UNIVERSITY OF SCIENCES AND TECHNOLOGIES OF HANOI

ICT DEPARTMENT

Introduction to Deep Learning

Group Project Report

Study the Vision Transformer for Remote Sensing Image Segmentation

Group members:

Giáp Đỗ Anh Minh - 22BI13282 - ICT

Nguyễn Gia Bách - 22BI13052 - ICT

Nguyễn Đăng Nguyên - 22BI13340 - ICT

Phùng Tiến Đạt - 22BI13081 - ICT

Nguyễn Quốc Khánh - 22BI13211 - ICT

Nguyễn Thị Hà Anh - 22BI13029 - ICT

Table of Contents

Abstract

For years, CNNs (Convolutional Neural Networks) have been the go-to choice for image segmentation. Models like U-Net and Mask R-CNN captured the details that are needed to distinguish different objects in an image, making them state-of-the-art for segmentation tasks. However, with the introduction of Vision Transformer, it has achieved excellent results compared to state-of-the-art convolutional networks while requiring fewer computational resources to train when pre-trained on large amounts of data and transferred to multiple mid-sized or small image recognition benchmarks.

I, Introduction

1, Overview

This report aims to study and analyze the Vision Transformer (ViT) model, with a focus on its structure and functionality. The study delves into the fundamental concepts of the ViT, explaining what it is and how it operates. The report seeks to provide readers with a clear understanding of the essential components of this model, while also addressing key insights and significant aspects of the Vision Transformer.

We also introduce “Building segmentation for urban planning” as our case study. This field image segmentation task is important in identifying and classifying the buildings from satellite images. By using a Vision Transformer, we have automated the process of detecting features which help to segment buildings and not building. This assists planners in making the right decisions based on accurate data.

2, Vision Transformer

Transformer model is used widely on sequences, it’s used a lot in Natural Language Processing models [1]. The same concept has been applied to images through the paper “An image is worth 16x16 words” [2] where each image is treated as a sequence (patch).

The input images are split into fixed-size patches and then are flattened. Each of the patches is linearly embedded by projecting it into a higher-dimensional space. Unlike with classification, we do not need to prepend class tokens. A position embedding is added to each patch

embedding to retain spatial information. The resulting embedding will be fed into the series of Transformer Encoders, each containing multi-head self-attention mechanisms (which are based on attention mechanisms [3]) and feed-forward neural networks. Finally, a segmentation head reconstructs the pixel-level segmentation.

3, Remote Sensing Image Segmentation

Remote sensing refers to the process of collecting and analyzing data about the Earth's surface from a distance, typically using satellites, drones, or aircraft. This technology is widely used in fields like environmental monitoring, urban planning, agriculture, and disaster management. One of the most critical tasks in remote sensing is image segmentation, which involves dividing a remote sensing image into multiple meaningful segments or regions based on specific criteria, such as land cover, vegetation, water bodies, or urban areas, which can represent different land cover types, objects, or features in the environment.

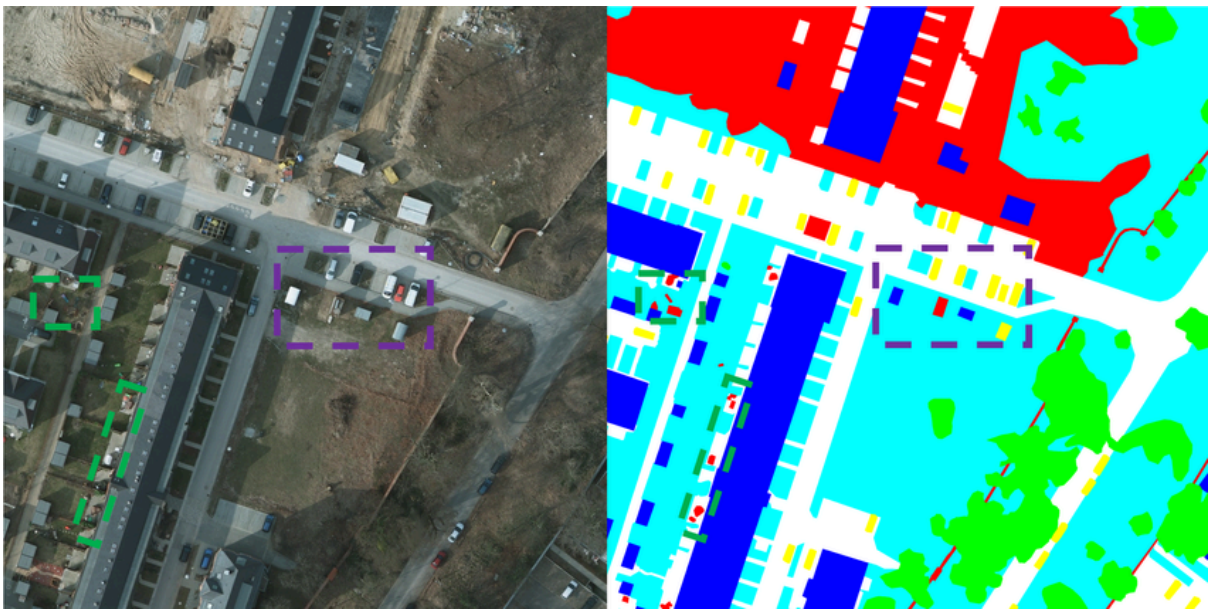

Figure 1.3: An example of remote sensing image segmentation

II, Methodology

1, Dataset

We used the INRIA Aerial Image Labeling dataset [4] for training. The dataset consists of 360 RGB tiles of 5000×5000px of 10 cities across the globe. Half of the cities are used for training and are associated with a public ground truth of building footprints (masks). The rest of the dataset is used only for evaluation with a hidden ground truth.

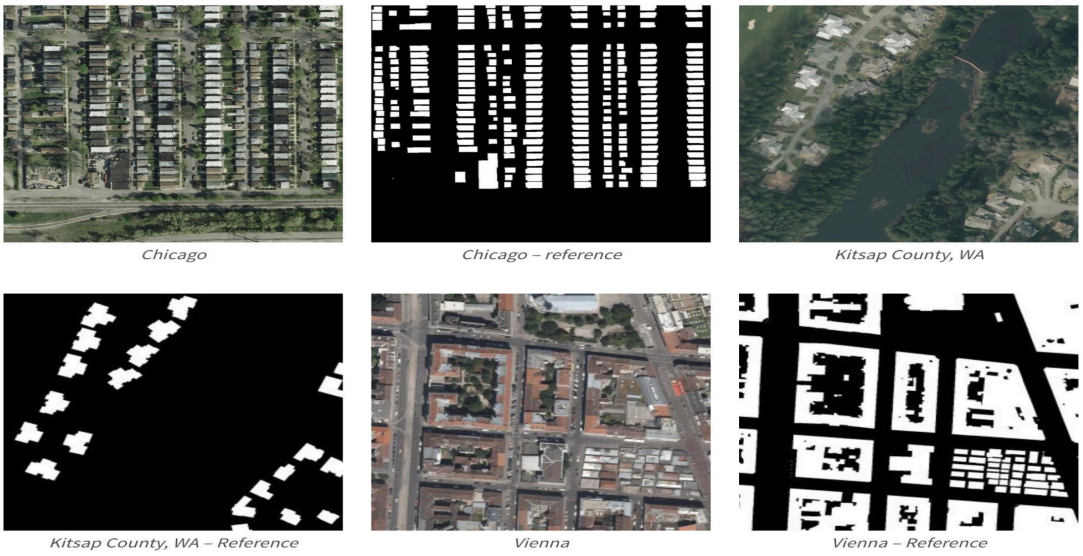

Figure 2.1: Example of INRIA Aerial Image Labeling dataset

2, Data Preprocessing

As the dataset was previously divided into training set and test set, we can directly preprocess the training set without division. For the training set, we started by calculating the mean and standard deviation of the training images across all channels (R, G, B). We then transformed the training images including resizing to size 224x224 pixels, normalizing using the calculated mean and standard deviation, converting images to tensors. For the masks of training images, we converted them to grayscale for better visualization, resized them 224x224 pixels and converted them to tensors. The testing images were also resized, normalized and converted to tensors to match the model's required format.

3, Model architecture

We applied the Vision Transformation (ViT) model for the image segmentation task based on the architecture of Vision Transformer ViT-base patch 16 224 [2] but we made some adjustments to make it suitable for our image segmentation task.

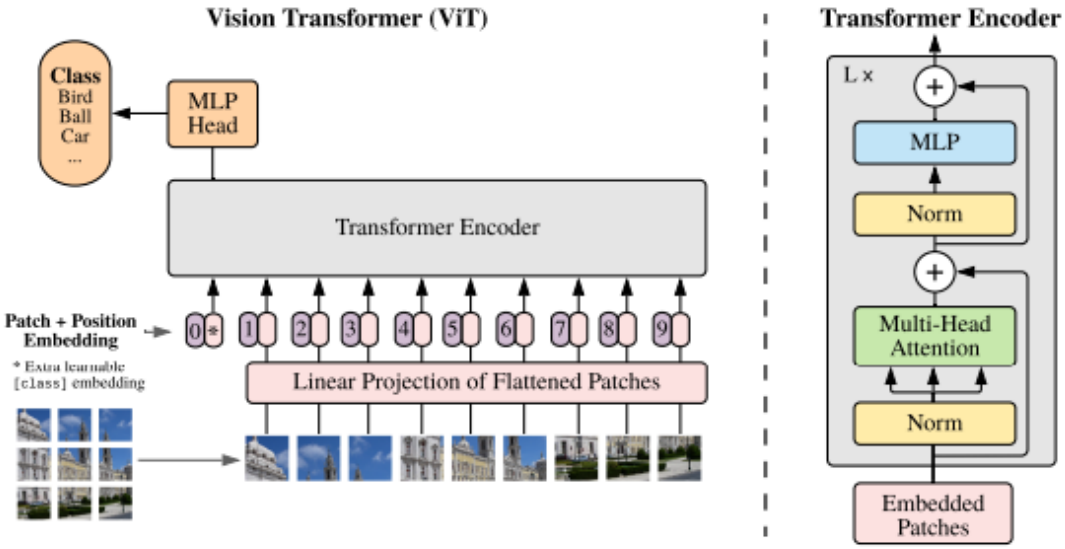

Figure 2.3: Vision transformer architecture

Our Vision Transformer model contains three main components:

a) Input embedding class:

This class prepares the input images by dividing it into small patches. Each patch is flattened and transformed into a vector form using a linear layer. This transformation is necessary for the model to interpret the embeddings which are transformed from image patches. For more spatial information, a positional embedding is applied to each patch, helping the model to recognize the location of each patch in the original image. The results will be sent to the transformers encoders to continue the process.

The transformation is done by the following formula:

$$x(input) \in R^{(H*W*C)} \rightarrow x(sequence) \in R^{(N*(P^2*C))}$$

(H: height, W: weight): resolution of the original image

C: Number of channels

N: Number of patches

(P, P): resolution of the patch

b) Transformer Encoder class:

This class will handle the main processing of the patch embeddings. It consists of 12 encoder blocks, each blocks include:

- **Layer Normalization:** Stabilizes and improves the training process by normalizing the input from the previous step.
- **Multi-Head Self-Attention:** Allows the model to understand the relationships between patches in the image.
- **Residual Connection:** A skip connection is added around the self-attention and MLP layers, which ensures the original input is preserved while adding more learning features.
- **Multi-Layer Perceptron (MLP):** Further processes from the Attention mechanism to create more refined representations.

Those stacks of encoder layers will refine the patch embeddings, making them to be more informative, preparing for the segmentation task.

c) Vision Transformer class:

This class will combine the Input Embedding and the Stack of Encoder Layers. The embedded patches are processed by a series of encoder layers to extract complex features. The output from the encoders is reshaped and fed into the segmentation head, which acts as an MLP. This head applies Convolutional layers and Upsampling to produce a segmentation map, predicting the class of each pixel in the original image. This MLP head is important in transforming the learned features into pixel-wise classifications.

4, Model training

The training of our model includes the following steps:

- **Initializing the built model**: Make an object of the model we just built.
- **Loss function**: BCEWithLogitsLoss [5] was selected as the loss function. As our case study is to segment between two categories: building and non-building, this loss function is more suitable [6] than Cross Entropy Loss because it is designed for binary

segmentation tasks, whereas Cross Entropy Loss is for multiclass segmentation.

- **Optimizer**: We used AdamW optimization with learning rate 0.0001, weight decay of 0.0001.
- **Scheduler**: We used ReduceLROnPlateau as the scheduler. This scheduler will reduce the learning rate when the IoU metric has stopped improving.
- **Iterative training**: We trained the model with 40 epochs, at each epoch we will compute the loss and IoU score and adjust the learning rate with the scheduler.

5, Model evaluation

For evaluating the model, we used Intersection over Union (IoU) metric [7]. We customized IoU a little to match with our segmentation task. The metric is calculated by firstly applying the sigmoid function to the raw output values of the model to convert them to probabilities, which are calculated as this formula:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

A threshold of 0.5 is applied to convert probabilities to binary predictions. Both the predicted binary mask and the ground truth mask are flattened into one-dimensional tensors. The intersection is calculated by finding the element-wise product of the flattened predictions and targets. The union is computed by summing the counts of predictions and targets, then subtracting the intersection. The final IoU metric is calculated by the division between intersection and union.

III, Implementation

1, Training set up

We ran the training on Google Colab using T4 GPU, which took nearly 2 hours to complete.

2, Result and Discussion

The result was very poor, we tried running with different epochs but the IoU metric did not improve above 0.16 mark.

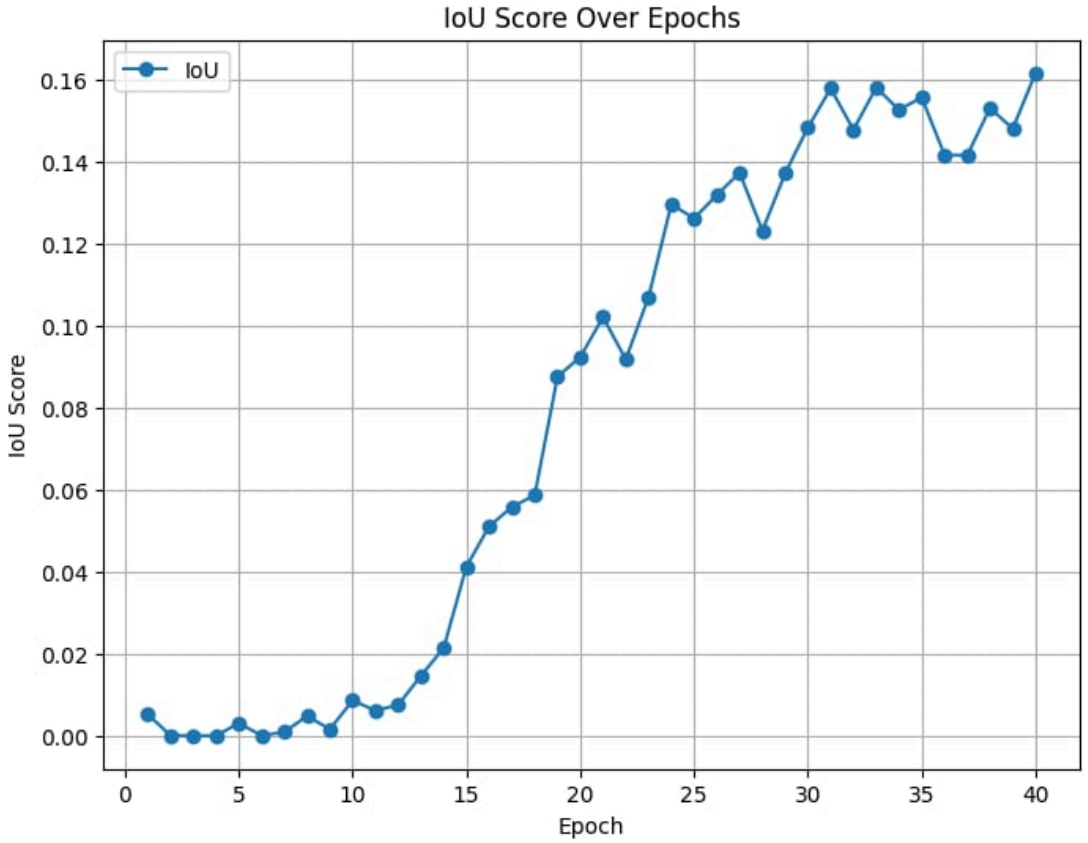

Figure 3.2.1: IoU Score throughout the epochs

We also let the model predict on the test set images and showed some sample results and they were not very good. Overall, the predicted masks seem blurry, not detailed enough, however, on the bright side, it could roughly predict the black part to be the non-building area and white part to be the building area.

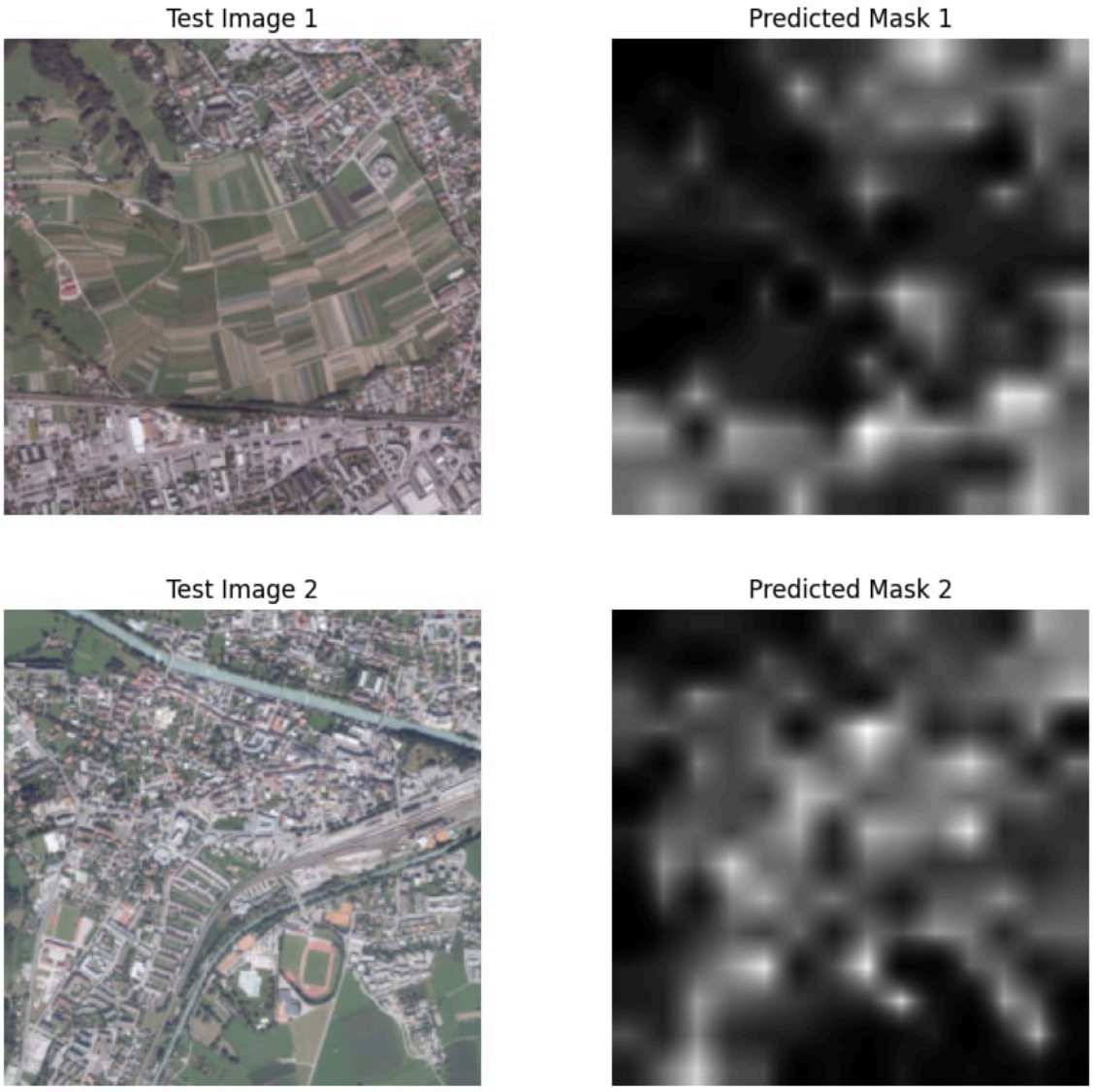

Figure 3.2.2: Some example outputs predicted by the model

Based on the result, we think the model performed not very well on this building segmentation for urban planning tasks. This is totally understandable since the Vision Transformer is ideally to be trained in a larger dataset with millions of images then fine tune to smaller dataset. Our dataset only contains 180 images with 180 masks to train, so it is definitely challenging for the model to learn.

In addition, we also tried to use a Vision Transformer pre-trained model on a larger building segmentation dataset but we could not find any. Instead, we tried a pre-trained model SegFormer [8], which was pre-trained on general segmentation dataset ADE20k and fine tuned to our dataset. We used the mit-b0 weight of SegFormer and the IoU score did improve to about 0.23 but it was still poor.

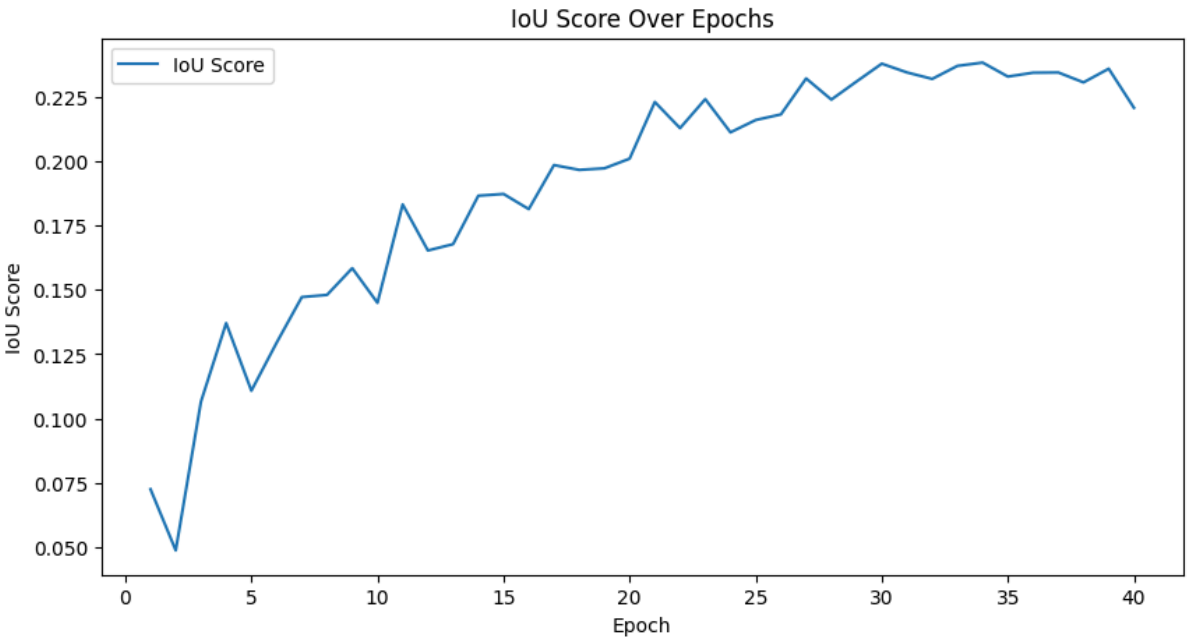

Figure 3.2.3: The IoU score of the model with SegFormer pre-trained weight

IV, Conclusion

1, What we have learned

In this project, we tried to apply a Vision Transformer (ViT) model for building segmentation from aerial images for urban planning. By building the model from scratch, we have learned the model architecture including the self-attention mechanisms, embedding techniques, linear projections. We also learned how to preprocess, convert the input image into the required format by the Vision Transformer.

Vision Transformer can be very powerful compared to traditional Convolutional Neural Network, especially in classification tasks due to its self-attention mechanisms. Despite the strong performance, Vision Transformer would require a large amount of data to learn, which is less ideal for our building segmentation task where the amount of data is limited.

2, Future work

The Vision Transformer (ViT) represents a significant improvement when applied to computer vision tasks. In the context of an increasingly large and diverse amount of data, we hope to have more large data available for training the Vision Transformer for building segmentation tasks. We will continue to improve our model which includes optimizing further, checking for overfitting, and using larger data to train our model if available. In spite of the poor result, we believe with enough data to learn, the model would become very powerful in correctly segmenting between building and non-building areas and could contribute to urban planning strategies for the governments.

V, References

[1] Transformers in NLP: A beginner friendly explanation - Towards Data Science

[2] Dosovitskiy, Alexey. "An image is worth 16x16 words: Transformers for image recognition at scale." *arXiv preprint arXiv:2010.11929* (2020).

[3] Vaswani, A. "Attention is all you need." *Advances in Neural Information Processing Systems* (2017).

[4] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat and Pierre Alliez. “Can Semantic Labeling Methods Generalize to Any City? The Inria Aerial Image Labeling Benchmark”. IEEE International Geoscience and Remote Sensing Symposium (IGARSS). 2017.

[5] BCEWithLogitsLoss — PyTorch 2.4 documentation

[6] Should i use nn.BCEWithLogitsLoss() or Cross Entropy loss for segmentation - PyTorch discussion

[7] Intersection over Union guide - v7labs

[8] Xie, Enze, et al. "SegFormer: Simple and efficient design for semantic segmentation with transformers." *Advances in neural information processing systems* 34 (2021): 12077-12090.