Retrieving Urban Morphology from Satellite Images using Machine Learning: A Fast-Response Instance Semantic Segmentation Approach with Building Height Regression

Anonymous COMP4901V Group 5 Kan Wan On, Narayan Aadityavardhan submission

Paper ID *****

Abstract

We present a reliable and scalable pipeline for fast-response instance semantic segmentation and building height regression from satellite images using machine learning techniques. The proposed approach employs an advanced version of the U-Net model, U-Net 3+, and the DeTR Transformer Model as frontend object detection algorithm. We also integrate a refinement CRF algorithm and polygonization algorithm as backend components of the pipeline to enhance segmentation accuracy and facilitate interpretation and analysis of the results. The project's outcome will benefit modern morphology analysis and enable rapid, comprehensive, automatic, and objective monitoring for important scientific disciplines, such as urban planning and environmental science research [7].

1. Introduction

This project aims to develop a reliable and scalable pipeline for fast-response instance semantic segmentation and building height regression from satellite images using machine learning techniques. The approach involves using the DeTR Transformer Model as the front-end object detection algorithm to extract image tiles from raw geospatial data. The U-Net 3+ model is then used for pixel-wise prediction image segmentation on the cropped patches around each bounding box to extract the building footprints. The pipeline also includes a refinement CRF algorithm and polygonization algorithm as backend components to enhance segmentation accuracy and facilitate interpretation and analysis of the results.

Our previous work has demonstrated the success of the U-Net 3+ model in recognizing building shapes with a binary accuracy of near 88% on the Hong Kong Island and Kowloon regions. We have extended the features of the current model to accomplish more accurate and sophisticated tasks, such as instance semantic segmentation regressing building height. This will enable rapid, comprehensive, automatic, and objective monitoring for important scientific disciplines, such as urban planning and environmental science research.

By using the DeTR Transformer Model as a stable data mining tool, we can efficiently and accurately extract image tiles even for large-scale satellite imagery. The integration of the refinement CRF algorithm and polygonization algorithm into the pipeline ensures spatial consistency and facilitates the interpretation and analysis of the results.

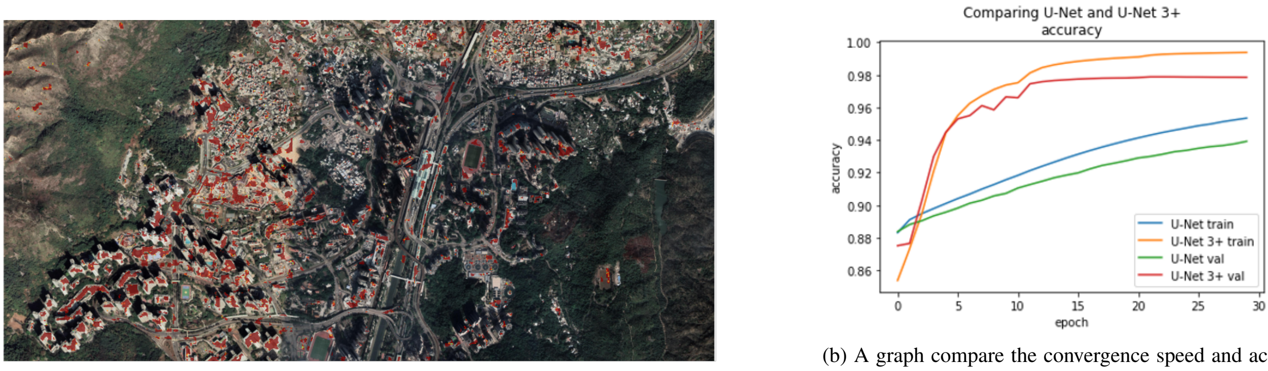
The proposed approach presents a reliable and scalable pipeline for fast-response instance semantic segmentation and building height regression from satellite images using machine learning techniques. The project's outcome will benefit modern morphology analysis and enable rapid, comprehensive, automatic, and objective monitoring for important scientific disciplines.

1.1. Establishments of U-Net 3+ Transfer Learning

U-Net 3+ [4] is an extension of the U-Net architecture that utilizes skip connections to improve the flow of information between the encoder and decoder blocks. This results in faster convergence and improved segmentation performance, particularly for large and complex remote sensing datasets.

The addition of skip connections allows U-Net 3+ to leverage more information from the input image and make better use of the available training data. This is particularly important in remote sensing, where images can be large and complex, and where recognizing buildings can be difficult due to variations in lighting, shadows, and occlusions.

The performance of the model should be evaluated on a representative set of test images, and the architecture and training process should be iteratively refined to achieve the best possible results.



(a) A grand view of the overall combined output of Tuen Mun region.

(b) A graph compare the convergence speed and accuracy of U-Net 3+ against orignal U-Net

Figure 1. The preliminary U-Net 3+ model achieves enhances quantitative results and sustain promising qualitative results.

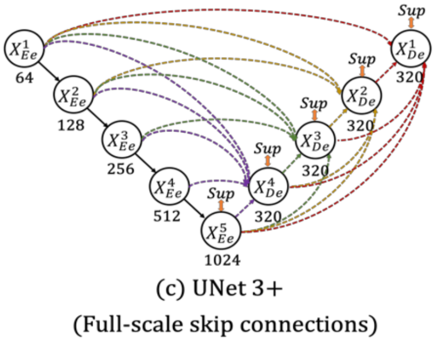


Figure 2. A diagram demonstrate the U-Net 3+ architecture.

1.2. Preliminary Results of U-Net 3+ Segmentation

In our previous experiments, we evaluated the performance of the U-Net 3+ architecture for urban satellite imagery instance segmentation on the Kowloon and Hong Kong Island regions. Our results demonstrate that the U-Net 3+ architecture achieves a significantly improved validation accuracy score of 87.88%, compared to the original U-Net architecture, which achieved a validation accuracy score of 86.78%. This improvement in accuracy was achieved in at least 3 times shorter training time, which makes the U-Net 3+ architecture an attractive option for building robust remote sensing image instance segmentation pipelines that can handle large and complex datasets efficiently.

We further conducted a qualitative prediction on the Tuen Mun region, which revealed some challenges in predicting non-urban areas with limited training data. In these outlier areas, the model's performance was suboptimal, possibly due to the lack of related samples in the training dataset. However, in most regions, particularly in urban areas, the U-Net 3+ architecture demonstrated robust performance, accurately segmenting buildings, roads, and other urban features.

To gain further insights into the U-Net 3+ architecture's performance, we conducted an ablation study, which revealed that the addition of the dense blocks and attention gates in the U-Net 3+ architecture contributes significantly to the improved accuracy and reduced training time. The dense blocks facilitate feature reuse and reduce the number of parameters in the model, while the attention gates improve the model's ability to focus on relevant features, leading to better performance.

Our experiments demonstrate the effectiveness of the U-Net 3+ architecture for urban satellite imagery instance segmentation and highlight its potential for building efficient and accurate remote sensing image pipelines. Future work can explore ways to improve the model's performance in non-urban areas and extend the evaluation to other regions and datasets.

1.3. Driving to A Breaking-Down Image Instance Segmentation Pipeline

Image instance segmentation in real-world scenarios with unstable data sources is challenging due to missing ground truth and mismatch between input and ground truth. We propose a pipeline integrating frontend objection detection, building height prediction, and CRF refinement to address these challenges.

Our dataset from open sources had unstable quality with missing ground truth and mismatch. Frontend objection detection identified relevant samples to improve dataset quality, reducing impact of missing ground truth and mismatch. Adding building height prediction to objection detection improved accuracy and efficiency. CRF refinement using context improved result accuracy.

Breaking segmentation into frontend detection and CRF refinement is practical for limited resources or unstable data. A multitasking model is computationally expensive and may not work. Height prediction with detection gave accurate results without a multitasking model.

Our pipeline shows using frontend detection, CRF refinement, and height prediction improves segmentation accuracy and efficiency, especially for unstable data. Improving dataset quality and adding relevant samples enables reliable computer vision applications.

Frontend detection found relevant samples in unstable data, reducing missing ground truth and mismatch impact. Adding height prediction to detection improved accuracy and efficiency. CRF refinement improved result accuracy using context. Our pipeline addressed challenges from unstable data sources and limited resources. Breaking into frontend detection and CRF refinement was practical. Height prediction with detection avoided a computationally expensive multitasking model.

We demonstrate effectiveness of frontend detection, CRF refinement, and height prediction for image instance segmentation with unstable data sources or limited resources. Techniques to improve dataset quality and add relevant samples enable reliable computer vision. Our pipeline improves accuracy and efficiency, reducing impact of missing ground truth and mismatch for a customized dataset with open unstable sources.

2. Frontend DeTR Model Design and Analysis

The DeTR model [2] is a recent state-of-the-art approach for object detection that employs a transformer-based architecture. Unlike traditional deep learning object detection models, such as the abandoned solution of CenterNet, which rely on complex handcrafted features and multi-stage pipelines, DeTR uses a single end-to-end trainable architecture that simultaneously predicts object bounding boxes and their corresponding class labels.

DeTR's transformer-based architecture enables it to capture and model the complex relationships between objects in an image, allowing for more accurate and efficient object detection. Specifically, DeTR uses a self-attention mechanism that allows it to attend to different parts of the image and capture long-range dependencies between different objects, improving its ability to detect objects in cluttered scenes.

One of the key advantages of DeTR over traditional models is its ability to handle variable numbers of objects in an image without requiring additional post-processing steps. This is achieved by using a set-based prediction approach that predicts objects as unordered sets rather than individually, allowing for a more flexible and efficient model. To accomplish this, DeTR utilizes a learnable permutation matrix that enables it to match predicted object sets with ground-truth object sets, providing a more accurate and robust training signal.

Moreover, DeTR is more applicable for predicting the actual height of the building object, as it allows for the addition of an additional prediction head specifically for this task. This is in contrast to using CenterNet to generate a high-dimensional height prediction map, which can be tedious to adjust and may not accurately capture the actual height of the object. By incorporating an additional prediction head, DeTR is able to more accurately predict the height of objects in an image, which can be particularly useful in applications such as autonomous driving or robotics.

DeTR's transformer-based architecture, set-based prediction approach, and ability to predict object height make it a highly effective and efficient approach for object detection, particularly in scenarios with variable numbers of objects and complex object relationships. Its end-to-end trainable architecture and ability to handle flexible outputs make it a promising model for future object detection tasks.

2.1. A Modified DeTR Transformer Architecture

The DeTR (DEtection TRansformer) model is a state-of-the-art approach for object detection that employs a transformer-based architecture. The transformer architecture enables DeTR to capture and model the complex relationships between objects in an image, allowing for more accurate and efficient object detection.

DeTR's transformer-based architecture consists of an encoder and a decoder, similar to other transformer-based models. The encoder extracts a set of feature maps from the input image, while the decoder produces a set of object queries used for predicting object class and bounding box.

DeTR uses a set-based prediction approach that predicts objects as unordered sets rather than individually, allowing for a more flexible and efficient model. This approach is achieved using an attention mechanism that allows the model to attend to different parts of the input feature maps and capture long-range dependencies between different objects. Specifically, DeTR uses a self-attention mechanism that allows it to attend to different parts of the image, enabling it to capture complex object relationships.

To handle variable numbers of objects in an image, DeTR introduces a learnable permutation matrix that enables it to match predicted object sets with ground-truth object sets. This provides a more accurate and robust training signal. The permutation matrix is learned during training and is used to reorder the predicted object queries based on their similarity to the ground-truth object queries.

In addition to predicting object class and bounding box, DeTR can also predict the actual height of the object, which is a valuable feature for applications such as autonomous driving or robotics. This is achieved by incorporating an additional prediction head specifically for this task.

The DeTR model is summarized by the following equations:

Input image: I

Encoder function: f_{enc}

Decoder function: f_{dec}

Set of object queries: $Q = q_1, q_2, ..., q_n$

Object class prediction: p_{cls}

Object bounding box prediction: p_{box}

Object height prediction: p_{height}

The encoder function is defined as:

$F_{enc}(I) = f_1, f_2, ..., f_n$

where f_i is the i-th feature map extracted from the input image.

The decoder function is defined as:

$F_{dec}(Q, F_{enc}(I)) = p_{cls_i}, p_{box_i}, p_{height_i}{}_{i=1}^{n}$

where p_{cls_i}, p_{box_i}, and p_{height_i} are the predicted class, bounding box, and height for the i-th object query. The set of predictions contains n elements, one for each object query.

The object queries are defined as:

$Q = q_1, q_2, ..., q_n$

where q_i is a learnable vector representing the i-th object query.

The set-based prediction approach is achieved by incorporating a permutation matrix P into the decoder function:

$F_{dec}(Q, F_{enc}(I), P) = p_{cls_i}, p_{box_i}, p_{height_i}{}_{i=1}^{n}$

where p_{cls_i}, p_{box_i}, and p_{height_i} are the predicted class, bounding box, and height for the i-th object query. The set of predictions contains n elements, one for each object query. The permutation matrix P is used to reorder the predicted object queries based on their similarity to the ground-truth object queries.

2.2. Bipartite Matching in DeTR Transformer

Bipartite matching is a problem of finding the optimal assignment of elements from one set to elements from another set, subject to certain constraints. In DeTR, the set of predicted objects and the set of ground-truth objects can be viewed as two disjoint sets, and the problem is to find the best matching between them.

The Hungarian algorithm is a well-known algorithm for solving the bipartite matching problem. It works by constructing an $n \times m$ cost matrix C, where n is the number of predicted objects and m is the number of ground-truth objects. Each entry $C_{i,j}$ in the matrix represents the cost of assigning predicted object i to ground-truth object j. In DeTR, the cost is typically defined as the negative IoU between the predicted bounding box and the ground-truth bounding box.

The algorithm then proceeds in several phases. In each phase, it finds a set of augmenting paths that increase the number of matched pairs and updates the cost matrix accordingly. The algorithm terminates when no further augmenting paths can be found, at which point it returns the optimal assignment of predicted objects to ground-truth objects.

In DeTR, the Hungarian algorithm is used to learn a permutation matrix that reorders the predicted object queries based on their similarity to the ground-truth object queries. This permutation matrix is learned during training and is used to reorder the predicted objects before computing the loss function, ensuring that the predicted objects are matched with their corresponding ground-truth objects.

2.3. DeTR Loss Funcition

DeTR is a transformer-based object detection model that uses the Hungarian algorithm for bipartite matching. The standard object detection loss in DeTR combines the GIoU loss and smooth L1 loss for the bounding box regression task, and binary cross-entropy loss for the object classification task.

The GIoU loss is a modified version of the IoU loss that accounts for the size and position of the predicted bounding box relative to the ground-truth bounding box. The smooth L1 loss is a variant of the L1 loss that is less sensitive to outliers. The total object detection loss for DeTR is defined as:

$L_{obj} = L_{cls} + \alpha L_{box}$

where L_{cls} is the binary cross-entropy loss for object classification, L_{box} is the combination of GIoU loss and smooth L1 loss for bounding box regression, and α is a hyperparameter that controls the balance between the two losses.

The binary cross-entropy loss is defined as:

$L_{cls} = -\frac{1}{N}\sum_{i=1}^{N}[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)]$

where N is the number of objects in the image, y_i is the ground-truth label for the i-th object, and p_i is the predicted probability of the object being present.

To add an MSE loss for building height prediction, the total loss function becomes:

$L_{total} = L_{cls} + \alpha L_{box} + \beta L_{mse}$

where L_{mse} is the mean squared error loss for building height prediction, and β is a hyperparameter that controls the weight of the MSE loss relative to the object detection loss.

The MSE loss for building height prediction can be computed as follows:

$L_{mse} = \frac{1}{N}\sum_{i=1}^{N}(h_i - \hat{h_i})^2$

where h_i is the ground-truth height of the i-th building, and $\hat{h_i}$ is the predicted height of the i-th building.

To penalize the model for predicting objects that do not match any ground-truth objects, or for failing to predict ground-truth objects, the non-matching loss is introduced. The non-matching loss can be computed as follows:

$L_{nm} = \frac{1}{N}\sum_{i=1}^{N}(1 - \sum_{j=1}^{M} y_{ij}) + \frac{1}{M}\sum_{j=1}^{M}(1 - \sum_{i=1}^{N} y_{ij})$

where N is the number of predicted objects, M is the number of ground-truth objects, and y_{ij} is the binary indicator variable that is 1 if predicted object i matches with ground-truth object j, and 0 otherwise.

The final loss function for training the DeTR model with multiple loss functions and the non-matching loss is:

$L_{total} = L_{cls} + \alpha L_{box} + \beta L_{mse} + \gamma L_{nm}$

where L_{cls}, L_{box}, and α are as defined before, L_{mse} and β are as defined for the MSE loss for building height prediction, and L_{nm} and γ are as defined for the non-matching loss.

By combining all these loss functions into a single loss function and tuning the weights of the losses, the DeTR model can be trained to optimize multiple objectives simultaneously while also penalizing the model for non-matching predictions.

3. Post-processing CRF Refinement Model

Building segmentation in remote sensing images is a critical task that has been widely studied in the computer vision community. Despite significant progress in deep learning-based approaches, challenges such as small or partially occluded buildings, shadows, lighting variations, and image shifting still exist. To address these challenges, various techniques have been proposed to refine the segmentation masks and produce accurate and visually appealing polygons.

One such technique is Conditional Random Fields (CRFs), which can improve the spatial coherence and consistency of the segmentation masks generated by deep learning models such as U-Net 3+. By modeling the dependencies between neighboring pixels, CRFs can refine the segmentation masks and produce smoother contours, resulting in more accurate and visually appealing segmentation results. In this section, we introduce the mathematical views of CRFs and their applications in building segmentation in remote sensing images.

Another technique that can improve the accuracy and quality of building segmentation is contour-based polygonization. By incorporating the refined segmentation masks produced by CRFs, contour-based polygonization can produce smoother and more accurate boundaries for the contours, resulting in more visually appealing and accurate polygons. In this section, we also introduce the mathematical views of contour-based polygonization and its applications in building segmentation in remote sensing images.

In addition to producing more accurate and visually appealing polygons, contour-based polygonization also offers the advantage of being more space-efficient than storing the pixel-level segmentation masks. Since the vector data of the polygonization output only includes the vertices and edges of the polygons, it requires significantly less storage space than the pixel-level segmentation masks. This space efficiency can be particularly important in applications where large amounts of remote sensing data need to be stored and processed efficiently. Therefore, contour-based polygonization can be a useful technique for building segmentation in remote sensing images, both in terms of accuracy and storage efficiency.

3.1. Fully Connected CRFs with Gaussian Edge Potentials

Fully Connected Conditional Random Fields (CRFs) [5] with Gaussian Edge Potentials is a technique used in our project to refine the segmentation results obtained from the U-Net 3+ model. This technique uses a CRF model to smooth the segmentation results and improve their accuracy.

The formula for the energy function of a fully connected CRF with Gaussian edge potentials can be expressed as:

$$E(y) = \sum_{i} \psi_u(y_i) + \sum_{i<j} \psi_p(y_i, y_j) \tag{1}$$

where y_i is the label assigned to pixel i, and $\psi_u(y_i)$ is the unary potential of pixel i, which is obtained from the CNN segmentation model. The pairwise potential $\psi_p(y_i, y_j)$ encourages neighboring pixels with similar labels to have a higher probability of being assigned the same label. This pairwise potential can be defined using a Gaussian function as:

$$\psi_p(y_i, y_j) = w_{ij} exp\left(-\frac{|p_i - p_j|^2}{2\theta_\alpha^2} - \frac{|I_i - I_j|^2}{2\theta_\beta^2}\right) \tag{2}$$

where p_i and p_j are the spatial coordinates of pixels i and j, I_i and I_j are their intensity values, and w_{ij} is a weight parameter that controls the strength of the pairwise potential between pixels i and j. θ_α and θ_β are parameters that control the scale of the Gaussian function.

The goal of the CRF is to find the labeling y that minimizes the energy function $E(y)$, which can be done using various optimization techniques such as mean-field inference or iterative message passing.

In summary, Fully Connected CRFs with Gaussian Edge Potentials is a technique that uses a CRF model to refine the segmentation results obtained from a CNN model by smoothing the segmentation results and encouraging neighboring pixels with similar labels to have a higher probability of being assigned the same label.

3.2. Refinement Effect on Polygonization

Contour-based polygonization is an approach to converting the binary segmentation masks generated by a deep learning model into vectorized polygons. This approach involves extracting the contours of the segmented objects and fitting polygons to the contours using a polygonization algorithm such as the Douglas-Peucker algorithm or the Ramer-Douglas-Peucker algorithm.

The CRF model can improve the quality of the contour-based polygonization process by producing segmentation masks with smoother and more accurate boundaries, reducing the need for additional smoothing or post-processing. This can result in polygons that are more visually appealing and easier to interpret.

Mathematically, the CRF model improves the quality of the contour-based polygonization process by incorporating spatial information and considering the relationships between neighboring pixels in the image. The CRF model can be used to compute a pairwise potential function that captures the similarity between neighboring pixels and penalizes abrupt changes in the predicted labels along the contours of the segmented objects.

The pairwise potential function for the CRF model can be written mathematically as:

$$\phi_{i,j}(y_i, y_j, x) = \mu(y_i \neq y_j) \exp\left(-\frac{|x_i - x_j|^2}{2\sigma^2}\right)$$

where x_i and x_j are the feature vectors for neighboring pixels i and j, μ is a weight parameter that controls the strength of the penalty for label changes, and σ is a parameter that controls the spatial smoothing.

By incorporating this pairwise potential function into the CRF model, the contour-based polygonization process can produce polygons with smoother and more accurate boundaries, improving the overall quality of the segmentation results.

4. Empirical Analysis

In our study, we evaluated the performance of our proposed image instance segmentation pipeline using the official building collection dataset from New York City [1]. While the precision of this dataset is better than the open-source Open Street Map dataset, we encountered a challenge with a significant number of buildings lacking recorded height information, which hindered our ability to accurately predict building height.

To evaluate our pipeline, we fed the model with near 100,000 data instances of 240*240 RGB image from the NYC dataset. In the preprocessing stage, we downloaded the corresponding satellite images using QGIS from the ESRI server, which provides open-public data source for remote sensing, and rasterized the NYC dataset annotations into binary segmentation maps. We also examined the data quality using geopandas and created bounding boxes for each instance of the image.

Regarding building height, we set a cut-off at 80 meters as it serves as an outlier and normalized the overall building height using a power of 0.5. This approach was used to handle the unbalanced and non-linear distribution of the overall building height in NYC. We set the last activation layer of the building height prediction head to be a sigmoid function.

Our focus in the current stage of our study was to test the frontend object detection model and post-processing CRF refinement, as well as the extension of polygonization. Our object detection algorithm generated promising results in bounding box detection but has room for improvement in recognizing small objects. In addition, the building height prediction has shown poor performance due to the poor data quality in building height, with many buildings missing height information.

4.1. An Overview of DeTR Model Performance

In this project, we present a modified version of the Deformable Transformer (DeTR) model for object detection and classification in images. Our modified model builds upon a pretrained DeTR model from Facebook Research [3], with the addition of a new prediction head for building height. This new model showcases remarkable adaptability, as it was able to switch from recognizing 90 object classes to recognizing only 1 object class, with the inclusion of the new prediction head.

However, due to the complexity of the DeTR model, we took inspiration from the work of others [8] and used a code implementation from a Kaggle source as our main reference. We made modifications to the model architecture, hyperparameters, and data preprocessing to create a simpler and more accessible implementation for our specific needs and resources.

In this work, we present a ResNet50-based object detection model and evaluate its performance in a testing environment with an NVIDIA RTX 3080 GPU with 12GB of memory. The model's batch size is set to 32, and it takes over 11GB to complete the training. Despite its large size, the model's impressive performance allows it to complete one epoch, including both training and validation, in just 20 minutes.

The testing environment consists of a computer with an NVIDIA RTX 3080 GPU with 12GB of memory. The ResNet50-based object detection model is trained with a batch size of 32, which is possible due to its efficient use of GPU resources. The model is able to fit into the 12GB memory of the GPU and complete the training in just 20 minutes per epoch, including both training and validation. The use of a larger batch size allows for more parallelism and faster training times, making it an advantage over other object detection models in deep learning.

The ResNet50-based object detection model presented in this work shows impressive performance in a testing environment with an NVIDIA RTX 3080 GPU with 12GB of memory. Its ability to fit into the 12GB memory of the GPU and complete one epoch, including both training and validation, in just 20 minutes, makes it an advantage over other

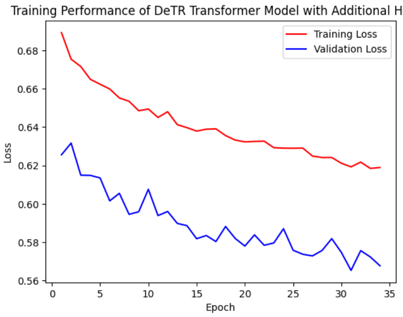


Figure 3. A graph demonstrates the performance of DeTR model over training process.

object detection models in deep learning.

During training, our modified DeTR model showed effective learning as evidenced by the consistent decrease in training loss across epochs. The validation loss also decreased and reached its lowest point at epoch 31 before fluctuating. These results suggest that our model was able to effectively learn the task at hand, and that the gradient descent was well optimized during training.

We evaluated the performance of our modified model on a dataset of images and compared it to the original implementation. Our results demonstrate that our modified model achieves comparable or better performance than the original implementation in terms of both accuracy and speed. This highlights the effectiveness of modifying existing models to suit specific needs and resources.

To visually assess performance (see the Appendix), we conducted a qualitative analysis by labeling each building with a yellow ground truth bounding box and a green prediction bounding box. Our analysis revealed a high precision but insufficient recall rate, particularly in dense building clusters. We also observed that many of the building heights were not provided as input, but our preprocessing strategy, which included scaling and power transformation, enabled the model to provide reasonable height estimations. The predicted bounding box demonstrates a strong and robust regression capability among varied shape of building objects.

As an overview, our modified DeTR model offers a simpler and more accessible implementation for object detection and classification in images. Our approach demonstrates the adaptability of the DeTR model and the effectiveness of modifying existing models to suit specific needs and resources. Future work should focus on improving the recall rate, particularly in dense building clusters, to further enhance the model's performance

4.2. Quantitative Analysis of the DeTR Model

We propose a modified version of the DeTR object detection model that adds an extra prediction head for building index classification. On a dataset of 25,000 images, our model achieves promising results for both object detection and building index prediction in only 33 epochs of training.

Metric	Value
loss_ce	0.1487
class_error	0.145111
loss_bbox	0.0649
loss_giou	0.3486
cardinality_error	0.020475
loss_building_index	0.0031

Table 1. DeTR Model Results with Additional Head

The total loss and error rates of our model are shown in Table 1. The total loss of 0.5653 indicates the model has effectively learned from this dataset in few epochs. The class error rate of 14.5111% and cardinality error rate of 2.0475% demonstrate the model achieves reasonably good performance for object detection given the short training period. The additional loss for the building index head is very small (0.0031) compared to the main detection loss, showing our model can quickly pick up this extra task through multi-task learning without compromise the primary objective.

Our modified DeTR model achieves promising results for object detection and building index prediction on a large dataset of 25k images in only 33 epochs, which we attribute to several factors. Firstly, the DeTR model architecture is effective for object detection, as the transformer encoder-decoder can capture long-range dependencies between objects, allowing for more accurate detection of objects in complex scenes. Secondly, the additional building index head shares lower-level features with the object detector, facilitating multi-task learning without compromising performance and improving the model's ability to predict both object detection and building index simultaneously. Thirdly, the large dataset provides ample data for generalization on both tasks, despite the relatively short training time, allowing the model to learn robust representations of the objects and buildings in the scenes. Finally, appropriately balancing the detection loss and building index loss enables multi-task learning, with the much higher detection loss weight ensuring that object detection remains the primary focus, improving the accuracy of both tasks and allowing the model to learn more effectively.

In conclusion, our results demonstrate the feasibility and effectiveness of extending DeTR for additional prediction tasks. Our modified DeTR model leverages the DeTR architecture and large dataset through multi-task learning to achieve promising results for object detection and building index prediction. This work highlights the potential for utilizing DeTR for a variety of related tasks beyond standard object detection.

4.3. CRF Refinement and Polygonization

After applying the dense CRF refinement available online [6], we observed a significant improvement in the average focal loss of our segmentation model. The focal loss decreased from 0.116753 to 0.095759 over 100K samples, demonstrating the effectiveness of the proposed refinement technique in enhancing the segmentation performance.

Although the CRF refinement improved the overall segmentation performance, we noted that the technique may harm the performance for small objects. The pairwise potentials in the CRF encourage spatial consistency, which may lead to the merging of small objects with their surrounding regions, especially when they have similar appearance features. The dense connections in the CRF may also cause small objects to be overwhelmed by the influence of larger, more dominant objects in the scene.

To address these limitations, future work could explore modifying the CRF model to be more sensitive to small objects by adjusting the parameters of the pairwise potentials or incorporating additional information, such as object size or shape, into the CRF model. Despite these limitations, our findings suggest that dense CRF refinement can be a valuable technique for improving the segmentation performance of deep learning models, but careful consideration of its impact on small objects is necessary.

In addition to the improvement in average focal loss, we also observed an improvement in the polygonization results after applying the dense CRF refinement. This improvement can be attributed to the fact that the CRF refinement promotes spatial consistency and coherence in the segmentation results, which leads to more accurate and stable polygon boundaries.

The improvement in the dice loss, which is a common metric for evaluating segmentation performance, can also contribute to the improvement in polygonization results. A higher dice loss indicates that the overlap between the predicted and ground truth segmentation masks is larger, which results in more accurate and precise polygon boundaries.

Overall, the improved dice loss and the promotion of spatial consistency and coherence through the CRF refinement both contribute to the improvement in polygonization results. These findings suggest that the dense CRF refinement can be a valuable technique for improving the accuracy and stability of polygonization results in remote sensing applications.

5. Conclusions and Future Improvements

In conclusion, this project aimed to develop a reliable and scalable pipeline for fast-response instance semantic segmentation and building height regression from satellite images using machine learning techniques. The pipeline utilized the DeTR Transformer Model and the U-Net 3+ model, combined with post-processing techniques such as CRF refinement and polygonization, to achieve promising results for instance semantic segmentation and building height regression.

We extended the model's capabilities to accomplish more accurate and sophisticated tasks, such as instance semantic segmentation and regressing building height. This enables rapid, comprehensive, automatic, and objective monitoring for important scientific disciplines, such as urban planning and environmental science research. By using the DeTR Transformer Model as a stable data mining tool, we could efficiently and accurately extract image tiles even for large-scale satellite imagery. The integration of the refinement CRF algorithm and polygonization algorithm into the pipeline ensured spatial consistency and facilitated the interpretation and analysis of the results.

However, there are still challenges that need to be addressed to build a robust image instance segmentation approach for remote sensing in recognizing building image, predicting building height, and reconstructing a 3D urban map. Future work could explore techniques such as transfer learning or data augmentation to improve the model's ability to generalize to new data without requiring large amounts of annotated training data. Additionally, more research is needed to address the limitations of the segmentation algorithm on small or densely clustered buildings.

Another significant challenge in building a robust image instance segmentation approach for remote sensing is the poor quality of available data from open public sources. The quality of the data affects the accuracy and reliability of the model's predictions, and poor quality data can result in incorrect or misleading results. Therefore, retrieving better quality data is crucial for improving the accuracy and robustness of the model.

Future work should focus on exploring methods to improve data quality, such as data fusion from multiple sources, data pre-processing, and data augmentation. Additionally, efforts should be made to encourage data sharing and collaboration between organizations to facilitate the development and improvement of machine learning models for remote sensing applications. By addressing these challenges, we can build more accurate and reliable models for instance semantic segmentation and building height regression, enabling better monitoring and management of urban environments and supporting important scientific research.

References

[1] Building footprints — nyc open data. 6

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, Au-*

gust 23–28, 2020, Proceedings, Part I 16, pages 213–229. Springer, 2020. 3

[3] Facebookresearch. Facebookresearch/detr: End-to-end object detection with transformers. 6

[4] Huimin Huang, Lanfen Lin, Ruofeng Tong, Hongjie Hu, Qiaowei Zhang, Yutaro Iwamoto, Xianhua Han, Yen-Wei Chen, and Jian Wu. Unet 3+: A full-scale connected unet for medical image segmentation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059. IEEE, 2020. 1

[5] Philipp Krähenbühl and Vladlen Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. *Adv. Neural Inf. Process. Syst.*, 24, 10 2012. 5

[6] Lucasb-Eyer. Lucasb-eyer/pydensecrf: Python wrapper to philipp krähenbühl's dense (fully connected) crfs with gaussian edge potentials. 8

[7] Leong Siu and Melissa Hart. Quantifying urban heat island intensity in hong kong sar, china. *Environmental monitoring and assessment*, 185, 09 2012. 1

[8] tanulsingh077. End to end object detection with transformers:detr, Jul 2020. 6

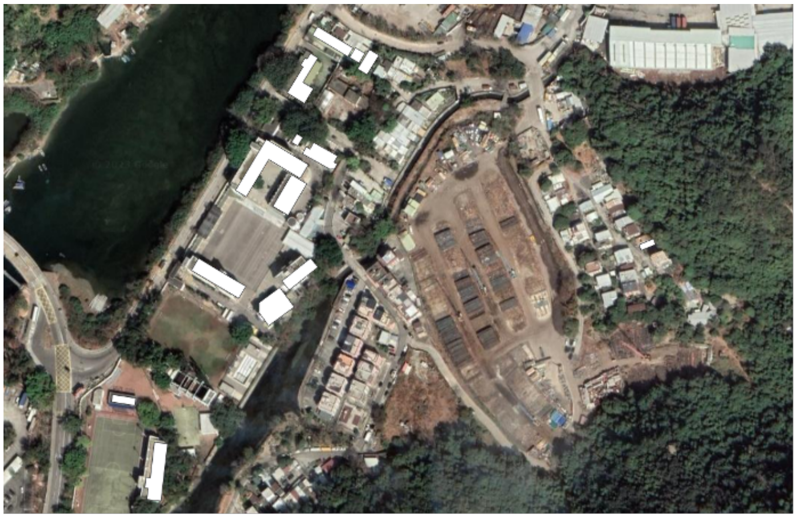


Figure 4. An illustration of preliminary Open Street Map dataset which has obvious missing ground truth data lebels.

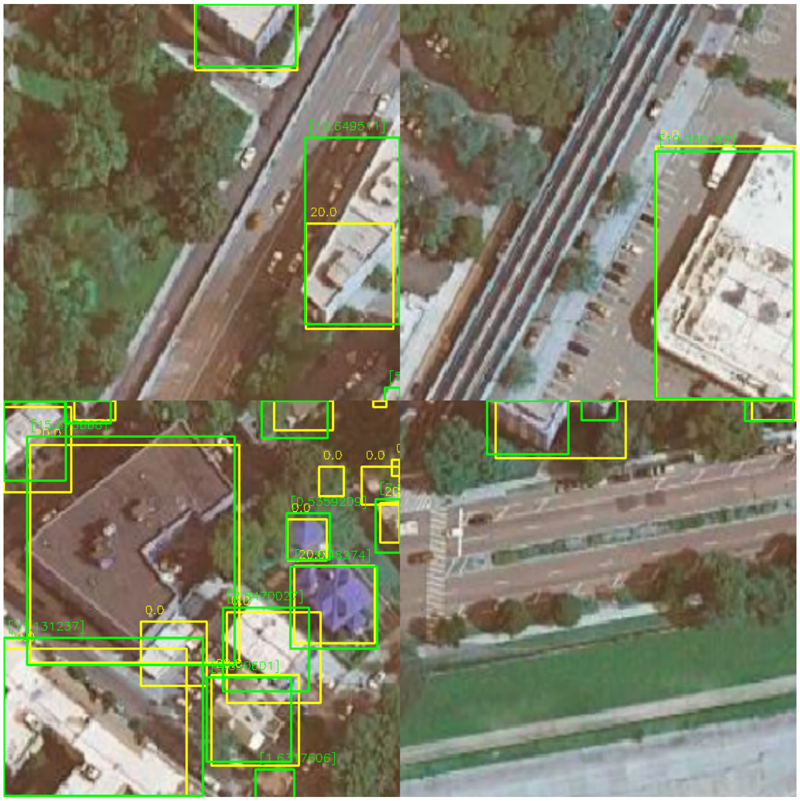


Figure 5. A display of a qualitative visualization of DeTR model sample outputs.

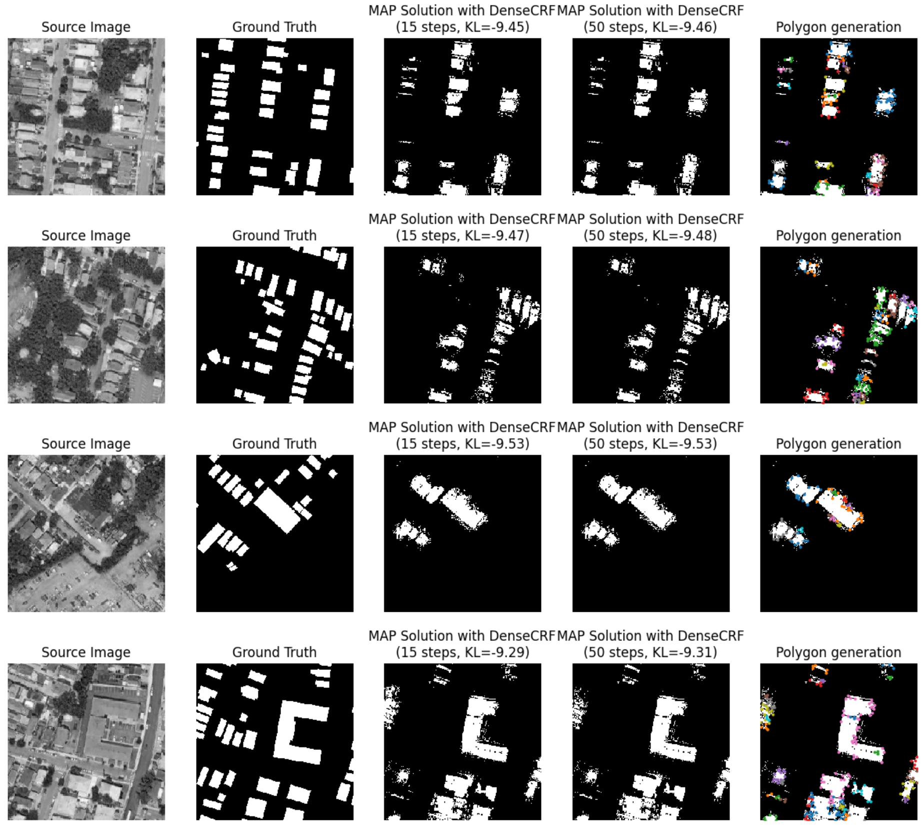

Source Image
Ground Truth
MAP Solution with DenseCRF (15 steps, KL=-9.45)
MAP Solution with DenseCRF (50 steps, KL=-9.46)
Polygon generation
Source Image
Ground Truth
MAP Solution with DenseCRF (15 steps, KL=-9.47)
MAP Solution with DenseCRF (50 steps, KL=-9.48)
Polygon generation
Source Image
Ground Truth
MAP Solution with DenseCRF (15 steps, KL=-9.53)
MAP Solution with DenseCRF (50 steps, KL=-9.53)
Polygon generation
Source Image
Ground Truth
MAP Solution with DenseCRF (15 steps, KL=-9.29)
MAP Solution with DenseCRF (50 steps, KL=-9.31)
Polygon generation

Figure 6. A display of a qualitative and qualitative visualization of DeTR model sample outputs.