLAM have been selected as the base dataset. First, the similarity of the images in the second sequence is computed and compared, and the relatively unique images are selected for further processing. Second, an appropriate platform is constructed to efficiently create annotations for these unique images. Third, construction-related objects frequently appearing in the images are categorized into multiple classes. Next, refined ground truth annotations are created for this relatively small dataset and exported in COCO format. It is noteworthy that this small dataset will be used in the other direction of the exploration, in which the remaining unlabeled data will be leveraged to improve the model.

The arduous nature of data labeling renders the creation of comprehensive annotations for datasets such as ConSLAM impractical, given the imbalance between the efficiency of acquiring raw data and the manual labeling process. Consequently, a novel workflow based on semi-supervised learning is proposed. This workflow comprises three distinct networks, with two of these networks functioning in a consecutive manner as *teacher* models, responsible for generating pseudo labels for unlabeled data. The third semantic segmentation model serves as the *student* model, trained on the extended dataset with pseudo labels. This model is used to benchmark the enhancement brought by this workflow.

In the second proposed workflow, one of the *teacher* models is substituted to further alleviate the manual intervention. The teacher models no longer require custom retraining, thus forming a zero-shot workflow.

Accordingly, the experiments conducted in this research consist of three parts, as outlined below.

1. **New Data:** A new dataset based on ConSLAM with manually annotated segments and construction-related object labels

2. **New Workflow *A*:** Generating pseudo labels with semi-supervised learning approach

3. **New Workflow *B*:** Generating pseudo labels with zero-shot approach

4.2 New Data: Segmentation Masks based on ConSLAM

4.2.1 Dataset Preprocessing

The ConSLAM dataset selected for this task is described in chapter 3. The second sequence of ConSLAM contains 4,168 RGB images, which is an unreasonable amount of data for a single person to annotate within a limited time. Moreover, the neighboring frames of the images are similar, as this dataset is created continuously with a hand-held device, much like capturing a video. To alleviate the burden of manual labeling, the similarity of the images is compared by computing the hash values of the images, and the identified redundant images are categorized as unlabeled data, which would be used in the later process.

The RGB images from ConSLAM have a high resolution of 2064×1544. Assessing the similarities of these images pixel by pixel is inefficient due to the heavy computation involved and the susceptibility to slight interferences. A more reliable method is described below:

1. Reduce the size of the input image to 32×32 and convert the RGB image to greyscale to filter out high frequencies and unnecessary details.

2. Perform Discrete Cosine Transform (DCT) to convert the 32×32 greyscale image from the spatial domain to the frequency domain, similar to the Fourier transform.

3. Retain only the 8×8 elements in the top left of the DCT matrix to further reduce the high-frequency components.

4. Set all elements in the reduced DCT to either 0 or 1 based on whether their values are smaller or larger than the mean value, thereby constructing a 64-bit binary integer.

After this processing, the image is irreversibly compressed into a 64-bit binary integer called perceptual hash (Krawetz, 2011). The output fingerprints are visualized in Figure 4.1 (DCT is not reduced for better visualization). These fingerprints of the images are then compared bit by bit. A certain threshold of error is set, and a subset of images that have larger differences in terms of the perceptual hash than the given threshold is created. The consecutive labeling work will be based on this subset.

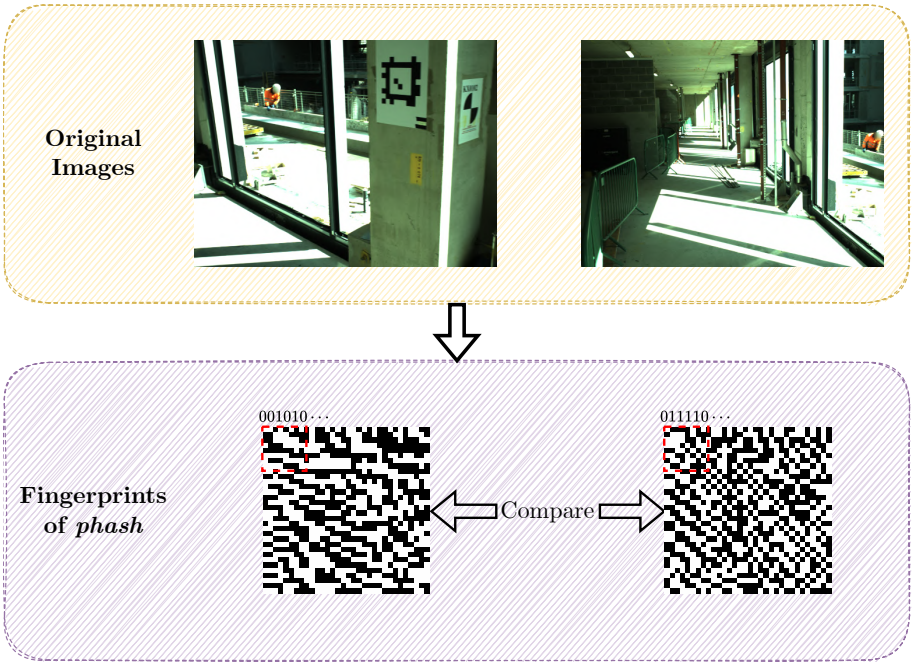


Figure 4.1: Fingerprints of the Perceptual Hash of Images

4.2.2 Annotation Tool: CVAT

The selection of annotation tools may not seem critical to the overall workflow, but if suitable toolkits for generating ground truth are not carefully chosen, the time and effort required for this tedious task can increase exponentially. Both academia and industry have a growing demand for efficient data annotation tools as algorithms continue to evolve. Initial solutions were primitive, typically limited to manual annotation and lacked any assisting mechanisms to facilitate the job. LabelMe and VGG Image Annotator are two typical examples of such tools. These are web-based, open-source annotation tools that can only be used for generating polygon annotations and pixel masks (Dutta & Zisserman, 2019; Russell et al., 2008). Although they are easy to deploy and handle, their lack of advanced functions significantly slows down the entire workflow.

To address these limitations, more advanced tools like RectLabel, PixelAnnotationTool, and COCO Annotator have been developed with embedded auxiliary functions to expedite the annotation task. However, these tools are either proprietary software or use outdated algorithms, such as the watershed algorithm, which is based on the topological characteristics of the image (Bréhéret, 2017; Brooks, 2019; Kawamura, 2017). Subsequently, several commercial companies have launched comprehensive solutions like LabelBox and Supervisely for the creation of large-scale datasets. These toolkits include almost every function necessary for fast and precise annotation, but they also contain modules not needed for individual research, making them cumbersome (Labelbox, 2024; Supervisely, 2023). Additionally, proprietary software is not preferable in the research workflows.

Ultimately, the Computer Vision Annotation Tool (CVAT), developed by CVAT.ai Corporation (2024), becomes the optimal choice for this research. CVAT is an open-source

annotation tool that strikes a balance between power and agility. It utilizes the cutting-edge SAM developed by Kirillov et al. (2023), which offers excellent performance on zero-shot segmentation. In the data preparation stage, SAM is used as an embedded component of CVAT. By providing positive and negative points as prompts, SAM could easily segment the given region in the images as demonstrated in Figure 4.2. Although the images captured on construction sites are not perfectly covered by its training set, SAM can still achieve satisfactory results. Due to the manual supervision and intervention during the annotation process, the ground-truth-level quality of the segmentation is assured.

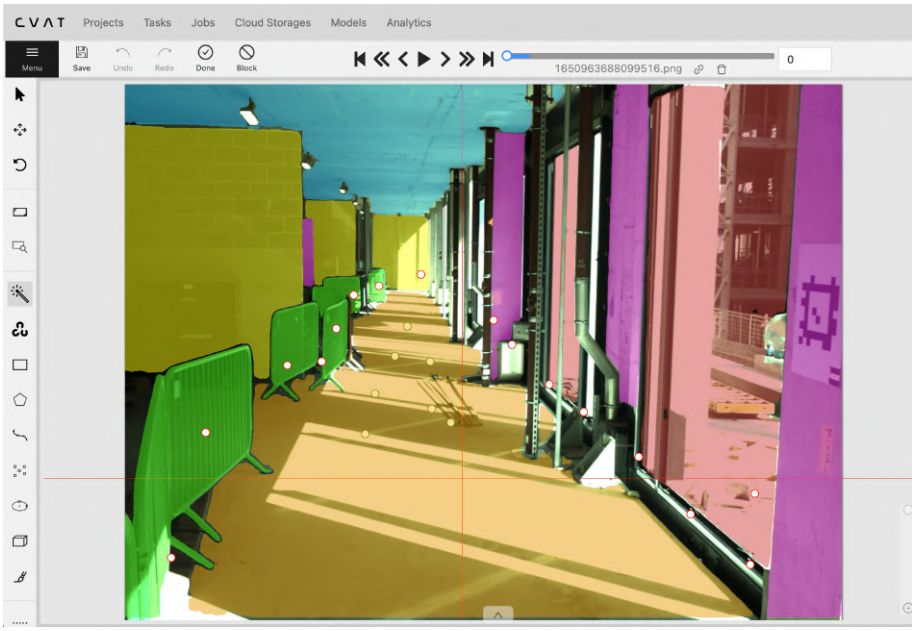


Figure 4.2: Prompting with Positive and Negative Points

4.2.3 Data Labeling

A subset of images were selected based on their similarity. These images were then imported into annotation tools for further processing. It is noteworthy that the annotations were made at the instance level, which means that the instance information of the objects was retained while generating semantic information. For instance, multiple columns exist within a single image. Despite the fact that different segments of columns are assigned the same label, the information for distinguishing different columns is preserved. This approach facilitates the implementation of the instance-level segmentation model. Furthermore, the segmentation can be directly converted to bounding boxes, which is also advantageous for the implementation of the semi-supervised workflow presented in the thesis.

4.2.4 Export, Convert, and Upload

Once the labeling work has been completed, the dataset is exported from the labeling platform. The format of annotation data used for this thesis is COCO, as proposed by the Lin et al. (2014). This format stores the segmentations with RLE and corresponding

bounding boxes in a compact, human-readable JSON text file. In order to enable the usage of this dataset for other researchers who may benefit from it, the labeled dataset is uploaded to Huggingface[1], a widely used platform for sharing machine learning-related resources. Given that the most prevalent format for segmentation data on this platform is image mask, the dataset is subsequently converted into this format.

4.3 Workflow *A*: Semi-supervised RTMDet-SAM

4.3.1 General Idea

With the dataset containing ground truth annotations created, the next step is to utilize the unlabeled data in the second sequence of ConSLAM. The workflow *A* of semi-supervised learning by pseudo labeling presented in this thesis is illustrated in Figure 4.3.

First, a lightweight object detection model is trained based on the small dataset with ground truth. This model converges with less data requirement and is able to generate reliable bounding boxes as prompts for subsequent segmentation.

In the next step, a transformer-based model that takes prompt input to generate accurate segmentation is applied. The raw image and the bounding boxes of the objects within serve as the input for this model. Notably, this model does not necessitate retraining of the pretrained model, i.e., it is capable of zero-shot detection. After this step, the output segmentation data from the model forms pseudo labels for the previously unlabeled data.

Finally, the data with ground truth and pseudo labels are jointly utilized to train another end-to-end semantic segmentation model, analogous to a *student* model. This model can be more complex than the previously trained lightweight model, containing more parameters and a more sophisticated network architecture. The extended dataset using pseudo labels enables the training of such a model, which could underfit when trained on limited data.

[1] https://huggingface.co/datasets/erwinqi/conslam_seq2_segmentation_gt

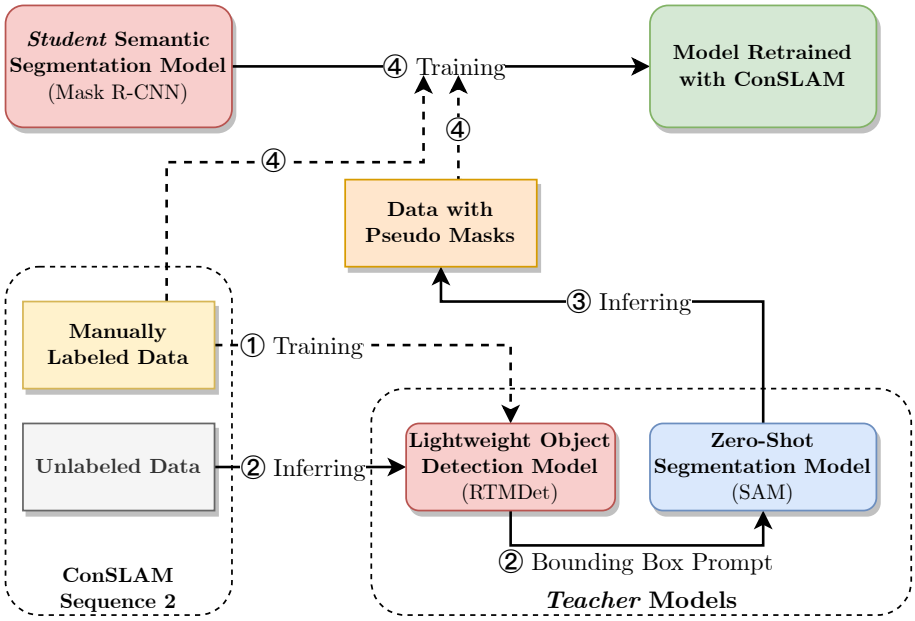


Figure 4.3: Workflow *A*: Semi-supervised learning Approach with RTMDet-SAM

To evaluate the degree of performance optimization, this segmentation model is also trained on bare manually labeled data as illustrated in Figure 4.4 and serves as a comparison.

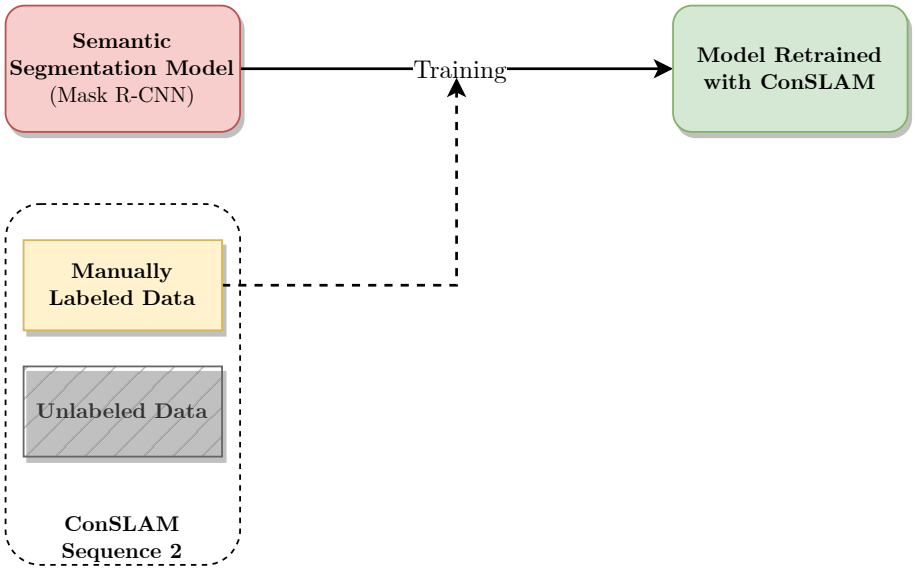


Figure 4.4: Workflow of Fully Supervised Approach in Comparison

4.3.2 Training Object Detection Model for Prompt Generation

As previously stated, the initial step in utilizing unlabelled data is to generate high-quality bounding boxes. However, due to the limited quantity of labeled data and the hardware available for training, a lightweight yet efficient algorithm for object detection is necessary. The RTMDet proposed by Lyu et al. (2022) provides a feasible solution based on the YOLO algorithms. This tiny version of the model contains only 4.8M parameters, yet it is still capable of achieving satisfactory performance even on large datasets.

To maintain the feature extraction capabilities of the model that was pretrained on a large dataset, the parameters of the backbone are initialized with weights that were pretrained on the ImageNet dataset proposed by J. Deng et al. (2009). This enables the training of a model with good performance even with limited data.

The image data is preprocessed with standardization and normalization before training. The mean and standard deviation of each channel are calculated over the entire dataset. Next, the values of each channel and each image are standardized as shown in Equation 4.1. Then, the values are normalized by scaling down from [0,255] to [0,1] as demonstrated in Equation 4.2. The remaining preprocessing methods, which originate from the RTMDet architecture, are retained in their original form.

$$\begin{aligned} \text{mean}_c &= \frac{1}{N}\sum_{i=1}^{N} x_{i,c} \\ \text{std}_c &= \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{i,c} - \text{mean}_c)^2} \\ \text{standardized\_image}_c &= \frac{\text{image}_c - \text{mean}_c}{\text{std}_c} \quad \text{for } c \in \{R, G, B\} \end{aligned} \tag{4.1}$$

$$\text{normalized\_image}_c = \frac{\text{standardized\_image}_c}{255.0} \quad \text{for } c \in \{R, G, B\} \tag{4.2}$$

In the training process, the Adam optimizer, proposed by Kingma and Ba (2014), is applied. This optimizer allows an adaptive learning rate for each parameter individually and uses the first and second moments of the gradients to find the optimal direction for the iterative optimization. Weight decay is a regularization technique used to prevent overfitting by adding a penalty on the size of the weights, which improves the generalization of the model.

4.3.3 Creating Pseudo Labels for Unlabeled Data

4.3.3.1 Bounding Box Prompts

Once the object detection model has converged, the unlabeled data can be pipelined into the newly retrained model to generate bounding boxes for subsequent processing. The quality of the bounding boxes generated in this stage is of paramount importance, as they will be used as prompts for zero-shot segmentation. The NMS plays a pivotal role here by removing redundant bounding boxes and preserving those with the highest confidence scores.

4.3.3.2 From Prompts to Segments

As a zero-shot segmentation model, SAM does not necessitate retraining. The pretrained weights using the SA-1B dataset are directly used in this process. This dataset, published with SAM, contains 11M images and over 1B masks (Kirillov et al., 2023). The pretrained SAM is available in different versions according to the scale of parameters in the Vision Transformer encoder. As no further training is required in this thesis, the huge size version of the encoder, which contains 636M parameters, is used. This deep network is fully capable of exhausting the power of the vast dataset, resulting in optimal performance in terms of generalization. While this choice results in a relatively slow inferring speed, it avoids the need for manual intervention and ensures the quality of the generated pseudo labels.

The segmentation results are presented in the form of color masks. In order to ensure coherence between the pseudo labels and the manually labeled ground truth, the color masks are converted to the COCO format.

4.3.4 Training Semantic Segmentation Model

Although SAM exhibits excellent performance in segmentation tasks, even handling scenes of construction sites, which are not typical in datasets for generic purposes, the automatic on-site data acquisition necessitates a real-time model that can process images with high speed. The semi-supervised workflow of the pseudo-label generation compromises inferring speed in exchange for the ability to utilize unlabeled data. Therefore, a new model for semantic segmentation is needed. This model strikes a balance between the complexity of the network and the inferring speed. The aforementioned workflow with RTMDet and SAM serves as the *teacher*, while the new model is the *student*. The ability to segment construction-related objects is transferred to the *student* model through training with pseudo labels generated by the *teacher*.

The Mask R-CNN model is selected as the *student* model. This two-stage network is capable of performing instance-level semantic segmentation while maintaining a reasonable processing speed. The pretrained backbone is retained and utilized for enhanced feature extraction. The backbone is ResNeXT101, a proposed variant of ResNet by Xie, which exhibits optimized performance in terms of both accuracy and computational efficiency. The Stochastic Gradient Descent (SGD) is employed as the optimizer for the network.Additionally, the image data is standardized and normalized in accordance with Equations 4.1 and 4.2.

To assess the performance of the *student* model relative to that of the model trained without the use of unlabeled data, the network is retrained with the same setup but only manually labeled data. The results of this comparison are presented in chapter 6.

4.4 Workflow *B*: Zero-shot Approach

Inspired by the zero-shot performance of the transformer-based segmentation model, another bold attempt is implemented in this thesis. The lightweight object detection model used for generating prompts is substituted with an Open-Vocabulary Object Detection (OVD) model. The workflow is illustrated in Figure 4.5. This model leverages the transformer architecture and uses semantic information from text prompts to detect objects in the images. This method goes even further and has the potential to create a completely zero-shot workflow.

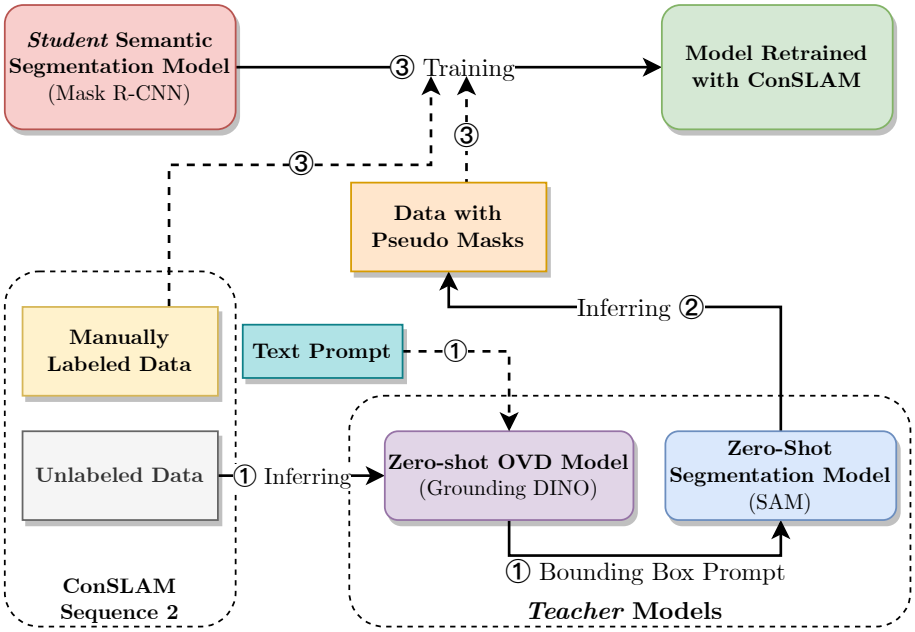


Figure 4.5: Workflow of Completely Zero-shot Approach

Given that pseudo labels are generated using bounding box prompts, it is worth considering further methods to automate the entire labeling process. The prosperity of NLP algorithms brought by the introduction of transformer architecture has also enabled the use of open-vocabulary object detection algorithms, which can absorb semantic information from text labels and construct mappings between text labels and bounding boxes in image data. This workflow has the potential to create an AEC-specific dataset by leveraging other datasets, obviating the necessity for manual intervention.

The Grounding DINO proposed by S. Liu et al. (2023) is selected as the OVD model in this thesis. The model is based on another transformer-based end-to-end object detection model, called DINO (Zhang et al., 2022). The Grounding DINO uses a vision-language modality fusion mechanism and allows arbitrary text to be used as queries to detect objects matching the label. The architecture of Grounding DINO is illustrated in Figure 4.6.

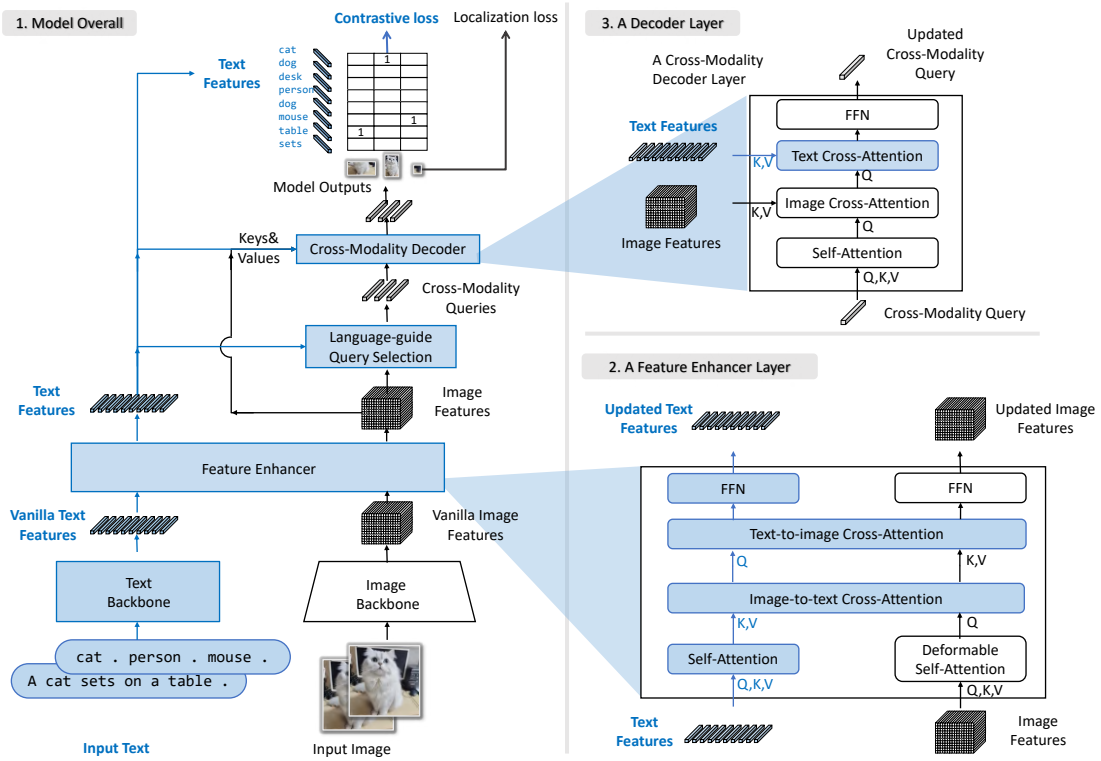


Figure 4.6: Architecture of Grounding DINO (S. Liu et al., 2023)

Compared to regular object detection models, the OVD model has greater potential for zero-shot detection. However, this capability comes with a drawback: the model relies solely on text prompts to recognize objects. The richness of semantic information in these text prompts determines the effectiveness of detection. In regular object detection, label text serves only to make the results human-readable, so it can be concise and somewhat ambiguous. However, when experimenting with OVD, the label text must be descriptive and comprehensively cover the features of the objects. This is the most challenging aspect of the experiment.

Chapter 5

Experiments

5.1 Hardware & Software Used in Experiment

5.1.1 Hardware

The self-deployed image annotation tool, the model training, and the inferring are completed on a 64-bit Debian 11-base Windows Subsystem Linux (WSL). The configuration is listed in Table 5.1.

Components	Specification
CPU	Intel i5-6600 (4) @ 3.311GHz
RAM	16GB
GPU	NVIDIA GeForce GTX 1080 Ti
VRAM	11GB

Table 5.1: Configuration of Computer Used in the Research

5.1.2 Software

The research conducted for this thesis involved the utilization of numerous open-source software and libraries, which are detailed below.

- Python 3.10.13 is employed for coding script for data preprocessing and postprocessing, as well as for model training and inferring in the thesis. Higher versions could encounter compatibility issues with some algorithms in MMDetection.
- In this thesis, the self-host CVAT[1] platform version 2.8.2 is employed for the purpose of data annotation, which is equipped with the capacity to annotate segment regions with prompts.
- MMDetection[2] is a deep learning framework built upon PyTorch, providing an easy-to-use interface, where users can modify the detail of the algorithms. The research in this thesis is conducted on MMDetection version 3.3.0
- The thesis employs PyTorch[3] version 2.2.1, as a low-level library utilized for model training and inferring.

[1] https://github.com/cvat-ai/cvat
[2] https://github.com/open-mmlab/mmdetection
[3] https://github.com/pytorch/pytorch

- The CUDA Toolkit[4] version 12.4 is used on Debian 11 installed on WSL2

5.2 Generating Ground Truth Labels

5.2.1 Deploying CVAT

CVAT is a computer vision data annotation platform that incorporates both a backend and a frontend, offering a range of advanced functions. Its deployment is more complex than that of simple tools such as LabelMe and VGG, as discussed in subsection 4.2.2. However, the use of Docker facilitates the deployment process, from the website to the SAM segmenting interactor, making it relatively straightforward.

It should be noted that the SAM included in the toolkits requires at least 8GB of Video Random Access Memory (VRAM) in order to function properly.

5.2.2 Creating Subset from ConSLAM

As described in chapter 4, the images to be labeled are selected by similarity to avoid introducing bias by neglecting scenes in the dataset. The perceptual hash of 4,168 images from Sequence 2 is computed and compared. The error-tolerance threshold determines the number of unique images judged by the algorithm. The experiment is repeated several times with different thresholds. The number of unique images identified each time is listed in Table 5.2. The final threshold chosen is 21, with 254 images identified as unique, which is a reasonable amount to balance the scarcity and labeling burden.

Threshold	1	5	10	15	20	21	23	25
Unique Amount	3,968	2,599	1,567	801	468	254	140	64

Table 5.2: Amount of Unique Images Determined by Different Thresholds

5.2.3 Initial Attempt with 19 Classes

5.2.3.1 Designing Classes

In the first attempt, after recording the construction-related objects appearing in the images, 19 different categories are identified, as listed in Table 5.3. Almost all objects in the scenes are listed.

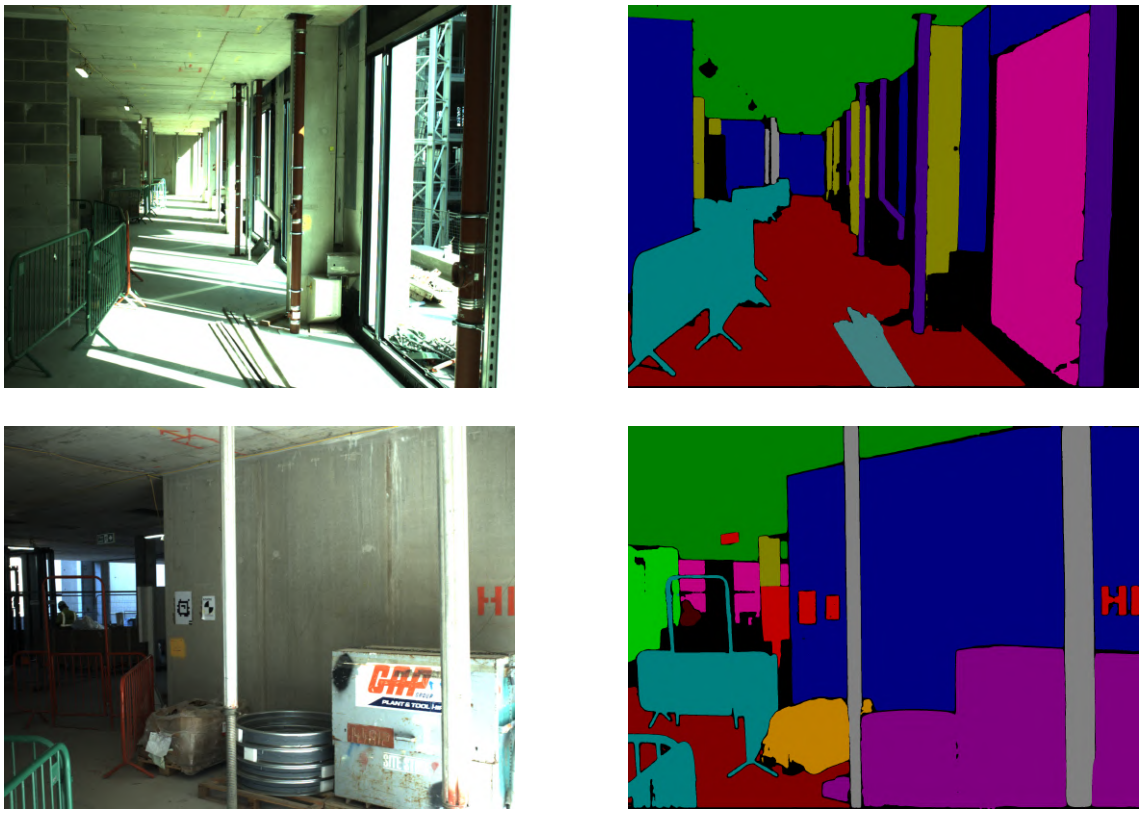
[4] https://developer.nvidia.com/cuda-downloads?target_os=Linux

Label Text	floor	ceiling	column	wall	equipment
	fence	temp shoring	personnel	sign	dumpster
	crate	pipe	opening	steel bar	concrete block
	bag	door	curtain wall frame	window	

Table 5.3: Labels of the 19 Classes

5.2.3.2 Annotating Images

The annotation of this attempt was aborted due to problems caused by an excessive number of classes. 99 images are annotated in this process, and a total number of 1,615 segments spread over 19 classes are created. Some examples of the annotations are shown in Figure 5.1.



(a) Original Image (b) Ground Truth Mask

Figure 5.1: Sample Annotations Created on CVAT with 19 Classes

The category distribution of the 1,615 segments created is shown in Figure 5.2. The number of segments created for steel bar, window, door, bag, concrete block, and dumpster are significantly less than for other objects. The imbalance between labels is a crucial reason that causes underfitting for minor labels and overfitting for major labels, which is detrimental to the generalization of the trained model.

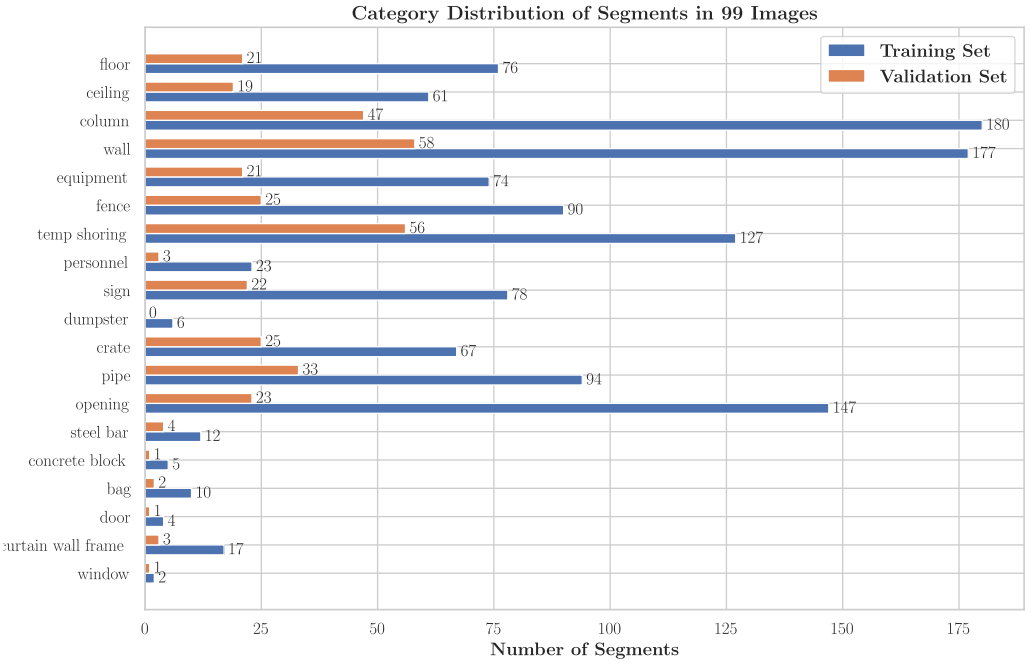


Figure 5.2: Category Distribution of Segments in 99 Images

The area distribution of the segments also shows the same problem. The outliers are observed in at least 7 categories, which means that the areas of the segments oscillate strongly, as shown in Figure 5.3. In order to avoid the negative influence of this when training the model, it is necessary to reduce the initially designed categories.

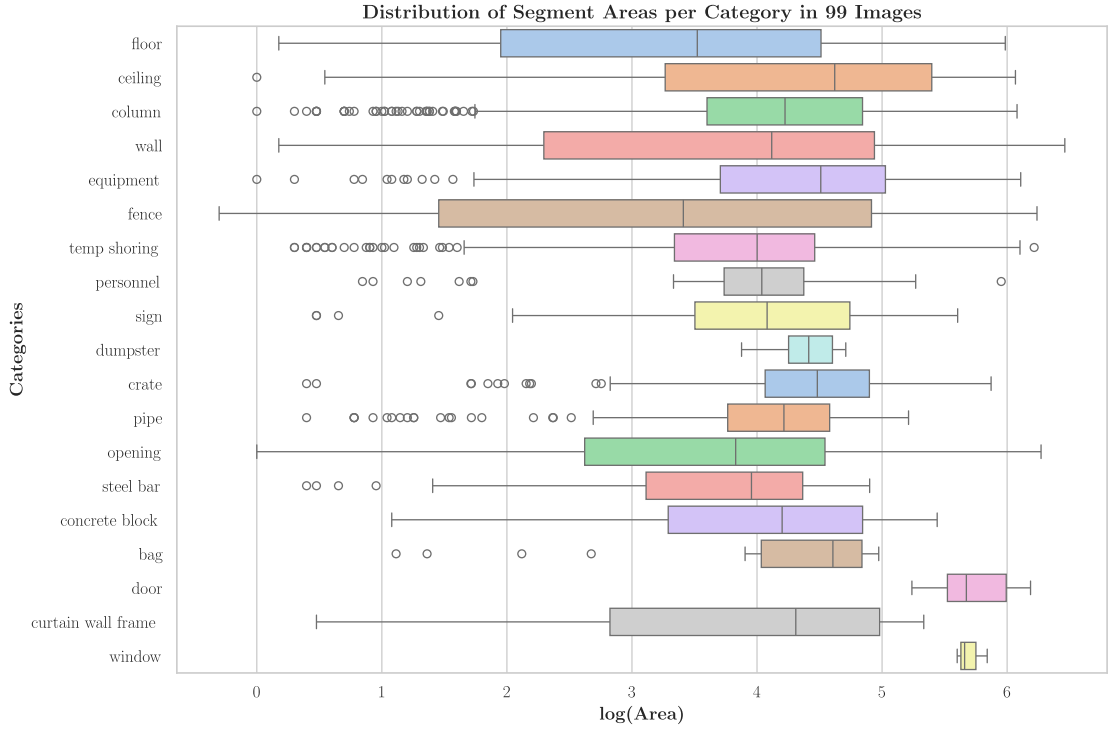


Figure 5.3: Distribution of Segment Areas per Category in 99 Images

5.2.4 Attempt with Reduced 8 Classes

5.2.4.1 Reducing Classes

After careful consideration, the minor categories were either merged with other categories or eliminated, resulting in 8 classes. The labels of these categories are listed in Table 5.4.

Label Text	ceiling	column	fence	floor	opening	personnel	wall	window

Table 5.4: Labels of the Reduced Classes

5.2.4.2 Annotating Images

The statistics of the dataset with annotations indicate that the problem of scarce categories causing unbalanced design is mitigated. The detailed result of this annotation process is presented in section 6.1.

5.3 Experiment with Workflow *A*

5.3.1 Training RTMDet Model

As shown in Figure 4.3, the RTMDet model used to generate bounding box prompts is trained on the ground truth data. The dataset is preprocessed before it is used in the workflow. The mean and standard deviation values used for standardization are calculated over the entire dataset of 4,168 images and are listed in Table 5.5.

	R	**G**	**B**
Mean	103.53	116.28	103.198
Standard Deviation	79.61	83.62	83.4

Table 5.5: Mean and Standard Deviation Values for Standardization

The hyperparameters are set to achieve both fast speed and optimal convergence. Their values are listed in Table 5.6.

Hyperparameter	`batch_size`	`max_epochs`	`lr`[5]	`weight_decay`
Value	32	300	0.004	0.05

Table 5.6: Hyperparameters Setting for RTMDet Training

To store the current state of training, a checkpoint is saved every 10 epochs. A checkpoint primarily contains the architecture of the model and the corresponding weights at the time of creation. Additionally, the gradients at the current epoch are also stored, enabling the

possibility to resume training from the current epoch using the checkpoint. During the training process, only the latest three checkpoints are retained. The final checkpoint can serve as the model weights used for generating prompts in subsequent processes.

5.3.2 Generating Bounding Box Prompts

The model weights saved after 300 epochs are used for generating prompts. This lightweight model requires only a portion of the computational power of the hardware, allowing the inference process to run in parallel. The confidence score threshold is set to 0.4, which suppresses low-quality results while preserving the maximal number of prompts.

The threshold of the IoU for judging redundancy is set to 0.65. This value filters the majority of the bounding boxes that may confuse the algorithm without deleting the bounding boxes of those crowded objects.

5.3.3 Generating Pseudo Labels

The SAM is capable of zero-shot segmentation, and therefore, retraining is not applied in this case. Instead, the unlabeled images, along with the bounding boxes generated in the previous step, are fed to the SAM to directly generate pseudo labels. However, there are still some considerations involved when applying the SAM. Due to the limitations of the hardware specification, the inference is not implemented in parallel. Besides, in order to ensure the optimal quality of the generated pseudo labels, the heavier version of the visual transformer, ViT-H, is selected as the backbone.

5.3.4 Training *Student* Model with Pseudo Labels

The Mask R-CNN model was trained twice with the same configuration but with different datasets. The mean and standard deviation values used for standardization were the same as those used in training RTMDet, as listed in Table 5.5. The hyperparameters that are vital for the model performance are listed below in Table 5.7.

Hyperparameter	`batch_size`	`max_epochs`	`lr`	`momentum`	`weight_decay`
Value	2	12	0.02	0.9	1×10^{-4}

Table 5.7: Hyperparameters Setting for Mask R-CNN Training

The analysis of the performance of the trained model is presented in chapter 6, where comparisons are made between the model trained solely with ground truth and the model trained with both ground truth and pseudo labels.

5.4 Experiment with Workflow *B*

5.4.1 Specification of Grounding DINO

As illustrated in Figure 4.5, this approach aims to remove the trainable component from the pseudo-labeling workflow, achieving a fully zero-shot workflow. This is accomplished by substituting the RTMDet with the transformer-based Grounding DINO, a model capable of inferring bounding boxes using text prompts about the objects. Similar to SAM, this model is computationally demanding. Therefore, the compact version using the SWIN-T backbone (Z. Liu et al., 2021) is applied in this thesis. This model allegedly achieved a mAP of 0.484 on the COCO dataset, even without training on it (S. Liu et al., 2023), which is promising for generating pseudo labels for AEC-specific visual data.

5.4.2 Hyperparameters: Two Thresholds

Despite the zero-shot workflow characteristic, the model still requires careful configuration of certain options. Two thresholds of confidence scores control two vital parts of the inference process with Grounding DINO, as listed in Table 5.8.

Hyperparameter	`TEXT_THRESHOLD`	`BOX_THRESHOLD`
Value Finally Chosen	0.25	0.28
Feasible Range	$0.25 \sim 0.35$	$0.25 \sim 0.35$

Table 5.8: Hyperparameters Setting for Workflow *B*

The first is the text threshold. As illustrated in Figure 4.6, text features are generated by the text backbone and feature enhancer based on the text prompts input. The output text features come with corresponding confidence scores. The text threshold suppresses the output with low confidence, which could filter a large number of misclassified objects. However, an excessively high threshold could also unexpectedly filter all outputs. As demonstrated in experiments, this threshold between 0.25 and 0.35 is suitable when conducting zero-shot detection.

The second is the box threshold. Similar to other object detection models, the bounding boxes are proposed by the model with an uncertainty reflected in the confidence score. This threshold is designed to prevent the erroneous bounding boxes that could confuse SAM during the segmentation process. Also, the optimal threshold value lies within the range of 0.25 to 0.35, as demonstrated in the experimental results.

5.4.3 Special Treatment with Label Text

In the experiment of this model, the object labels mentioned in Table 5.4 were extended as shown in Table 5.9 to fully describe the characteristics of the objects, as no training data was provided for the algorithm to accurately recognize the objects.

Label Text	Extended Label
ceiling	grey concrete ceiling
column	grey concrete column
fence	green red fence
floor	grey concrete floor
opening	opening outside
personnel	person helmet
wall	grey concrete wall
window	window curtain

Table 5.9: Extended Labels

5.5 Summary

In general, four distinct networks with varying architectural designs are utilized in the experiments. To facilitate a more comprehensive understanding of the workflows, the aforementioned models are summarized in Table 5.10

Models	RTMDet	SAM	Grounding DINO	Mask R-CNN
Training Involved	✓	×	×	✓
Inferring Involved	✓	✓	✓	✓
Type	Object Detection	Zero-Shot Segmentation	OVD	Semantic Segmentation
Workflow	A	A&B	B	A&B

Table 5.10: Summary of Applied Models

Chapter 6

Results & Analysis

6.1 Statistics of Manually Labeled Dataset

Once the labeling process is complete, a total of 2,096 segments are created in 254 images. While each image contains approximately 8 segments, the true number of segments in each image is uneven. This uneven distribution is taken into account when dividing the train and validation subsets. The 80% of the segments are marked as training data, while the remaining 20% is used as a validation set. The distribution of segments with different labels is plotted in Figure 6.1. Due to the distinctive characteristics of different objects, the area of these segments varies significantly. Figure 6.2 illustrates the distribution of varying areas of different classes. The data indicates that the areas of the segments for column objects vary considerably, with outliers distributed across a relatively narrow range. In contrast, the variation of segments for personnel is relatively small, although there are some outliers. The areas of the segments for other categories are quite coherent, which suggests that the data quality is good. Besides, Figure 6.3 demonstrates two samples of created segmentation masks.

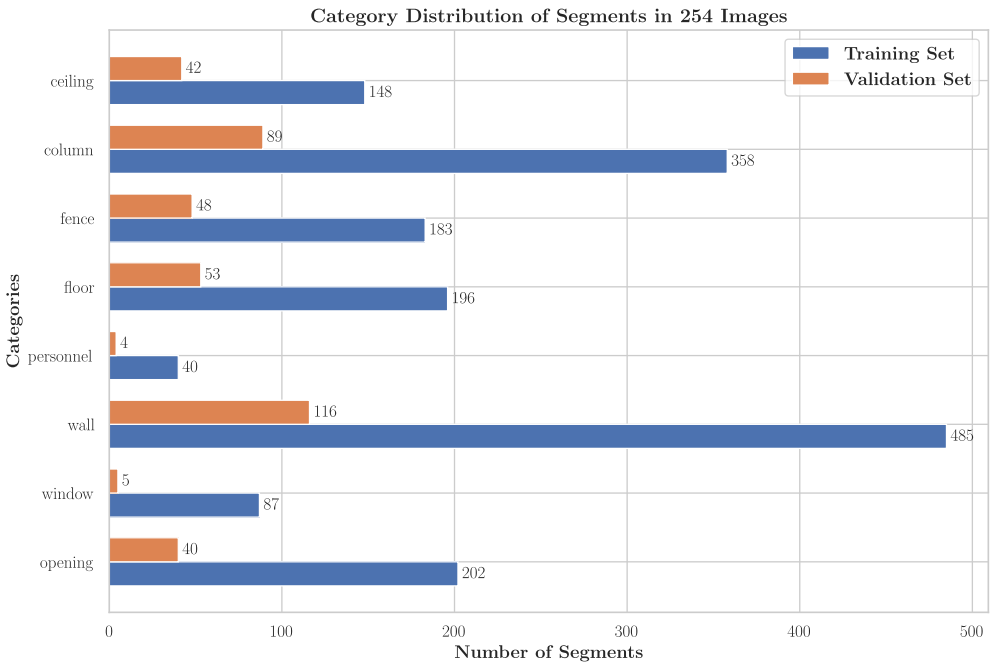


Figure 6.1: Category Distribution of Segments in 254 Images

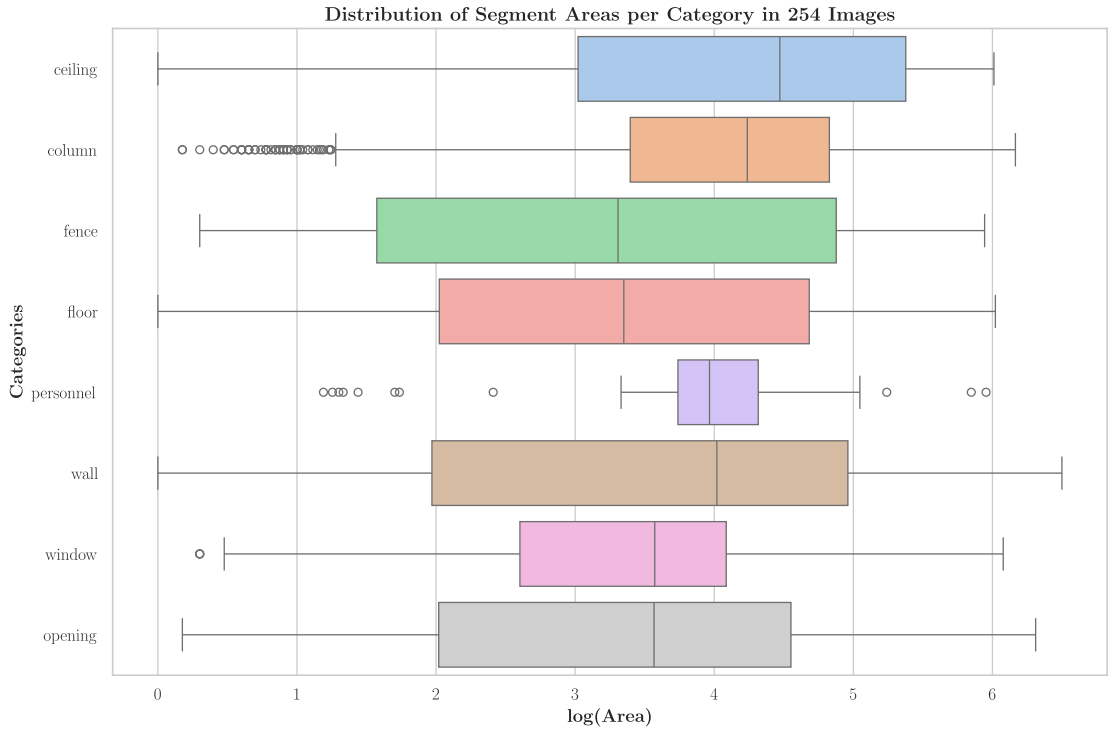
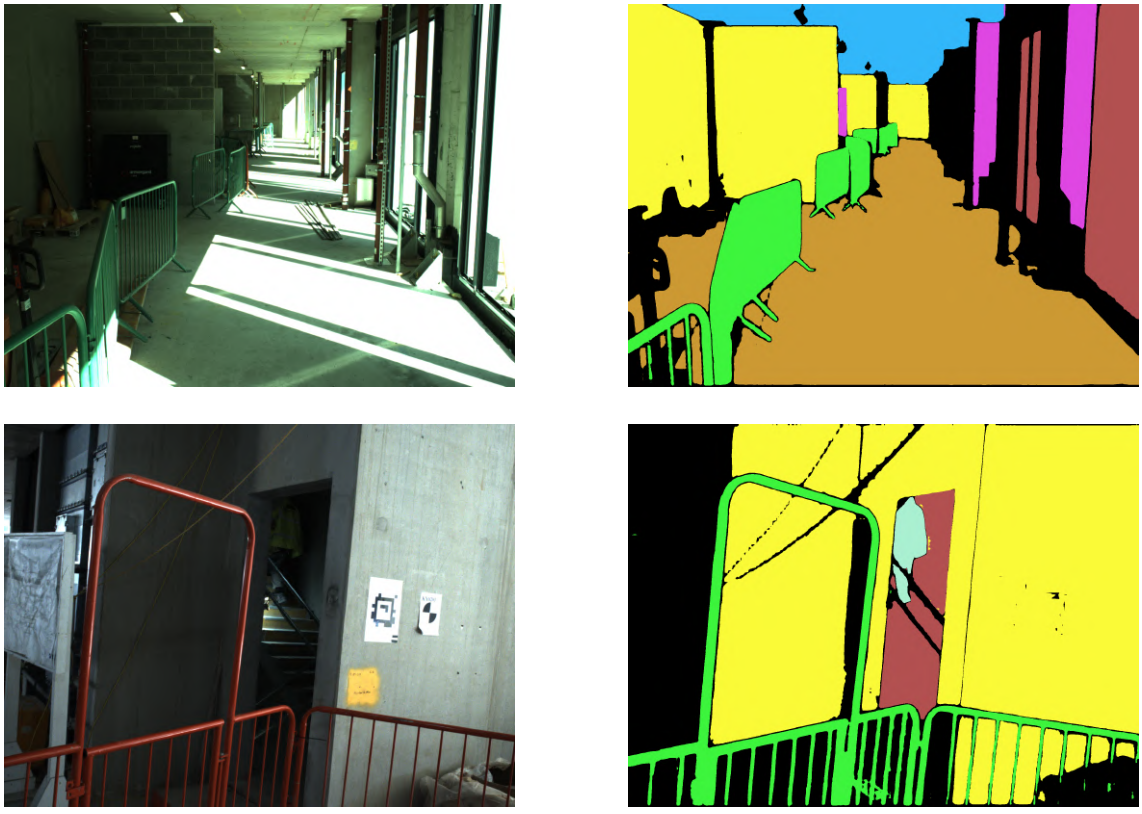


Figure 6.2: Distribution of Segment Areas per Category in 254 Images



(a) Original Image (b) Ground Truth Mask

Figure 6.3: Sample Annotations Created on CVAT with 8 Classes

This quantity of data may appear to be adequate for training a model with satisfactory performance. Nevertheless, the dataset remains suboptimal for a complex model that can achieve high accuracy in complex scenes, due to the risk of underfitting or overfitting. This is precisely the reason for augmenting the labeled dataset with additional information from the unlabeled dataset.

6.2 Performance of Semi-supervised Approach

6.2.1 Training RTMDet Network

Although only 6% of images in the entire dataset were manually labeled in the previous step, this provided 2,096 segments for model training. Despite this limited amount of labeled data, the RTMDet model still managed to converge. Due to the manageable scale of the dataset, it was possible to feed the entire dataset into the model at once in each epoch, as illustrated in Figure 6.4. The model converged after 300 epochs, taking a total of 3 hours, 14 minutes, and 6 seconds as listed in Table 6.2.

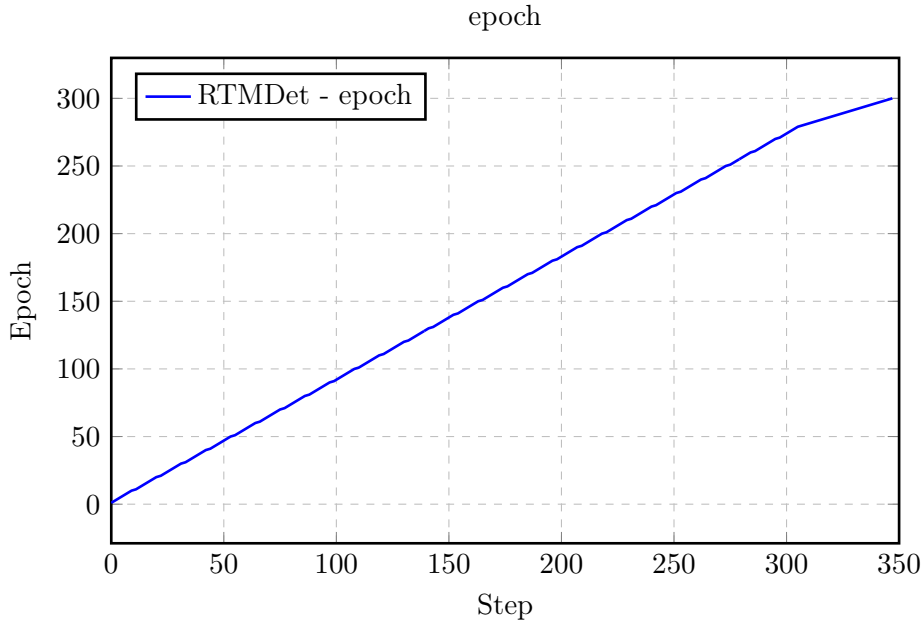


Figure 6.4: Epoch-Step Relation During RTMDet Training

6.2.1.1 Metrics of the Model

The loss function and the learning rate are the primary metrics for evaluating the training of the model. During the training process, the learning rate initially increases to the designated value and then drastically decreases as it approaches the optimal position. This behavior is also reflected in the loss function, which shows a slower rate of decrease as it nears the optimum. Figure 6.5 illustrates these two metrics.

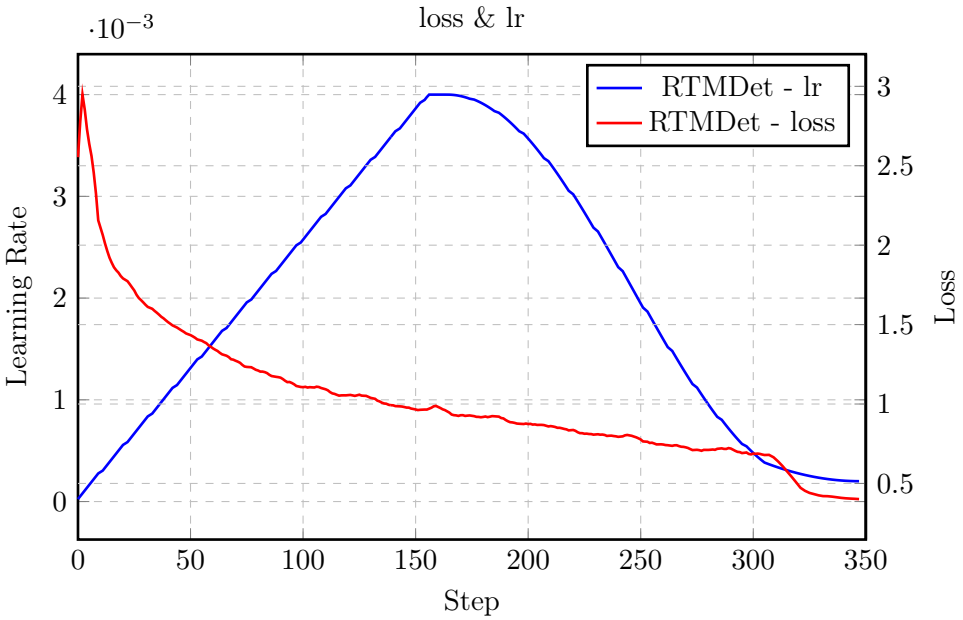


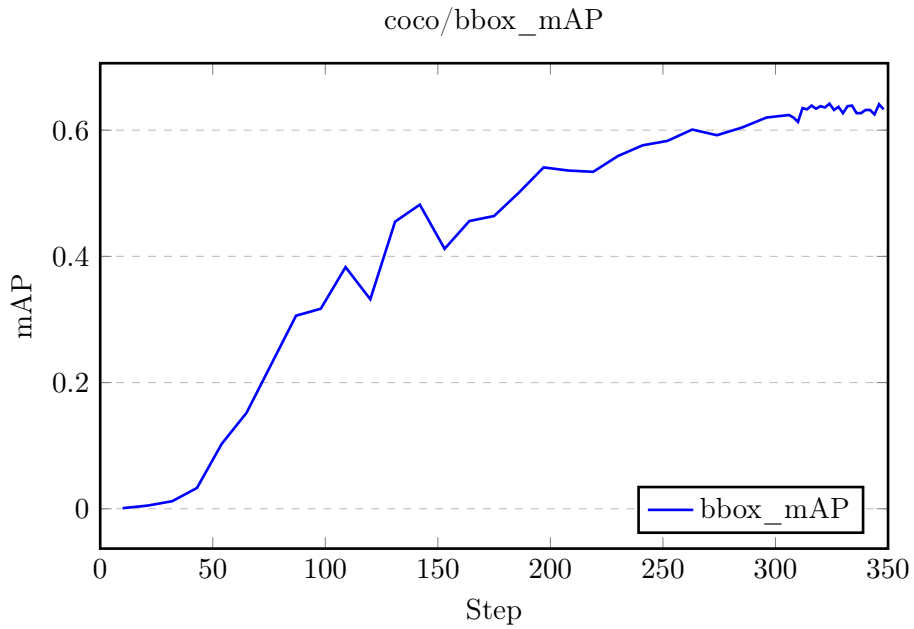
Figure 6.5: Learning Rate and Loss Function During RTMDet Training

The mAP and AR are two essential metrics for evaluating object detection models besides the loss function. These metrics reflect the precision and recall of the model, respectively, and are calculated during the evaluation stages along with the training process. During the first 280 epochs, evaluations occur at 10-epoch intervals, while in the last 20 epochs, performance is evaluated after every epoch.

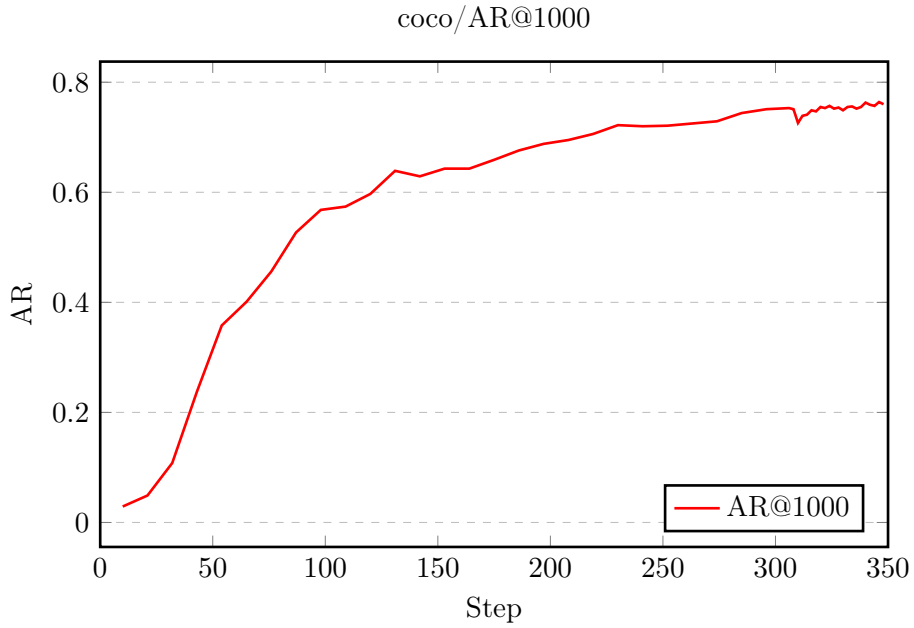
In this process, the metrics optimized by Lin et al. (2014) are utilized. To acquire a comprehensive mAP, the mAP values at IoU thresholds ranging from 0.5 to 0.95 at 0.05 intervals are averaged. This technique discourages overfitting caused by a single threshold and provides a more holistic view of the capability to accurately localize objects of different sizes and shapes. In the final evaluation, the mAP reaches 0.633 as illustrated in Figure 6.6a, which is a reasonable performance considering the scarcity of the training data.

The AR is calculated with a maximum of 1,000 bounding boxes per image and is also averaged over the aforementioned IoU thresholds. The final result reaches 0.76 as shown in Figure 6.6b, which is a preferable value indicating that the model is capable of comprehensively generating prompts for subsequent processes.

The values of the metrics calculated in the last evaluation stage are listed in Table 6.1



(a) mAP



(b) AR

Figure 6.6: mAP and AR During RTMDet Training

	Final Learning Rate	**Loss**	**mAP**	**AR**
RTMDet	2.001×10^{-4}	0.4016	0.633	0.76

Table 6.1: Metrics at the Final Evaluation after 300 Epochs

6.2.2 Bounding Box Prompts from RTMDet

The prompt generation for 4,168 images is completed in 42 minutes and 57 seconds, as summarized in Table 6.2. Figure 6.7 shows some samples of the results. Note that this is only a visualization of the prompts, while the prompts used in the next step are stored as coordinates and sizes of bounding boxes in each image.

Figure 6.7: Bounding Box Prompts Generated by RTMDet

6.2.3 Pseudo Labels from SAM

In the next step, pseudo labels in the form of masks are generated using SAM. Compared to RTMDet, SAM is much heavier and cannot be deployed in parallel on the current hardware. Consequently, the time required to infer 4,168 images is significantly higher, reaching 17 hours and 25 minutes, as listed in Table 6.2. The resulting pseudo masks are demonstrated in Figure 6.8, compared with ground truth. These results are then converted to COCO format and split into training and validation sets with a ratio of 4:1. The distribution of classes in the dataset with pseudo labels is illustrated in Figure 6.9, and the distribution of area of the segments is shown in Figure 6.10. A comparison of the distribution of areas of manually labeled data demonstrated in Figure 6.2 with that of the pseudo labels created for the entire dataset reveals a similar pattern. Notably, the outliers now cluster in the range of higher area, which may be attributed to the coexistence of two major clusters of area values in the column category. The other outliers are observed in the personnel category. The consistent pattern suggests a high degree of coherence with the ground truth data.

The dataset is later reorganized and uploaded to Huggingface[1] for the purpose of sharing with researchers who may be interested in it.

(a) Pseudo Label (b) Ground Truth

Figure 6.8: Comparison between Pseudo Labels and Ground Truth

[1] https://huggingface.co/datasets/erwinqi/conslam_seq2_segmentation_pseudo

Figure 6.9: Category Distribution of Segments in Whole Dataset

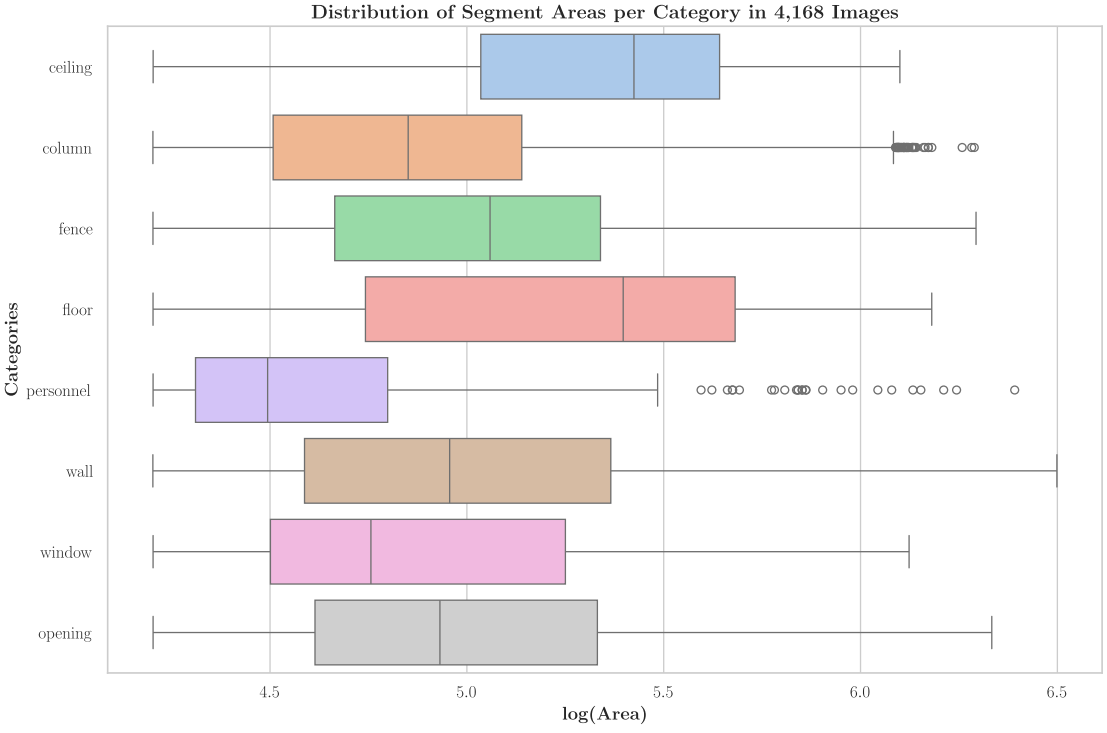


Figure 6.10: Log-Transformed Distribution of Segment Areas per Category in Whole Dataset

	Training Time	Inference Time
RTMDet	3h14m6s	42m57s
SAM	N/A	17h25m

Table 6.2: Running Time of *Teacher* Models in Workflow *A*

6.2.4 *Student* Semantic Segmentation Model

6.2.4.1 Model Trained with Pseudo Labels

As mentioned before, Mask R-CNN serves as the student model to be trained using the pseudo labels generated by the RTMDet-SAM workflow. The mAP and AR are similarly defined as the metrics used while training RTMDet. The training process took 7 hours and 4 minutes as listed in Table 6.3. The learning rate during the training is illustrated in Figure 6.11, and the overall loss, and the RPN loss, which reflects the recall performance of the model, are illustrated in Figure 6.13. The segment mAP and AR is illustrated in Figures 6.12a and 6.12b.

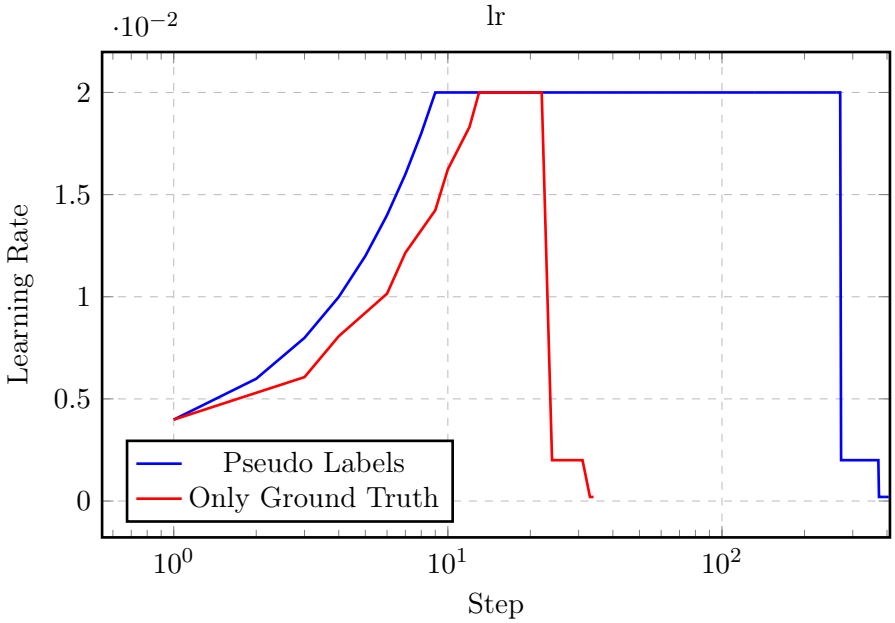
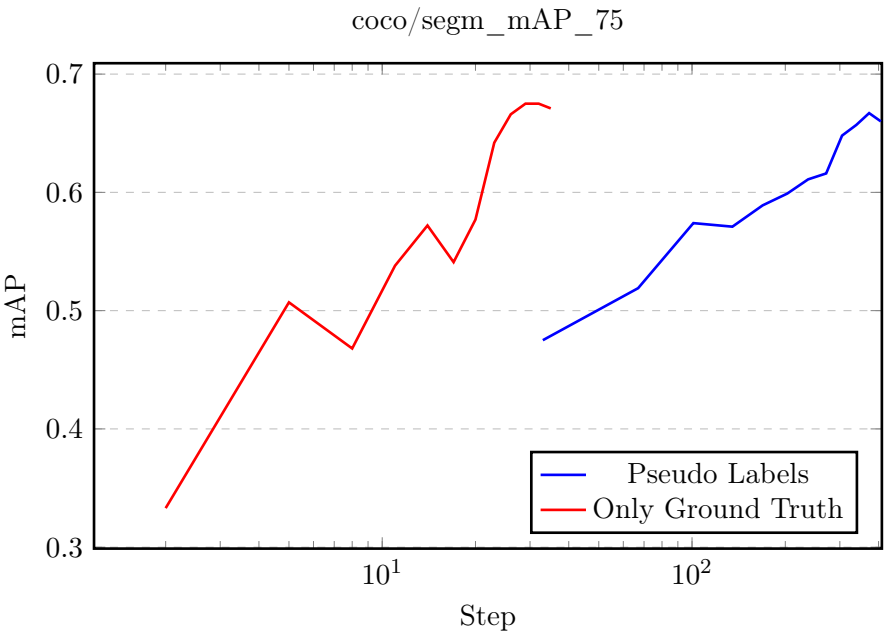


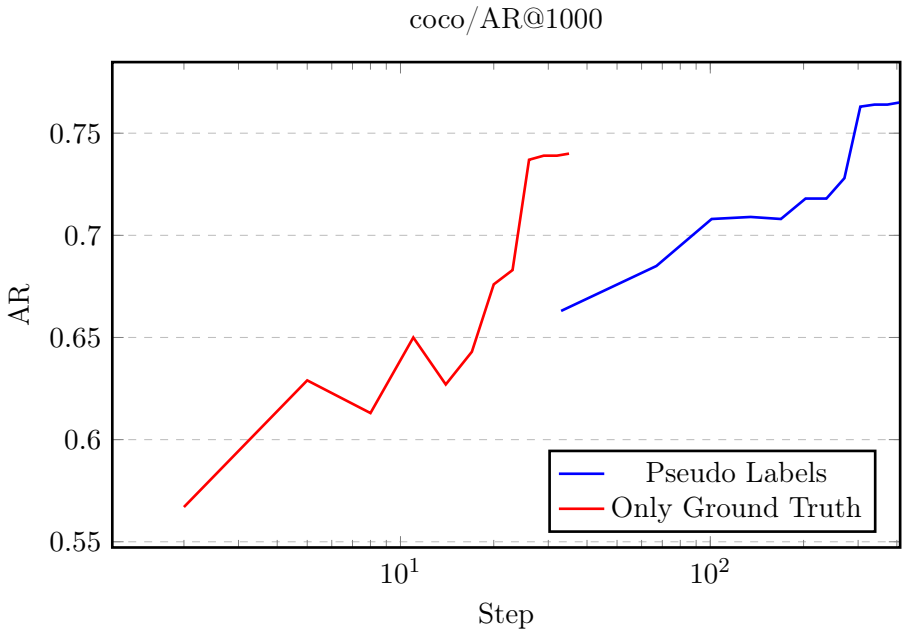
Figure 6.11: Learning Rate During Mask R-CNN Training

6.2.4.2 Model Trained with Only Ground Truth

In order to assess the impact of unlabeled data on the Mask R-CNN network, the network was trained again with the same configuration but only manually labeled ground truth. The training required 12 epochs, with only two steps per epoch, and took 29 minutes and 36 seconds to converge, as listed in Table 6.3. The learning rate, loss, mAP, and AR are also illustrated in the same places of Figures 6.11 to 6.13



(a) mAP



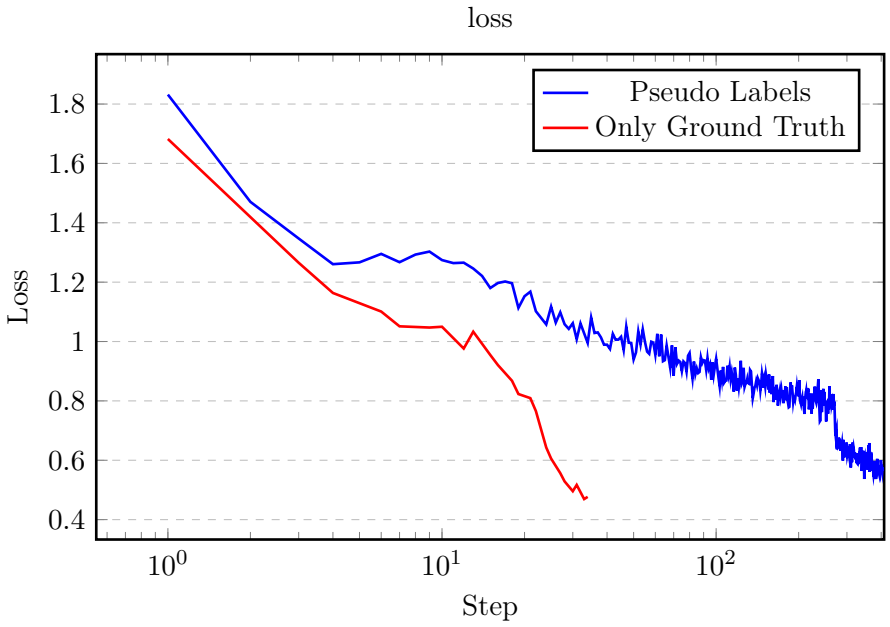
(b) AR

Figure 6.12: mAP and AR During Mask R-CNN Training

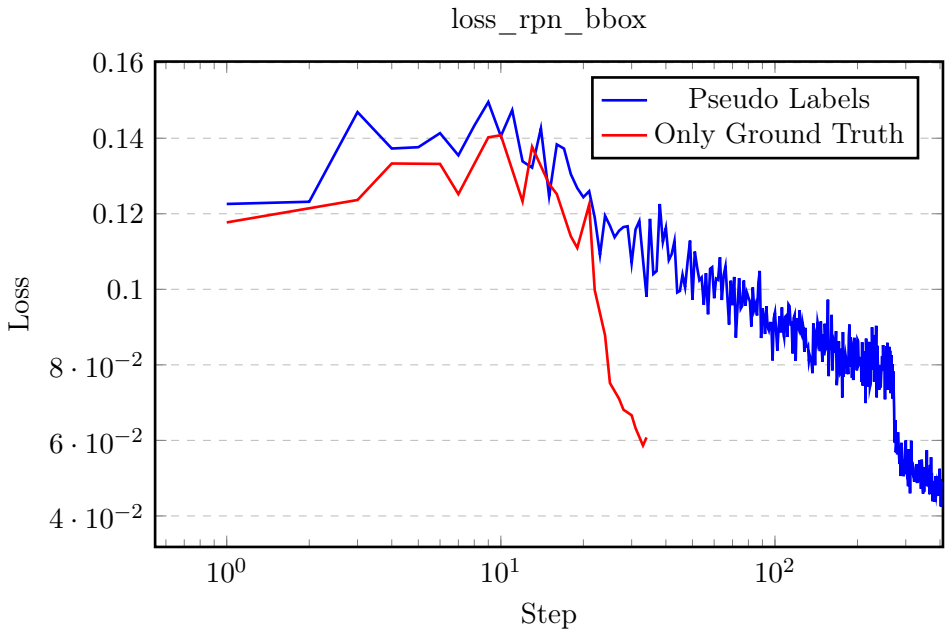
6.2.4.3 Comparison between Two Models

The model trained with both pseudo labels and ground truth data exhibited similar performance to the model trained exclusively with ground truth data. This indicates that the pseudo labels generated by the RTMDet-SAM workflow have established a reliable pipeline for semi-supervised learning, effectively leveraging the unlabeled data.

It is notable that the RPN loss of the network trained with pseudo data is significantly lower. As the RPN is a vital component that extracts the region proposals in the first stage of Mask R-CNN, the lower RPN loss results in higher recall performance, as evidenced by subsequent analysis.



(a) Overall Loss



(b) RPN Loss

Figure 6.13: Overall Loss and RPN Loss During Mask R-CNN Training

In terms of mAP, a slight decline in performance is observed in the model trained with pseudo labels, as shown in Table 6.3. This could be attributed to the noise introduced by the pseudo labels. In some cases, SAM is confused by the ambiguous prompts provided by RTMDet, resulting in inaccurate segmentation, as observed in the first sample in Figure 6.8,

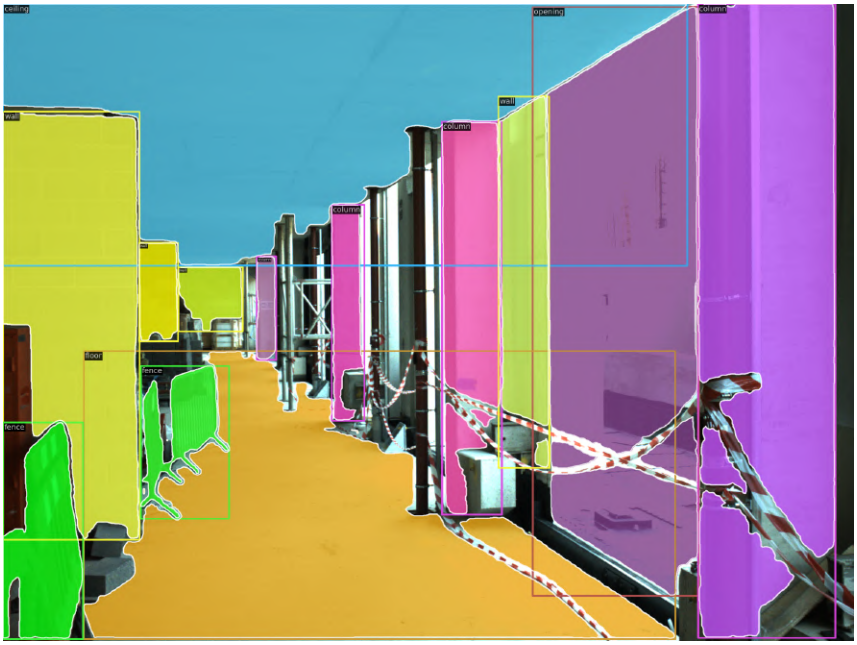
where the segment of a fence is defective. This is also reflected in the higher loss in the model trained with pseudo labels, indicating that some defective data influenced the convergence.

However, due to the aid of pseudo labels, the recall performance improved by 2.5%. The model is capable of extracting the RoI more comprehensively.

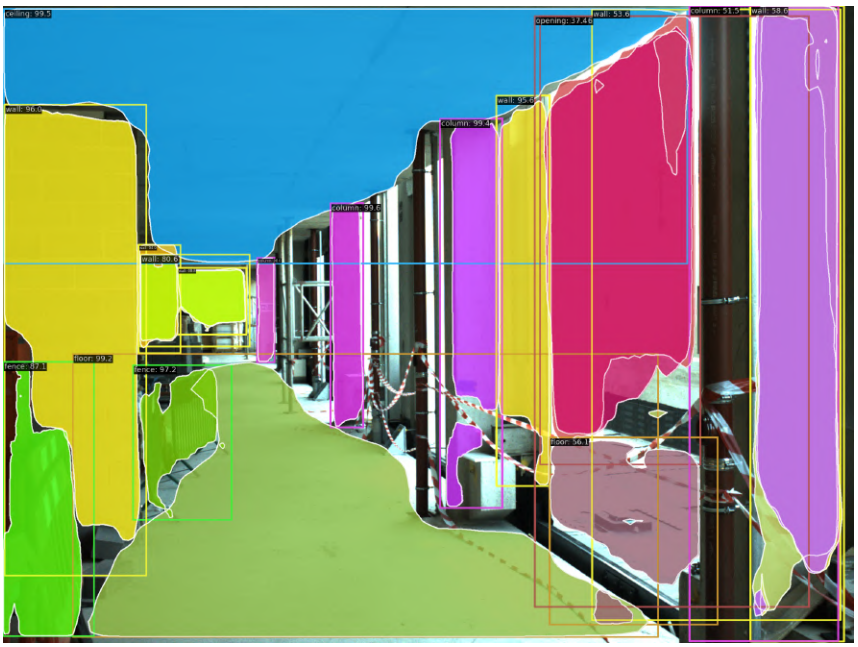
	Final Learning Rate	**Overall Loss**	**RPN Loss**	**mAP**	**AR**	**Training Time**
With Pseudo Labels	2×10^{-4}	0.5768	0.04897	0.66	0.765	7h4m
Only Ground Truth	2.127×10^{-4}	0.4742	0.06	0.671	0.741	29m36s

Table 6.3: Metrics of Mask R-CNN at the Final Evaluation after 12 Epochs

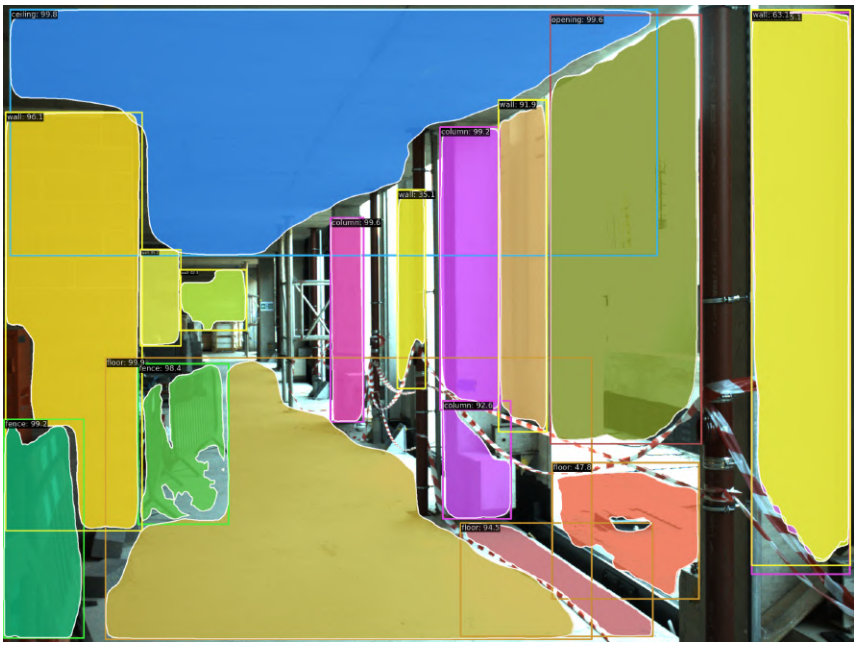
The most significant improvement is reflected in the robustness of the model. As demonstrated in Figure 6.14, the samples of inference with two models indicate that the confidence scores of certain segments proposed by the model trained exclusively with the ground truth range between 0.35 and 0.5, while the model trained with pseudo labels exhibits higher confidence about these segments with scores exceeding 0.9. The pseudo labels generated by *teacher* models certainly enhanced the performance of Mask R-CNN.



(a) Ground Truth



(b) Mask R-CNN without Pseudo Labels



(c) Mask R-CNN with Pseudo Labels

Figure 6.14: Samples of Ground Truth and Inference Results with 2 Mask R-CNN Models

6.2.4.4 Inference Speed of Models

Both Mask R-CNN models are based on the same architecture. Therefore, their inference speeds should not differ significantly. The time taken for inference per image is presented in Table 6.4. However, as a student model, its inference speed is significantly faster than that of the RTMDet-SAM workflow. This indicates that the pseudo labeling workflow alone is not suitable as a practical segmentation model for BIM applications.

	Mask R-CNN (Pseudo Labels)	Mask R-CNN (Ground Truth)	RTMDet-SAM
Seconds per Image	0.330	0.345	16.494

Table 6.4: Time Consumed for Inference per Image

6.3 Experiment of Zero-shot Approach

The performance of the workflow is challenging to optimize due to the inflexibility introduced by the use of pretrained models without fine-tuning. The output appears to be unstable, even with slight changes in thresholds or prompt text. It is difficult to conduct a rigorous analysis of the performance due to the arbitrary selection of prompt text. The results of the experiment are presented after hyperparameter tweaking, and they contain a significant number of defects. Consequently, this approach is at odds with the initial objective of reducing manual intervention. Nevertheless, open-vocabulary object detection remains a cutting-edge area of research, with the potential for more robust and effective models in the future.

The samples of inference results are demonstrated in Figure 6.15, which are inferred with the same images as Figure 6.8.

(a) Inference Results (b) Ground Truth

Figure 6.15: Comparison between Ground Truth and Inference with Zero-shot Approach

Chapter 7

Discussion & Conclusion

7.1 Recall the Objectives

The objective of the research conducted in the thesis is outlined in section 1.2, which presents two primary research directions with specific targets. After completing all the explorations, it is necessary to revisit the initial objectives to assess whether they have been met.

1. ***Identifying the specific types of data necessary to accurately interpret and understand scenes in construction environments.***

   chapter 3 discusses several types of data collected in building environments. The primary goal of this data collection is to better serve design, construction, and facility management. In this process, BIM serves as a nexus that integrates all the information in one place. Conversely, 4D-BIM poses a greater challenge to data collection, where real-time data of the structures must be updated. The registration of the real components to the elements in BIM is one of the obstacles. The prerequisite for solving this problem is to establish a robust mapping between visual data and BIM elements. The visual data captured in the construction environment varies from 2D to 3D. However, the most prevalent data format is still conventional 2D RGB images. Its abundance is prominent compared to those data that require special devices for capturing. Besides, it can also benefit from fruitful research on scene understanding algorithms.

   Therefore, the RGB image is the most necessary data for the development of AEC-specific scene understanding algorithms.

2. ***Exploring methods and tools to streamline the creation of high-quality segmentation data, minimizing time and effort.***

   In subsection 4.2.2, various data annotation tools specific to computer vision tasks are described and discussed. Following years of evolution, these tools have evolved from toys used in laboratories to professional platforms with embedded project management functions. More importantly, recently developed tools attempt to facilitate annotation work by leveraging AI algorithms. CVAT is one of the best examples of such a platform. In the context of this thesis, the annotation of segments greatly benefits from the use of CVAT, as it is otherwise almost impossible to generate such an amount of high-quality segments within a limited time. Indeed, their efforts have made this laborious work less onerous, but the manual data labeling process remains one in which the outcome is linearly related to the amount of time and effort invested.

Compared to the substantial amount of data continuously captured in the construction environments, manual labeling is inadequate. This is exactly the rationale behind the adoption of semi-supervised learning methods.

3. ***Generating more data of the same quality as ground truth without manual intervention.***

   The paucity of data available for training is a persistent challenge in the field of machine learning. Researchers have proposed the use of synthetic data generation based on virtual environment technology as a means of acquiring more data with segmentation information (Bauer et al., 2024). However, it is crucial to prioritize the utilization of data captured in real-world construction environments, which contain valuable information, before synthesizing additional raw data. This thesis aims to find a viable approach for automatically generating segmentation information. A total of 42,933 segment annotations were created by leveraging 2,096 manual segment annotations. The manual intervention was minimized in the process, and the quality of the synthesized segmentation information is comparable to that of the ground truth.

4. ***Evaluating the performance of the segmentation model under conditions with limited labeled data.***

   While not absolute, the performance of the semantic segmentation model is somewhat related to the scale of the parameters of the model. The deeper the network is, the more parameters it has, and consequently, it requires more data to converge. The Mask R-CNN network, applied as the *student* model, is a two-stage algorithm that uses a conventional CNN architecture with 62.63M parameters. Without an adequate number of data, it would be difficult for such a model to achieve a reasonable performance. The experimental results indicate that the model trained on manually labeled data exhibits inferior performance in terms of recall and confidence scores, i.e., accuracy and robustness. In comparison to state-of-the-art models, the Mask R-CNN is relatively simpler, which may result in even inferior performance when other sophisticated models, such as transformer-based models, are applied. It is therefore essential to have more data when handling such models.

5. ***Examining the impact of pseudo labels generated through the semi-supervised method on the ability of the segmentation model to generalize and enhance accuracy.***

   As demonstrated in chapter 6, the Mask R-CNN trained with pseudo labels generated through the RTMDet-SAM workflow exhibits superior recall, indicating enhanced capacity for comprehensive object detection and accurate segmentation. It is noteworthy that the model exhibits a slight decline in precision. This may be attributed to the presence of erroneous instances in the pseudo labels, which could potentially impede model training. Another highlight of this approach is the improvement of generalization exhibited by the model. In the context of dealing with previously unseen data, the higher confidence scores lead to more stable and reliable results.

This aligns perfectly with the initial objective of augmenting model training with unlabeled data.

6. ***Exploring techniques to further automate the generation of pseudo labels, reducing the need for manual intervention.***

   The emergence of the OVD algorithms offers a promising avenue for pseudo-label generation. A novel pipeline for generating pseudo segments from semantically rich text has been established with Grounding DINO. This method entirely alleviates the burden of manual data labeling, yet it also results in the loss of control over the quality of the generated pseudo labels. As the models utilized in this workflow do not necessitate training, the sole hyperparameters that can be adjusted are two thresholds and a list of enriched text labels employed for prompting the objects. Even a minor alteration to the thresholds or text prompts can result in markedly disparate outcomes. Pseudo labels generated using coincidentally optimal hyperparameters appear promising, yet the workflow necessitates further optimization.

7.2 Contribution

In this thesis, three primary contributions of a new dataset and new workflows are presented alongside several minor findings. It is anticipated that these contributions will serve as a foundation for further research on scene understanding in construction environments.

- A new dataset[1] with construction-related objects segment annotations based on RGB images captured in construction environments is derived.

- The pseudo labels[2] over the entire RGB dataset of ConSLAM Sequence 2 were generated through the RTMDet-SAM workflow. The pseudo labels exhibited a quality that was comparable to the ground truth, which could be utilized for other research purposes.

- Although the annotations with 19 classes[3] are proven excessive and imbalanced, the ground truth information created in this process still has the potential to be enhanced and further utilized.

- The proposed RTMDet-SAM[4] workflow generates pseudo labels in the form of segmentation masks with near-ground-truth quality. Nevertheless, there is still room for improvement in the performance of this workflow.

[1] Published at https://huggingface.co/datasets/erwinqi/conslam_seq2_segmentation_gt
[2] Published at https://huggingface.co/datasets/erwinqi/conslam_seq2_segmentation_pseudo
[3] Masks represent category IDs, published at https://huggingface.co/datasets/erwinqi/conslam_seq2_19classes_segmentation_gt
[4] The retrained RTMDet model is published at https://huggingface.co/erwinqi/rtmdet_tiny_8xb32-300e_conslam

- A semantic segmentation model, Mask R-CNN[5], was trained using a dataset augmented by unlabeled data with pseudo labels, demonstrating superior performance compared to a model trained only on ground truth.

- The experiment with Grounding DINO suggests that OVD models have the potential to be applied in pseudo labeling and even directly in scene understanding.

7.3 Limitation & Outlook

Although the explorations conducted in this research have yielded a multitude of outcomes, there are still numerous aspects that this thesis did not cover. These are either constrained by the time available for the research or by the rudimentary stage of the algorithms employed in the research. The limitation of this thesis may provide some insights for future research.

- Although annotation with 19 classes is not preferable in this thesis, its issue of imbalanced categories could still be addressed by annotating substantially more images. This task is challenging. However, if successfully completed, the resulting dataset would be more valuable than the current 8-class dataset due to its richer semantic information.

- The ConSLAM dataset utilized in this research comprises frames captured within relatively short time intervals, resulting in homogeneity among adjacent image frames. This limitation restricts the generalization of RTMDet. If the RTMDet-SAM workflow were applied to a dataset containing more diverse construction environment scenes, its performance could potentially improve due to enhanced generalization.

- As Kirillov et al. (2023) acknowledged, SAM is not anticipated to perform flawlessly with fine structures in images. However, objects in construction environments often possess fine structures, which poses a challenge for SAM. If a substantial number of labeled images from construction environments were available, SAM could potentially be fine-tuned for AEC-specific tasks. This fine-tuning is anticipated in the future when sufficient data becomes available.

- Image segmentation is also a prominent topic in biology, particularly for segmenting bioimages that contain a large amount of fine details. Berg et al. (2019) have developed the *ilastik* toolkit for interactive image segmentation, which, similar to SAM, offers superior performance in handling fine structures. The workflow proposed in this thesis could benefit from advancements in bioimage segmentation algorithms. Exploring this combination could be an excellent direction for future research.

- As observed in the metrics of Mask R-CNN trained with pseudo labels, the mAP declined while the final loss was somewhat higher compared to the model trained

[5]Model trained on pseudo labels is published at https://huggingface.co/erwinqi/mask-rcnn_x101-32x4d_fpn_1x_conslam_pseudo, the model only on ground truth is published at https://huggingface.co/erwinqi/mask-rcnn_x101-32x4d_fpn_1x_conslam_gt

exclusively with ground truth labels. This decline may be attributed to defects in the pseudo labels. If these erroneous data could be filtered out, the pseudo labels would be of higher quality and cause less confusion for the model trained on them. Developing an appropriate filtering method is a promising direction for future improvement.

- The Mask R-CNN model is somewhat outdated, especially in light of emerging novel architectures for image segmentation. The purpose of selecting Mask R-CNN was to identify a model sophisticated enough to serve as a *student* model. The initial choice was Mask2Former (Cheng et al., 2022), a state-of-the-art model with a higher potential for improvement through training with pseudo labels. However, transformer-based models such as Mask2Former require VRAM greater than 11GB. If better hardware is available, it is highly recommended to use Mask2Former as the *student* model to achieve superior performance.

Bibliography

Afzal, M., & Shafiq, M. T. (2021). Evaluating 4d-bim and vr for effective safety communication and training: A case study of multilingual construction job-site crew. *Buildings*, *11*(8), 319. https://www.mdpi.com/2075-5309/11/8/319

Alloghani, M., Al-Jumeily, D., Mustafina, J., Hussain, A., & Aljaaf, A. J. (2020). A systematic review on supervised and unsupervised machine learning algorithms for data science. In M. W. Berry, A. Mohamed, & B. W. Yap (Eds.), *Supervised and unsupervised learning for data science* (pp. 3–21). Springer International Publishing. https://doi.org/10.1007/978-3-030-22475-2_1

Amit, Y., Felzenszwalb, P., & Girshick, R. (2021). Object detection. In K. Ikeuchi (Ed.), *Computer vision: A reference guide* (pp. 875–883). Springer International Publishing. https://doi.org/10.1007/978-3-030-63416-2_660

Andoh, A. R., Su, X., & Cai, H. (2012). A framework of rfid and gps for tracking construction site dynamics. *Construction Research Congress 2012*, 818–827. https://doi.org/doi:10.1061/9780784412329.083

Badrinarayanan, V., Kendall, A., & Cipolla, R. (2017). Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, *39*(12), 2481–2495.

Bauer, A., Trapp, S., Stenger, M., Leppich, R., Kounev, S., Leznik, M., Chard, K., & Foster, I. (2024). Comprehensive exploration of synthetic data generation: A survey. *arXiv preprint arXiv:2401.02524*.

Berg, S., Kutra, D., Kroeger, T., Straehle, C. N., Kausler, B. X., Haubold, C., Schiegg, M., Ales, J., Beier, T., Rudy, M., Eren, K., Cervantes, J. I., Xu, B., Beuttenmueller, F., Wolny, A., Zhang, C., Koethe, U., Hamprecht, F. A., & Kreshuk, A. (2019). Ilastik: Interactive machine learning for (bio)image analysis. *Nature Methods*, *16*(12), 1226–1232. https://doi.org/10.1038/s41592-019-0582-9

Birajdar, A., Agarwal, H., Bolia, M., & Gupte, V. (2019). Image compression using run length encoding and its optimisation. *2019 Global Conference for Advancement in Technology (GCAT)*, 1–6. https://doi.org/10.1109/GCAT47503.2019.8978464

Bréhéret, A. (2017). Pixel annotation tool. https://github.com/abreheret/PixelAnnotationTool

Brooks, J. (2019). Coco annotator. https://github.com/jsbroks/coco-annotator/

Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., & Girdhar, R. (2022). Masked-attention mask transformer for universal image segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 1290–1299.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., & Schiele, B. (2016). The cityscapes dataset for semantic urban scene understanding. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3213–3223.

Cruz, S., Hutchcroft, W., Li, Y., Khosravan, N., Boyadzhiev, I., & Kang, S. B. (2021). Zillow indoor dataset: Annotated floor plans with 360deg panoramas and 3d room layouts. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2133–2143.

Csurka, G., Volpi, R., & Chidlovskii, B. (2022). Semantic image segmentation: Two decades of research. *Foundations and Trends® in Computer Graphics and Vision*, *14*(1-2), 1–162.

CVAT.ai Corporation. (2024). Computer vision annotation tool (cvat) (v2.14.0). *Zenodo*. https://doi.org/https://doi.org/10.5281/zenodo.11239396

Datagen. (2023). Semantic segmentation: A quick guide. https://datagen.tech/guides/image-annotation/semantic-segmentation

Davtalab, O. (2017). Benefits of real-time data driven bim for fm departments in operations control and maintenance. In *Computing in civil engineering 2017* (pp. 202–210).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. *2009 IEEE conference on computer vision and pattern recognition*, 248–255.

Deng, L. (2012). The mnist database of handwritten digit images for machine learning research [best of the web]. *IEEE signal processing magazine*, *29*(6), 141–142.

Deng, Z., Sun, H., Zhou, S., Zhao, J., Lei, L., & Zou, H. (2018). Multi-scale object detection in remote sensing imagery with convolutional neural networks. *ISPRS journal of photogrammetry and remote sensing*, *145*, 3–22.

Duque-Arias, D., Velasco-Forero, S., Deschaud, J.-E., Goulette, F., Serna, A., Decencière, E., & Marcotegui, B. (2021). On power jaccard losses for semantic segmentation [Engineering Sciences [physics] Engineering Sciences [physics]/Signal and Image processingConference papers]. *VISAPP 2021 : 16th International Conference on Computer Vision Theory and Applications*. https://hal.science/hal-03139997

Dutta, A., & Zisserman, A. (2019). The via annotation software for images, audio and video. *Proceedings of the 27th ACM international conference on multimedia*, 2276–2279.

Ellis, G. (2023). Laser scanning in construction: Everything you need to know. https://www.autodesk.com/blogs/construction/laser-scanning-in-construction

Everingham, M., Eslami, S. A., Van Gool, L., Williams, C. K., Winn, J., & Zisserman, A. (2015). The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, *111*, 98–136.

Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., Wiesbeck, W., & Dietmayer, K. (2020). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, *22*(3), 1341–1360.

Geotechnical Observations. (2024a). Inclinometers — geotechnical observations. https://www.geo-observations.com/inclinometers

Geotechnical Observations. (2024b). Strain and load — geotechnical observations. https://www.geo-observations.com/strain-and-load

Hamma-adama, M., Kouider, T., & Salman, H. (2020). Analysis of barriers and drivers for bim adoption. *International journal of BIMa and engineering science*, *3*(1).

Henderson, P., & Ferrari, V. (2017). End-to-end training of object class detectors for mean average precision. *Computer Vision–ACCV 2016: 13th Asian Conference on Computer Vision, Taipei, Taiwan, November 20-24, 2016, Revised Selected Papers, Part V 13*, 198–213.

Intel RealSense. (2020). Beginner's guide to depth (updated). https://www.intelrealsense.com/beginners-guide-to-depth

Kadambi, A., Bhandari, A., & Raskar, R. (2014). 3d depth cameras in vision: Benefits and limitations of the hardware: With an emphasis on the first-and second-generation kinect models. In *Computer vision and machine learning with rgb-d sensors* (pp. 3–26).

Kaloop, M. R., Elbeltagi, E., Hu, J. W., & Elrefai, A. (2017). Recent advances of structures monitoring and evaluation using gps-time series monitoring systems: A review. *ISPRS International Journal of Geo-Information*, *6*(12), 382. https://www.mdpi.com/2220-9964/6/12/382

Kawamura, R. (2017). Rectlabel. https://rectlabel.com/

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., & Lo, W.-Y. (2023). Segment anything. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 4015–4026.

Krawetz, N. (2011). Looks like it. https://www.hackerfactor.com/blog/index.php?/archives/432-Looks-Like-It.html

Labelbox. (2024). Labelbox. https://labelbox.com

Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. *Workshop on challenges in representation learning, ICML*, *3*, 896.

Lee, H.-S., Lee, K.-P., Park, M., Baek, Y., & Lee, S. (2012). Rfid-based real-time locating system for construction safety management. *Journal of Computing in Civil Engineering*, *26*(3), 366–377. https://doi.org/doi:10.1061/(ASCE)CP.1943-5487.0000144

Leica Geosystems. (2024). 3d laser scanning solutions for surveyors. https://leica-geosystems.com/industries/pure-surveying/surveying-solutions/3d-laser-scanning-solutions-for-surveyors

Lhoest, Q., del Moral, A. V., Jernite, Y., Thakur, A., von Platen, P., Patil, S., Chaumond, J., Drame, M., Plu, J., & Tunstall, L. (2021). Datasets: A community library for natural language processing. *arXiv preprint arXiv:2109.02846*.

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft coco: Common objects in context. *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, 740–755.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., & Zhu, J. (2023). Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., & Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022.

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

Lyu, C., Zhang, W., Huang, H., Zhou, Y., Wang, Y., Liu, Y., Zhang, S., & Chen, K. (2022). Rtmdet: An empirical study of designing real-time object detectors. *arXiv e-prints*, arXiv: 2212.07784.

MMYOLO Contributors. (2022). MMYOLO: OpenMMLab YOLO series toolbox and benchmark. https://github.com/open-mmlab/mmyolo

Padilla, R., Netto, S. L., & Da Silva, E. A. (2020). A survey on performance metrics for object-detection algorithms. *2020 international conference on systems, signals and image processing (IWSSIP)*, 237–242.

Pang, Y., & Cao, J. (2019). Deep learning in object detection. In X. Jiang, A. Hadid, Y. Pang, E. Granger, & X. Feng (Eds.), *Deep learning in object detection and recognition* (pp. 19–57). Springer Singapore. https://doi.org/10.1007/978-981-10-5152-4_2

Polyga. (2024). 3d scanning 101: Size limitations of structured light 3d. https://polyga.com/blog/3d-scanning-101-size-limitations-of-structured-light-3d-scanning

Ramón, J. E., Gandía-Romero, J. M., Bataller, R., López, J. A., Valcuende, M., & Soto, J. (2022). Real-time corrosion monitoring of an ultra-high performance fibre-reinforced concrete offshore raft by using an autonomous sensor system. *Structural Control and Health Monitoring*, *29*(11), e3102. https://doi.org/https://doi.org/10.1002/stc.3102

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

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, *28*.

Rezatofighi, H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I., & Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 658–666.

Roboflow Universe Projects. (2023). Construction site safety dataset [visited on 2024-04-12]. https://universe.roboflow.com/roboflow-universe-projects/construction-site-safety

Ronneberger, O., Fischer, P., & Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. *Medical image computing and computer-assisted in-*

tervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18, 234–241.

Russell, B. C., Torralba, A., Murphy, K. P., & Freeman, W. T. (2008). Labelme: A database and web-based tool for image annotation. *International journal of computer vision*, *77*, 157–173.

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). *Bim handbook: A guide to building information modeling for owners, designers, engineers, contractors, and facility managers*. John Wiley & Sons.

Sharma, G. (2023). A gentle introduction to semi supervised learning. *Medium*. https://medium.com/@gayatri_sharma/a-gentle-introduction-to-semi-supervised-learning-7afa5539beea

Shehzadi, T., Hashmi, K. A., Stricker, D., & Afzal, M. Z. (2023). 2d object detection with transformers: A review. *arXiv preprint arXiv:2306.04670*.

Shotton, J., & Kohli, P. (2020). Semantic image segmentation: Traditional approach. In *Computer vision: A reference guide* (pp. 1–4). Springer.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

Supervisely. (2023). Supervisely computer vision platform. https://supervisely.com

Thisanke, H., Deshan, C., Chamith, K., Seneviratne, S., Vidanaarachchi, R., & Herath, D. (2023). Semantic segmentation using vision transformers: A survey. *Engineering Applications of Artificial Intelligence*, *126*, 106669.

Trzeciak, M., Pluta, K., Fathy, Y., Alcalde, L., Chee, S., Bromley, A., Brilakis, I., & Alliez, P. (2023). Conslam: Periodically collected real-world construction dataset for slam and progress monitoring, 317–331. https://doi.org/10.1007/978-3-031-25082-8_21

Valero, E., Adán, A., & Cerrada, C. (2015). Evolution of rfid applications in construction: A literature review. *Sensors*, *15*(7), 15988–16008. https://www.mdpi.com/1424-8220/15/7/15988

van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*(2), 373–440. https://doi.org/10.1007/s10994-019-05855-6

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*.

Wang, C.-Y., Bochkovskiy, A., & Liao, H.-Y. M. (2023). Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 7464–7475.

Wang, J., Sun, W., Shou, W., Wang, X., Wu, C., Chong, H.-Y., Liu, Y., & Sun, C. (2015). Integrating bim and lidar for real-time construction quality control. *Journal of Intelligent & Robotic Systems*, *79*(3), 417–432. https://doi.org/10.1007/s10846-014-0116-8

Yan, P., Li, G., Xie, Y., Li, Z., Wang, C., Chen, T., & Lin, L. (2019). Semi-supervised video salient object detection using pseudo-labels. *Proceedings of the IEEE/CVF international conference on computer vision*, 7284–7293.

Zhang, H., Li, F., Liu, S., Zhang, L., Su, H., Zhu, J., Ni, L. M., & Shum, H.-Y. (2022). Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*.

Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., & Torr, P. H. (2021). Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 6881–6890.

Zhu, M. (2004). Recall, precision and average precision. *Department of Statistics and Actuarial Science, University of Waterloo, Waterloo*, *2*(30), 6.

Zou, Y., Zhang, Z., Zhang, H., Li, C.-L., Bian, X., Huang, J.-B., & Pfister, T. (2020). Pseudoseg: Designing pseudo labels for semantic segmentation. *arXiv preprint arXiv:2010.09713*.

Declaration

I hereby affirm that I have independently written the thesis submitted by me and have not used any sources or aids other than those indicated.

Munich, June 1 2024 

Location, Date, Signature