Road Semantic Segmentation: Between CNNs and Vision Transformers

Dareen Deeb	Rand Ahed	Aseel Sharsheer
0205291	0201194	0203882

Supervised by: Dr. Tamam Al Sarhan

University Of Jordan, AI department, Semester: 2024-2025

Abstract

This project explores the differences between convolutional neural network (CNN)-based methods, such as ConvNeXt and C-UNet, and transformer-based architectures, particularly SegFormer, for the task of semantic road segmentation in high-resolution satellite imagery. Using the EPFL Machine Learning Road Segmentation dataset, we compare the performance of these models in accurately distinguishing road networks from background features. Our findings show that while CNN-based models effectively leverage multi-scale feature extraction, transformer-based models like SegFormer consistently outperform them by capturing long-range dependencies and multi-level contextual information. The results highlight SegFormer' s superior accuracy and efficiency, demonstrating the transformative potential of lightweight vision transformers in remote sensing and geospatial analysis tasks.

1. Introduction

Roads are among the most critical infrastructures supporting human society and serve as fundamental components of geospatial information systems. They play pivotal roles in various domains, including urban planning, disaster management, agricultural development, and intelligent transportation systems. Extracting accurate and up-to-date road information from high-resolution remote sensing imagery is therefore essential for maintaining and enhancing geospatial databases.

Automated road extraction from remotely sensed images has become an indispensable task in numerous remote sensing applications, such as intelligent transportation management (Guerrero-Ibañez et al., 2021), image registration (Tondewad and Dale, 2020), and topographic database updating (Mena, 2003). Moreover, the precision of road extraction directly influences the detection and analysis of other critical objects, including vehicles (Abraham & Sasikumar, 2013), buildings (Simler, 2011), and oil well pads (He et al., 2022). As a result, developing robust automated road segmentation methods is of broad interest to the remote sensing and geospatial research communities.

Deep learning (DL)-based road segmentation approaches typically frame the task as a binary pixel-wise classification problem, where each pixel in a remote sensing image is classified as either road (foreground) or non-road (background). These models commonly employ an encoder–decoder architecture: the encoder progressively downsamples the input image through convolutional operations to extract high-level feature representations, while the decoder reconstructs a dense, pixel-level segmentation map by gradually upsampling the encoded features. Each pixel in the output segmentation mask is thus assigned to a semantic class, allowing for precise delineation of road networks.

Training such DL-based road segmentation models is generally conducted in a supervised manner, where the network learns to predict accurate segmentation maps based on labeled training data (ground truth). The effectiveness of this learning process depends heavily on the design of the loss function, which guides the model in adjusting its parameters to minimize prediction errors on unseen data.

Road extraction is a specialized application within the broader field of semantic segmentation, and recent research has placed strong emphasis on improving both network architectures and feature extraction techniques. This has led to the development of innovative models that achieve superior performance in terms of accuracy, efficiency, and generalization.

In this work, we propose using SegFormer, a Vision Transformer-based architecture, for road segmentation tasks and compare its performance against state-of-the-art convolutional methods, specifically ConvNeXt and C-UNet. We employed the SegFormer-B2 variant, which features a hierarchical Transformer encoder paired with a lightweight multilayer perceptron (MLP) decoder. This architecture enables multiscale feature representation without

requiring positional encodings, allowing the model to generalize effectively across varying image resolutions and scene complexities.

We trained the SegFormer-B2 model using the EPFL Machine Learning Road Segmentation dataset, which provides high-resolution satellite imagery focused on road networks. Our results demonstrate that SegFormer-B2 achieves robust segmentation performance, outperforming competing models in terms of accuracy, model size, and runtime on both the EPFL dataset and an additional publicly available benchmark. Furthermore, we conducted inference experiments on satellite imagery of road networks in Jordan, obtained from Google Earth, where the model maintained strong performance, showcasing its practical applicability across diverse geographic regions.

2. Related Work

2.1 CNN-Based Methods

Road segmentation in remote sensing is typically framed as a binary semantic segmentation task, where the goal is to extract road regions from high-resolution aerial or satellite images. Early methods relied heavily on convolutional neural networks (CNNs) due to their strong local feature extraction capabilities and robust generalization.

A foundational architecture in this space is U-Net (Ronneberger et al., 2015), which uses an encoder–decoder structure with skip connections to combine low-level details with high-level semantic features, achieving excellent performance in segmentation tasks. Subsequent models like DeepLab (Chen et al., 2018) further enhanced segmentation performance by introducing dilated convolutions and multi-scale context modules**,** expanding the receptive field without sacrificing resolution.

However, classical CNN architectures like U-Net often struggle with road extraction in cases where the roads vary greatly in width, shape, and connectivity. To address these challenges, C-UNet (Hou et al., 2021) was introduced as a complementary network combining a standard U-Net with a multi-scale dilated convolution U-Net (MD-UNet). This two-stage approach first segments easy-to-detect road areas and then refines the segmentation by focusing on more difficult or finer structures, improving accuracy on complex road networks.

More recently, ConvNeXt, introduced by Liu et al. in the paper *"A ConvNet for the 2020s"*, has emerged as a modern CNN architecture inspired by Vision Transformers. By incorporating architectural innovations such as larger convolutional kernels, inverted bottlenecks, GELU activations, and LayerNorm, ConvNeXt significantly improves the representational power of convolutional networks while maintaining their computational efficiency. Building on this backbone, Wang et al. (2024) proposed a ConvNeXt-UperNet-based deep learning model for semantic road extraction from high-resolution remote sensing images, demonstrating how integrating ConvNeXt with advanced decoder frameworks like UperNet can enhance performance on dense prediction tasks. Motivated by these innovations, we explored both C-UNet and ConvNeXt in our experiments to combine multi-scale feature extraction with enhanced backbone design.

2.2 Transformer-Based Methods

While CNN-based methods have historically dominated image segmentation, Transformer-based architectures have recently revolutionized the field by introducing the ability to capture long-range dependencies and global context.

The Vision Transformer (ViT) (Dosovitskiy et al., 2020) was the first to show that a pure Transformer architecture could achieve state-of-the-art performance in image classification by treating image patches as tokens in a sequence. Following this, models like DeiT (Touvron et al., 2021) introduced data-efficient training strategies, while others such as T2T-ViT (Yuan et al., 2021), CPVT (Chu et al., 2021), TNT (Han et al., 2021), CrossViT (Chen et al., 2021), and LocalViT (Li et al., 2021) introduced tailored modifications to further improve classification performance.

A striking recent finding is that ViTs are inherently good at segmentation**,** even without explicit supervision for dense prediction. The work *"Your ViT is Secretly an Image Segmentation Model"* (Zhou et al., 2025) demonstrates that ViTs naturally induce patch-level grouping aligned with object boundaries, making them excellent candidates for downstream segmentation tasks. This insight helps explain why Transformer-based models have rapidly gained traction in segmentation applications.

To extend Transformers beyond classification**,** Pyramid Vision Transformer (PVT) (Wang et al., 2021) introduced a pyramid structure, demonstrating the

potential of Transformers to replace CNN backbones in dense prediction tasks. This line of work was further advanced by models like Swin Transformer (Liu et al., 2021), CvT (Wu et al., 2021), CoaT (Xu et al., 2021**),** LeViT (Graham et al., 2021), and Twin**s** (Chu et al., 2021), which focused on improving local feature continuity and removing fixed-size positional embeddings.

Among these, SegFormer (Xie et al., 2021) emerged as a lightweight yet highly effective Transformer architecture specifically designed for semantic segmentation. In our experiments, while both C-UNet and ConvNeXt showed promising results, they did not outperform Transformer-based models. Notably, SegFormer consistently delivered superior accuracy and efficiency, confirming the strength of Transformer-based designs in capturing global context and multi-scale information critical for road segmentation.

3. Methods

In this section, we present the models applied to our road semantic segmentation dataset, which has not been used in prior SegFormer or Convnext publications. We compare the performance of three architectures: SegFormer (*Xie et al.*, 2021)), ConvNeXt-UPerNet (*Wang et al.*, 2024), C-UNet (*Hou et al.*, 2021)

3.1 SegFormer

SegFormer is a transformer-based model specifically designed for semantic segmentation, with a focus on efficiency and accuracy. The model consists of two main components:

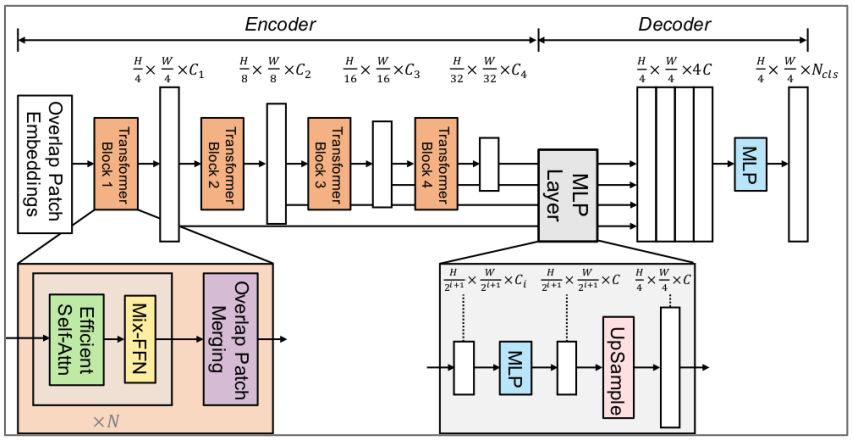

Figure 1: The SegFormer framework includes a hierarchical Transformer encoder for extracting multiscale features and a lightweight All-MLP decoder that fuses these features to predict the semantic segmentation mask. “FFN” refers to the feed-forward network

1. Hierarchical Transformer Encoder

The encoder uses a Mix of Transformers (MiT) backbone available in several scales (MiT-B0 to MiT-B5) all with the same architectural blocks but varying depths and widths. The encoder's goal is to transform the input image H $\times W \times 3$ into multi-scale feature maps at progressively coarser resolutions:

$$F_1: \frac{H}{4} \times \frac{W}{4} \times C_1, \quad F_2: \frac{H}{8} \times \frac{W}{8} \times C_2, \quad F_3: \frac{H}{16} \times \frac{W}{16} \times C_3, \quad F_4: \frac{H}{32} \times \frac{W}{32} \times C_4$$

These features combine high-resolution coarse information with low-resolution fine details, which are crucial for dense prediction tasks like segmentation.

Key Technical Components:

In a standard **Vision Transformer (ViT)**, the input image is divided into non-overlapping patches, and the attention mechanism is applied to the resulting sequence of patch embeddings. The attention for each pair of patches is computed using the following formula (1) :

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

Where:

- Q(Query), K (Key), and V (Value) are derived from the input sequence of patch embeddings.
- d_k is the dimensionality of the queries and keys.
- The SoftMax operation ensures that the attention weights sum to 1.

This mechanism has quadratic complexity $O(N^2)$, meaning that the computational cost increases rapidly as the number of patches N grows. This makes ViT less efficient for tasks like semantic segmentation, where high-resolution feature maps are needed.

In contrast, SegFormer enhances the traditional self-attention mechanism in several ways:

Overlapping Patch Merging:

Instead of the original ViT's non-overlapping patch merging, SegFormer uses overlapping patches. This is defined by:

$$K = 7, S = 4, P = 3 \quad or \quad K = 3, S = 2, P = 1$$

This overlapping patch strategy preserves local continuity across patch boundaries, ensuring smoother feature representations and better segmentation performance, especially in the context of high-resolution segmentation outputs.

Efficient Self-Attention with Sequence Reduction:

SegFormer overcomes the Normal ViT complexity limitation by using sequence reduction, which reshapes the input feature matrix and applies a linear projection (2). This reduces the attention complexity from $O(N^2)$ to $O\left(\frac{N^2}{R}\right)$, where R is a reduction factor that decreases across different stages of the model. This makes SegFormer more computationally efficient and suitable for large-scale segmentation tasks.

(2)

$$\widehat{K} = \text{Reshape}\left(\frac{N}{R}, C \cdot R\right)(K), \quad K = \text{Linear}(C \cdot R, C)\left(\widehat{K}\right)$$

In practice, SegFormer sets R to values like $[64,16,4,1]$ from stage 1 to stage 4, which progressively reduces the computational cost as the model moves to deeper layers.

Mix-FFN (Mixed Feed-Forward Network):

Unlike the traditional Vision Transformer, which relies on explicit positional encodings (which can require interpolation if the input resolution changes), SegFormer adopts a simpler and more efficient approach. It uses a 3×3 depth-wise convolution inside the FFN, which provides implicit positional information. This method leverages padding effects in convolutions, making the model more flexible and efficient without the need for fixed-size positional codes. This enables better performance on high-resolution segmentation tasks where the input resolution may vary.

Visualization of Feature Maps

Figure (2) shows the feature maps resulting from multiple stages within the SegFormer model. Each row in the figure represents a different stage or layer in the encoder, with each map showing a set of channels derived from that specific stage. The upper rows, corresponding to the early stages, reveal simple patterns such as edges and gradients, which indicate the extraction of low-level features. In the middle rows, the patterns evolve into more complex

and noisy shapes, reflecting the model's ability to recognize intricate spatial relationships and regions potentially belonging to specific classes. In the lower rows, the object features become more clearly defined, with the feature maps displaying fine-grained details resembling the distribution of roads, buildings, and other semantic structures in the original satellite image. This progression illustrates how SegFormer effectively encodes multiscale semantic information that is later fused by the MLP decoder for accurate segmentation.

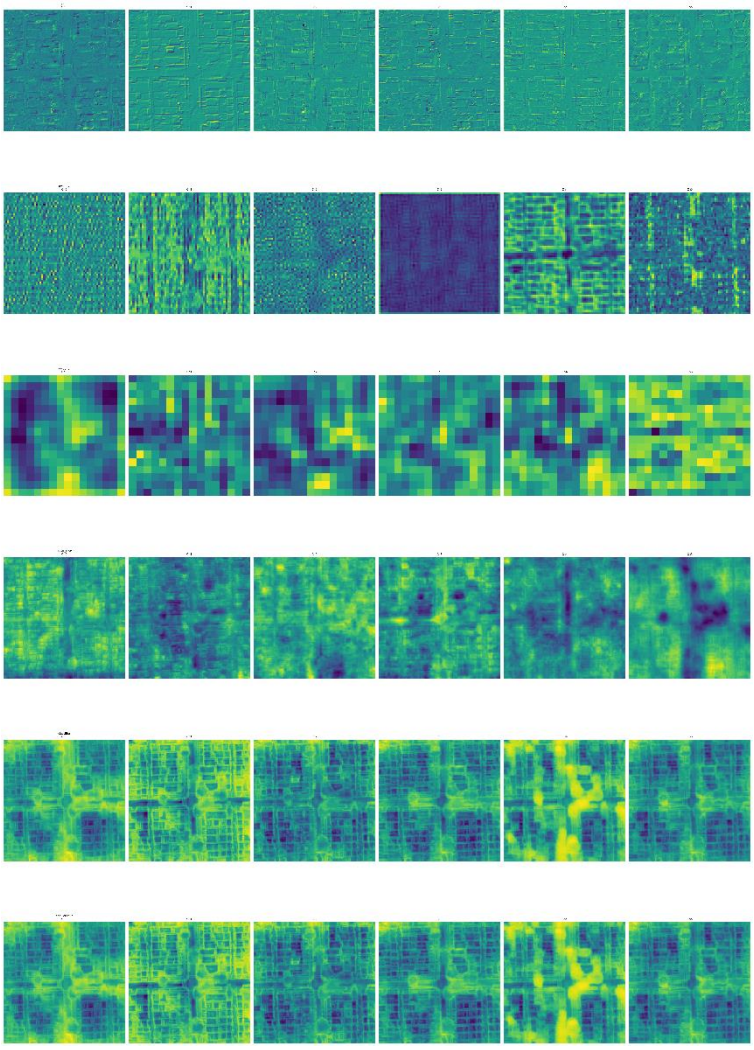

Figure 2:Feature Map of Segfromer

2. Lightweight All-MLP Decoder

The decoder in SegFormer is designed to avoid the heavy, hand-crafted modules commonly found in many previous segmentation models. It consists of:

Channel Unification: Each feature map F_i from the encoder is passed through a linear layer $\text{Linear}(C_i, C)$, where C is a shared channel dimension.

Up sampling: All features are upsampled to $\frac{H}{4} \times \frac{W}{4}$

Feature Fusion: The up sampled features are concatenated and passed through another MLP $\text{Linear}(4C, C)$.

Final Prediction: A final linear layer $\text{Linear}(C, N_{\text{cls}})$ generates the output segmentation mask M of shape $\frac{H}{4} \times \frac{W}{4} \times N_{\text{cls}}$, where$N_{\text{cls}}$ is the number of classes.

This lightweight design works effectively thanks to the encoder's rich multi-scale features and large receptive field.

3.2 Convnext

ConvNeXt, which was introduced by Facebook AI Research in their paper *"A ConvNet for the 2020s"* is a modern convolutional neural network (CNN) architecture designed to extract hierarchical features from high-resolution images efficiently (Figure 3). In this study, ConvNeXt serves as the backbone for feature extraction from road images in remote sensing, which is essential for road segmentation tasks. Its architecture is carefully designed to combine the strengths of traditional CNNs with innovations inspired by Vision Transformers (ViTs), such as large kernel sizes, inverted bottlenecks, and layer normalization, achieving state-of-the-art performance while retaining convolutional efficiency.

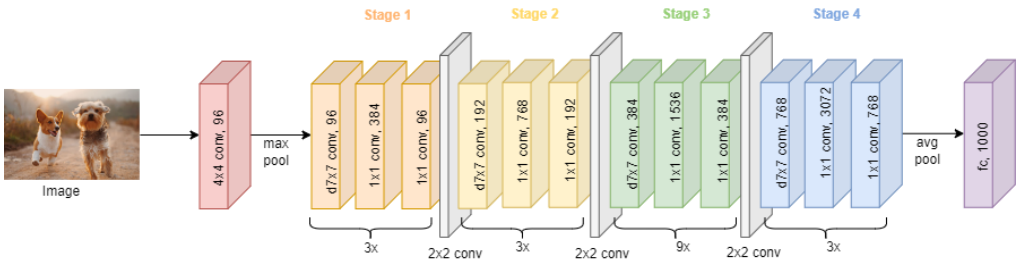

Figure 3: Convnext General architecture

The input to the ConvNeXt model is an image $I \in R^{H \times W \times 3}$,The first step in ConvNeXt is the patch embedding, where the input image is split into smaller

patches, which are then processed independently to form the initial representation. This allows the model to capture both local and global features across the image.

ConvNeXt utilizes stage-wise architecture, consisting of four stages where the spatial resolution of the features decreases progressively. Let F_i represent the feature maps at stage i, where $i \in \{1, 2, 3, 4\}$. At the first stage, the input is processed with convolutions to produce a set of low-level feature maps, denoted as $F_1 \in R^{H/4 \times W/4 \times C_1}$, where C_1 is the number of channels at stage 1. The spatial resolution decreases by a factor of 2 after each subsequent stage, so the feature maps at stages 2, 3, and 4 have sizes $F_2 \in R^{H/8 \times W/8 \times C_2}$, $F_3 \in R^{H/16 \times W/16 \times C_3}$, and $F_4 \in R^{H/32 \times W/32 \times C_4}$, respectively, where C_2, C_3, C_4 are the corresponding numbers of channels.

The key feature of ConvNeXt is its use of depth wise separable convolutions at each stage, which reduce computational complexity by applying convolutions to each channel independently. This results in more efficient architecture while still preserving the model's ability to learn complex features. In addition to depth wise convolutions, ConvNeXt adopts modern innovations such as Layer Normalization instead of batch normalization, which stabilizes training and speeds up convergence. The activations are transformed using GELU (Gaussian Error Linear Units), providing a smoother non-linear transformation compared to traditional ReLU functions. (Figure 4)

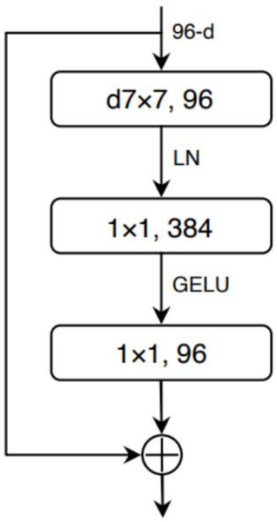

Figure 4:Simple Structure of ConvNeXt

At the output of the ConvNeXt backbone, we obtain four sets of feature maps F_1, F_2, F_3, F_4 , each representing progressively abstract levels of visual information. These feature maps capture both fine-grained details and high-level semantic information, making them suitable for downstream tasks such as road segmentation.

UPerNet Decoder: Feature Fusion and Road Segmentation

Following the feature extraction process in ConvNeXt, the **UPerNet (Unified Perceptual Parsing Network)** decoder is employed to utilize the hierarchical feature maps for the task of semantic segmentation. UPerNet is specifically designed to merge multi-scale features from the backbone, which enables precise segmentation by capturing both large-scale contextual information and fine-grained local details.

The decoder in UPerNet operates through a top-down processing flow, starting with the highest-resolution features and progressively fusing them with lower-resolution, semantically richer feature maps. The first step is the application of a **Pyramid Pooling Module (PPM)**, which processes the feature map $F_4 \in R^{H/32 \times W/32 \times C_4}$ from the deepest layer of the ConvNeXt backbone. The PPM applies pooling operations at different scales, such as global pooling and local pooling with kernel sizes 2x2, 3x3, and 6x6, to capture varying levels of contextual information. The resulting pooled features are then concatenated and upsampled to the original resolution of the image, producing multi-scale features that incorporate global context.

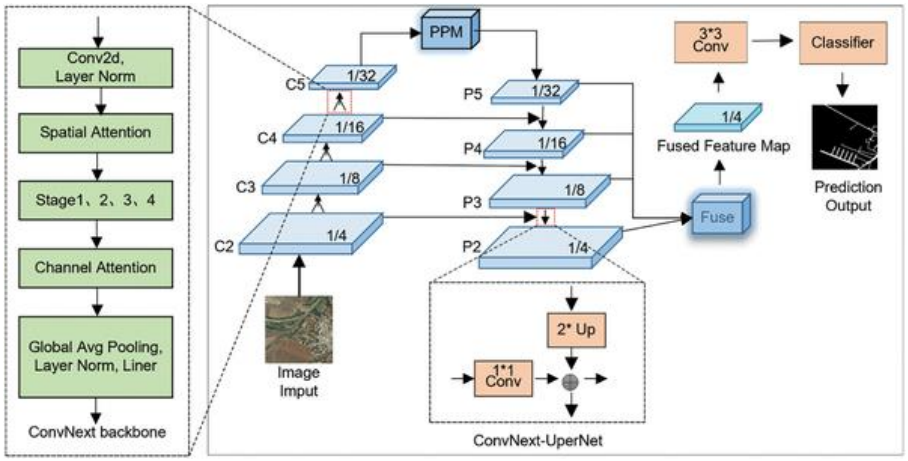

Figure 5: ConvNeXt-UPerNet network structure

Once the pooled features from the PPM are combined, the decoder applies **upsampling** operations to progressively recover the resolution of the earlier

feature maps F_3, F_2, F_1. The decoder fuses these multi-resolution features by utilizing **skip connections**, which directly pass information from corresponding layers in the encoder (ConvNeXt backbone) to the decoder. This ensures that the fine-grained spatial details preserved in the earlier stages of the ConvNeXt backbone are retained and used effectively for the segmentation task.

The multi-scale fused features are passed through a final segmentation head, which consists of a series of convolutional layers that map the fused feature maps to a pixel-wise classification output. This output is a segmentation mask $\hat{Y} \in R^{H \times W \times C}$, where C is the number of classes (2 for binary segmentation: road and background). A **SoftMax** activation function is applied to produce the probability of each pixel belonging to a given class.

Through this integration of the hierarchical features from ConvNeXt and the sophisticated multi-scale fusion mechanisms in UPerNet, the model can effectively segment road features in remote sensing images, preserving both local details and large-scale contextual information. This approach enables accurate segmentation even in complex scenarios where roads are occluded or present at varying scales.

3.3 C-Unet

This section presents the complete architecture of the Complement UNet (C-UNet) based on the paper by Hou et al. (2021), *C-UNet: Complementary U-Net for Remote Sensing Road Extraction*, published in *Sensors*. designed to enhance road segmentation performance in high-resolution remote sensing images (Figure 6). We describe its four main components: the standard UNet backbone, the erasing module, the multi-scale dilated convolution UNet (MD-UNet), and the fusion module. The goal of this architecture is to sequentially extract both the easily segmentable road regions and the harder-to-detect road segments, ultimately producing a complete and accurate road segmentation map.

Let $X \in R^{H \times W \times C}$ denote the input remote sensing image. In the first stage, the image is passed through a standard UNet encoder-decoder backbone, yielding a feature representation F_{unet} (3):

$$F_{\text{unet}} = \text{UNet}(X) \quad (3)$$

To obtain the first segmentation mask, the output feature map F_{unet} is passed through a sigmoid activation function, generating a probability map:

$$\text{Pre}_{\text{unet}} = \sigma(F_{\text{unet}}) \quad (4)$$

where $\sigma(\cdot)$ denotes the element-wise sigmoid. A binarization step is then applied, producing the first predicted mask:

$$\text{Pre}_1 = \text{binarize}(\text{Pre}_{\text{unet}}) \quad (5)$$

In the second stage, a threshold-based erasing module removes pixels confidently predicted as roads (i.e., those exceeding a predefined threshold δ from F_{unet}. The erased feature map F'_{unet} is computed.

The third stage introduces the **multi-scale dilated convolution UNet (MD-UNet)**, which takes the erased feature map F'_{unet} as input and produces a second segmentation probability map:

$$\text{Pre}_{\text{mdunet}} = \text{MD-UNet}(F'_{\text{unet}}) \quad (6)$$

After applying a binary thresholding operation, the second predicted mask is obtained:

$$\text{Pre}_2 = \text{binarize}(\text{Pre}_{\text{mdunet}}) \quad (7)$$

The MD-UNet integrates multi-scale dilated convolutions to expand the receptive field, capturing fine-grained details and thin road segments that the first UNet stage might miss.

Finally, the **fusion module** combines the outputs of the first and third modules to generate the final segmentation result.

The detailed architecture of C-UNet, including the standard UNet backbone, erasing operations, MD-UNet structure, and the fusion pipeline, is depicted in Figure 6. This sequential design allows C-UNet to balance coarse and fine road extraction, addressing the limitations of single-stage segmentation networks.

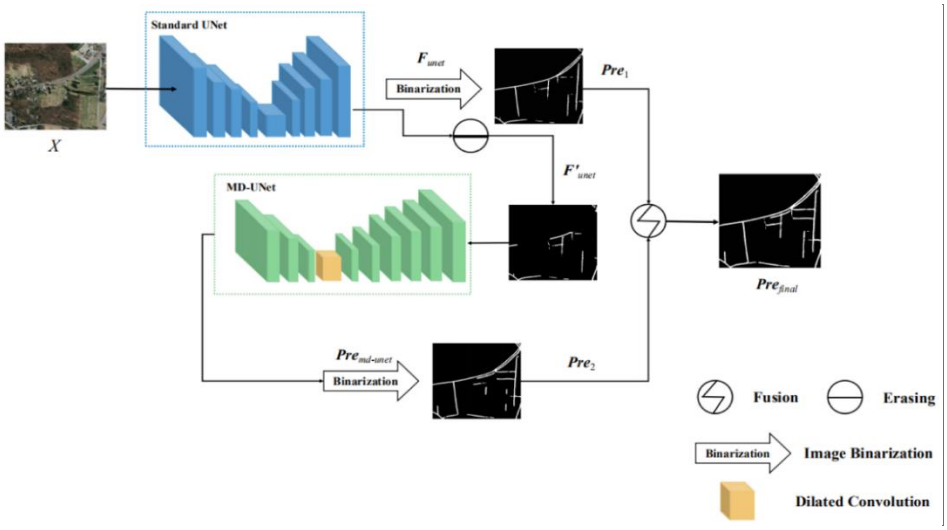

Figure 6 : C-Unet model Architecture

We employed a combined loss function tailored to binary segmentation, integrating **Binary Cross-Entropy (BCE) Loss** and **Dice Loss**. Given the predicted probability map $\widehat{M} \in [0,1]^{H\times W}$, the BCE loss is defined as:

(8)

$$\mathcal{L}\,(BCE) = -\frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W}\left[M_{ij}\log\widehat{M_{ij}} + \left(1 - M_{ij}\right)\log\left(1 - \widehat{M_{ij}}\right)\right]$$

which penalizes the pixel-wise classification errors.

Complementarily, the **Dice Loss** encourages overlap between the predicted mask and the ground truth:

(9)

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_{i,j} M_{ij}\widehat{M_{ij}}}{\sum_{i,j} M_{ij} + \sum_{i,j}\widehat{M_{ij}} + \epsilon}$$

where ϵ is a small constant to avoid division by zero. This loss is especially beneficial for addressing class imbalance, as it directly optimizes the overlap between road pixels and predicted road regions. The final training loss is a weighted sum:

(10)

$$\mathcal{L}(total) = \alpha\mathcal{L}(BCE) + (1 - \alpha)\mathcal{L}_{\text{Dice}}$$

where α balances the contribution of each term.

4. Experiments

4.1 Dataset Description

We conducted all experiments using the EPFL Machine Learning Road Segmentation dataset, a benchmark specifically designed for binary semantic segmentation of road networks from aerial and satellite imagery. The dataset is divided into two main subsets: a training set and a test set. The training set comprises high-resolution satellite images sourced from Google Maps, showcasing diverse urban and suburban scenes, including roads, buildings, parking lots, factories, and vehicles. Each satellite image is paired with a binary ground truth mask, where each pixel is labeled either as “road” or “background.” The test set contains distinct satellite images that do not overlap with the training set, allowing for a robust evaluation of the model’s ability to generalize to unseen road configurations and landscape patterns.

4.2 SegFormer Experiment

For the SegFormer experiment, several preprocessing steps were applied to prepare the data. All images and masks were resized to a uniform resolution of 512 × 512 pixels to ensure consistency across the dataset and compatibility with the SegFormer-B2 architecture. The pixel values of the images were normalized to a range of [0, 1], which is a standard practice to improve numerical stability and accelerate convergence during training. The training set was further divided into a training and validation subset using an 80:20 split, enabling continuous monitoring of validation performance and helping prevent overfitting.

To increase the effective size of the training set and improve the model’s ability to generalize, we applied both offline and online data augmentation strategies using the Albumentations library. Offline, we generated ten augmented versions of each original image-mask pair using random horizontal and vertical flips, rotations, and scaling transformations. Online, we applied additional on-the-fly augmentations during training, such as random 90-degree rotations, Gaussian noise injection, and geometric distortions. These augmentation strategies played a crucial role in improving the model’s robustness to spatial variations and noise.

The SegFormer-B2 model was trained using binary cross-entropy loss to measure the discrepancy between the predicted masks and the ground truth labels. The Adam optimizer was used for its adaptive learning rate and efficient convergence. We employed a ReduceLROnPlateau learning rate scheduler that reduced the learning rate by a factor of 0.5 if no improvement in validation loss was observed over three consecutive epochs. To prevent overfitting, we used early stopping with a patience of 10 epochs, halting training if validation loss stopped improving. Additionally, the best-performing model, determined by the lowest validation loss, was saved via model checkpointing for final evaluation.

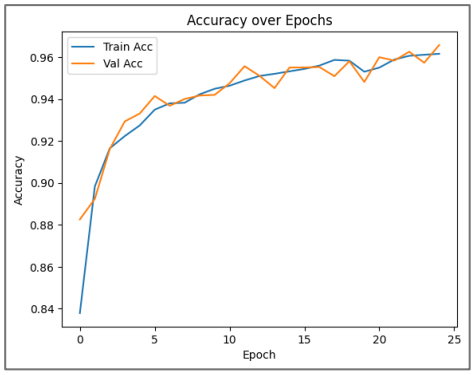

Figure 7:- training & validation accuracies plot for the SegFormer model

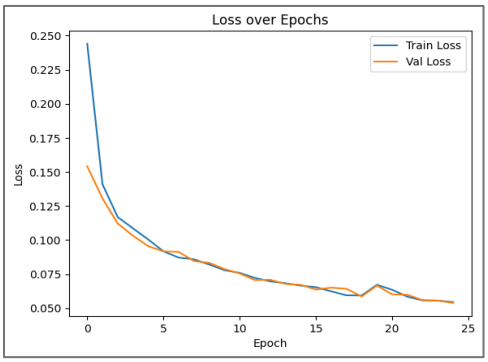

Figure 8:training & validation losses plots for the SegFormer model

Initially, the model was trained for 10 epochs, during which both training and validation performance improved steadily. Encouraged by these results, we extended training to 30 epochs; however, overfitting signs emerged, leading us to reduce the final training to 25 epochs to balance performance and generalization. The final SegFormer-B2 model achieved a training loss of 0.0545 and a validation loss of 0.0539, with accuracy scores of 96.16% (training) and 96.58% (validation). The Dice coefficients reached 0.9615 on the training set and 0.9625 on the validation set, indicating excellent segmentation performance.

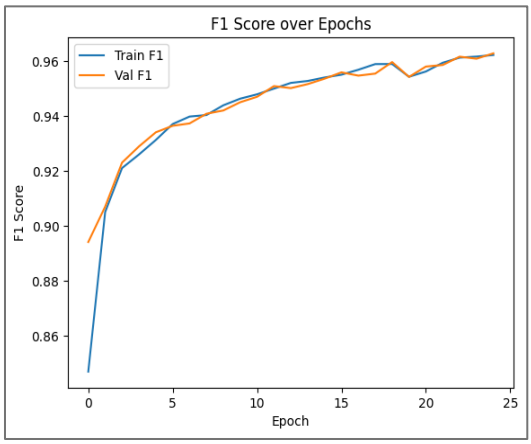

Figure 9:F1 score plot through epochs

Notably, all experiments were conducted on Google Colab's free tier, completing in approximately 1.5 hours, highlighting the approach's computational efficiency and accessibility. Figure 10 presents SegFormer's segmentation on a test set image, while Figure 11 demonstrates its performance on a real-world satellite image from the Jubaiha region in Jordan, underscoring the model's adaptability to slightly distorted, locally sourced samples.

Figure 10:segFormer model inference on a sample from the EPFL test dataset

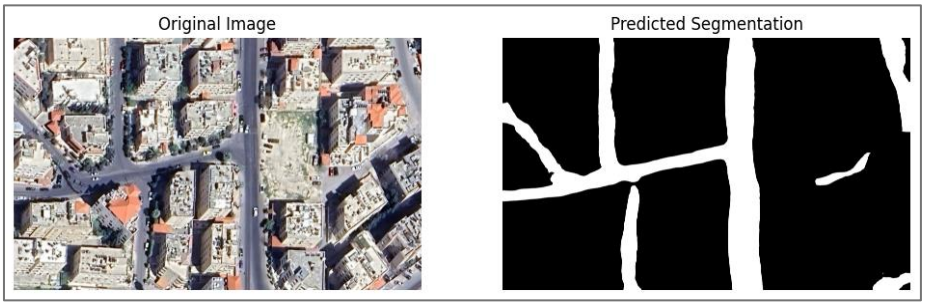

Figure 11: SegFormer model inference on a sample for the Jubieha area taken from google earth.

Metric	Training	Validation
Loss	0.0545	0.0539
Accuracy	0.9616	0.9658
F1 Score	0.9623	0.9629
Dice Score	0.9615	0.9625

Table 1:Table presents the results & evaluation metrics obtained from training the SegFormer model

4.3 C-UNet Experiment

For the C-UNet experiment, the model was trained on the EPFL training set over four distinct stages, each consisting of 10 epochs, resulting in a total of 40 training epochs. During each epoch, we recorded key performance metrics, including training loss, training F1 score, validation loss, and validation F1 score, to monitor learning progress. In the early phases of training, we observed a gradual decrease in both the loss values and a steady increase in the F1 scores, indicating that the model was effectively learning to segment road regions from the background. This positive learning trend continued as the model advanced through the training stages, reflecting continuous performance improvements.

In the final stage, spanning epochs 31 to 40, the training loss dropped from 0.7688 to 0.4573, while the training F1 score increased from 0.7297 to 0.843, signaling the model's enhanced capacity to generate accurate predictions on the training data. Correspondingly, the validation loss improved from 0.5732 to 0.412, and the validation F1 score rose from 0.8399 to 0.854, demonstrating the model's ability to generalize well to unseen validation data. This performance trend is visually depicted in Figure 12, which shows the downward trajectories of both the training and validation loss curves, confirming the model's robust learning behavior across training iterations.

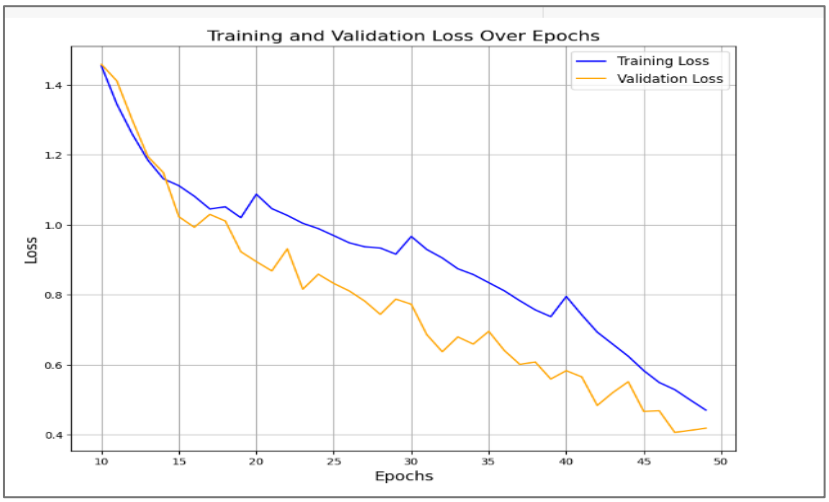

Figure 12:training and validation loss of the C-Unet model

Despite these promising results, several limitations were encountered. Notably, while a T4 GPU was used to accelerate the training process, we observed a progressive increase in RAM usage as the training epochs advanced. After completing 40 epochs, the system's memory consumption

exceeded the allowable threshold, causing the session to terminate unexpectedly. Additionally, the predicted segmentation masks produced by the trained C-UNet model were able to capture most major road networks; however, some artifacts were present. Specifically, the predicted roads appeared thicker than in the original masks, some edge noise was visible, and finer road structures were partially missed or fragmented. These limitations suggest that, while C-UNet is a powerful multi-scale segmentation architecture, it is computationally heavy and requires larger datasets to fully leverage its capacity. Given these constraints, we also experimented with replacing the multi-scale dilated U-Net backbone with a simpler U-Net to better accommodate the available dataset size and hardware constraints. Figure 13 illustrates an inference result from the final C-UNet model, demonstrating its general segmentation capabilities despite these challenges.

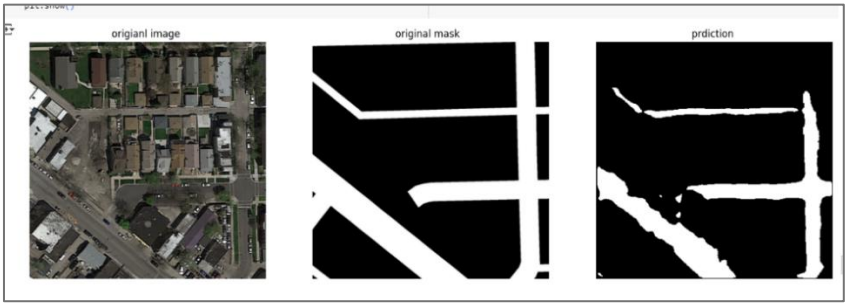

Figure 13:- C-Unet inference on a training sample from the EPFL dataset

4.4 Convnext Experiment

The ConvNeXt model was trained on the EPFL road segmentation dataset for 50 epochs, using 256 × 256 input image patches. We used a combined cross-entropy and Dice loss function to optimize both pixel-wise classification accuracy and global shape consistency in the predicted masks. Model performance was assessed on the validation set using the Intersection-over-Union (IoU) metric, where the model achieved a final validation IoU of 0.5079 and a loss of 0.5523. As visualized in Figure 14, the model encountered notable challenges in complex regions, particularly around road boundaries, thin road segments, and areas affected by shadows, occlusions, or background clutter. These weaknesses ultimately led to suboptimal segmentation outcomes compared to other methods in our study. Future improvements could focus on enhanced data preprocessing and augmentation

strategies tailored to the ConvNeXt architecture, potentially improving its sensitivity to fine-grained road details and difficult spatial patterns.

Figure 14:Convnext inference on a training sample from the EPFL dataset

4.5 Cross-Dataset Evaluation with Satellite Road Segmentation Dataset

To assess the generalization capability of our best-performing model, SegFormer-B2, we conducted a cross-dataset evaluation using the Satellite Road Segmentation Dataset (Timothé Laborie, 2023). This dataset contains high-resolution aerial imagery of Boston and Los Angeles, along with precise road masks, offering a geographically diverse and complex testbed for road segmentation tasks. With a size of approximately 1.7 GB, it provides a realistic benchmark for applications such as urban planning and infrastructure analysis.

In this experiment, the SegFormer model was first trained on the EPFL Road Segmentation Dataset, where it achieved strong results: 96.58% validation accuracy, 96.25% Dice score, and 96.29% F1 score, with a final validation loss of 0.0539 over 25 epochs. We then fine-tuned the model on the Satellite Road Segmentation Dataset for 15 epochs. While performance decreased to 88.2% accuracy, 88.26% Dice score, and 88.34% F1 score, with a final loss of 0.1532, this decline reflects the increased complexity and geographic diversity of the new dataset.

Due to computational constraints (Google Colab free tier), we were unable to extend training to additional epochs, which may have further improved

adaptation. Nevertheless, the small gap between training and validation metrics indicates that the fine-tuned SegFormer retained a robust generalization. A comparative summary of the performance on both datasets is provided in Table 2.

	EPFL Dataset	**Satellite Road Segmentation Dataset**
Accuracy	0.9658	0.8782
Dice Score	0.9625	0.8826
F1 Score	0.9629	0.8834
Loss	0.0539	0.1532
Epochs	25	15

Table 2: Metrices Results between EPFL dataset and Timothé Laborie, 2023 dataset.

4.6 Ablation Study

Objective

This ablation study investigates the impact of key training configurations on the performance of the SegFormer-B2 model applied to the EPFL Road Segmentation Dataset. Specifically, we evaluate:

1. the number of training epochs,
2. the application of on-the-fly data augmentations, and
3. the choice of loss function (Cross-Entropy vs. combined Cross-Entropy and Dice loss).

The goal is to identify configurations that optimize performance and generalization.

Experimental Setup

- **Model Architecture**: SegFormer-B2
- **Dataset**: EPFL Road Segmentation Dataset
- **Baseline Configuration**:
    - Epochs: 25
    - Data Augmentation: Enabled (offline + online)
    - Loss Function: Cross-Entropy Loss
    - Evaluation Metrics: F1 Score, Accuracy, Dice Score, and Loss

1. Varying the Number of Epochs

Configuration	Epochs	Dice Score	F1 Score	Accuracy	Loss
Baseline	25	0.9625	0.9629	0.9658	0.0539
Experiment 1	30	0.9612	0.9619	0.9627	0.0552

Table 3: ablation study on the varying of the number of epochs in Segformer.

Observation

Extending the training beyond 25 epochs led to marginally better training performance but slightly worse validation metrics, indicating potential overfitting. Therefore, 25 epochs appear to provide an optimal balance between training accuracy and generalization for this model and dataset.

2. Disabling On-the-Fly Data Augmentations

Configuration	Data Augmentation	Dice Score	F1 Score	Accuracy	Loss
Baseline	Enabled	0.9625	0.9629	0.9658	0.0539
Without Augmentation	Disabled	0.9726	0.9730	0.9726	0.0563

Table 4:ablation study on disabling on the fly augmentations on Segformer.

Observation

Disabling on-the-fly data augmentation led to improved training and validation metrics, suggesting the model fit the training data more closely. However, the slight increase in validation loss hints at reduced robustness and potential overfitting. Maintaining data augmentation is therefore recommended to improve the model's generalization.

3. Changing the Loss Function

Configuration	Loss Function	Dice Score	F1 Score	Accuracy	Loss
Baseline	Cross-Entropy	0.9625	0.9629	0.9658	0.0539
Combined Loss	Cross-Entropy + Dice	0.9613	0.9620	0.9632	0.0436

Table 5: ablation study on the changing the loss function from cross entropy to combined loss on segformer.

Observation

Incorporating Dice loss alongside Cross-Entropy improved training convergence, as reflected in lower overall loss values. However, this combined loss did not yield significant improvements in segmentation performance metrics such as accuracy, F1 score, or Dice coefficient when compared to Cross-Entropy loss alone. These results suggest that while the combined loss

enhances optimization efficiency, it does not necessarily translate to better generalization for this task.

Conclusion

In this ablation study, we assessed the effects of varying training epochs, data augmentation, and loss function configurations on the SegFormer-B2 model using the EPFL road segmentation dataset. Our findings indicate that training for 25 epochs offers an optimal balance between model performance and generalization, as extending to 30 epochs resulted in slight overfitting. Disabling data augmentation led to improved training metrics but increased validation loss, suggesting potential overfitting and underscoring the importance of augmentation for generalization. Lastly, combining Cross-Entropy and Dice losses improved training convergence but did not yield significant enhancements in segmentation performance metrics compared to using Cross-Entropy loss alone. These results highlight the necessity of carefully selecting training configurations to achieve optimal model performance.

5. Discussion

The comparative analysis of the C-UNet, ConvNeXt-UPerNet, and SegFormer-B2 architecture for road segmentation offers valuable insights into their respective strengths, weaknesses, and practical applicability. Although all three models demonstrated measurable success, their overall suitability varied significantly depending on the specific task requirements and computational constraints.

Among the tested models, **SegFormer-B2** emerged as the clear winner, significantly outperforming the others in accuracy, loss, and Dice score, achieving a near-perfect validation Dice score of 0.9625 and an accuracy of 96.58%. Notably, the model exhibited excellent generalization even when tested on unseen real-world satellite images from Jordan, underscoring its robustness across diverse geographic contexts and imaging conditions. The combination of its strong encoder-decoder attention mechanisms and lightweight transformer-based design enabled efficient training even on resource-limited platforms, making it a highly promising architecture for road segmentation tasks involving complex urban layouts. Its success highlights the power of transformer architecture in capturing both global context and fine-grained spatial details, which are essential for accurately segmenting road networks, especially in heterogeneous environments.

In contrast, the **C-UNet** model, while showing some promise (achieving an F1 score of 0.854 on the validation set), exhibited inconsistent performance across different evaluation conditions. Although its two-stage prediction system improved detection in ambiguous regions, several limitations were apparent: over-segmentation resulting in thicker road predictions compared to the ground truth, RAM instability due to memory leaks during extended training, and failure to detect thin or narrow roads, which are often critical in urban road networks. These issues highlight the need for architectural or preprocessing improvements to make C-UNet a competitive option in this domain.

Finally, the **ConvNeXt-UPerNet** model, despite its sophisticated backbone, underperformed notably, achieving only a 0.5079 IoU on the validation set. The main challenges observed included ambiguity at road boundaries, where the model struggled to precisely delineate road edges, and a pronounced sensitivity to scale, particularly with roads narrower than five pixels. These limitations suggest that while ConvNeXt-based designs hold theoretical promise, they may require significant adaptation or additional architectural tuning to handle the unique challenges of road segmentation from aerial imagery effectively.

Overall, the findings from this comparative study emphasize that while modern CNN and transformer-based models can all achieve some level of success in road segmentation, the choice of architecture has substantial implications for accuracy, robustness, and scalability. SegFormer-B2's balance of efficiency and generalization makes it particularly well-suited for practical deployment, while improvements to C-UNet and ConvNeXt-UPerNet architectures could focus on addressing the specific weaknesses identified in this work.

6. Conclusion and Future Work

This project developed a road segmentation system based on the SegFormer model, trained and evaluated on the EPFL Road Segmentation Dataset. The system achieved strong results, with a validation accuracy of 0.9658, an F1 score of 0.9629, and a Dice coefficient of 0.9625, demonstrating the model's effectiveness in handling road segmentation tasks. SegFormer was deliberately selected due to its balance between high performance and lightweight architecture, making it particularly suitable for deployment on resource-constrained platforms. To address the challenges posed by limited data, transfer learning and on-the-fly data augmentation techniques were incorporated, enhancing the model's ability to generalize despite the relatively small dataset size.

Looking ahead, the project aims to expand the dataset to include satellite imagery of roads in Jordan, introducing additional layers of complexity such as varying weather conditions, soil types, and geographic diversity. Incorporating these factors will not only improve the robustness of the segmentation predictions but also test the model's adaptability across diverse environmental conditions. Furthermore, the project plans to explore advanced models, such as DeepLabV3+, and conduct systematic comparisons with SegFormer to benchmark their relative strengths and weaknesses. The ultimate goal is to integrate the refined segmentation system into smart navigation solutions and autonomous vehicle applications, where precise and reliable road extraction plays a critical role in enabling safe and efficient real-world deployment.

7. References

An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. (2020). *International Conference on Learning Representations (ICLR)*.

Complement UNet: A Complementary Network for Remote Sensing Road Extraction. Hou, Y., Liu, Q., He, K., Wang, H. (2021). *Sensors*, 21(6), 2153.

C-UNet: Complement UNet for Remote Sensing Road Extraction. Hou, Y., Liu, Z., Zhang, T., Li, Y. (2021).

Conditional Positional Encodings for Vision Transformers. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C., Wang, L. (2021). *International Conference on Computer Vision (ICCV)*.

CoaT: Co-Scale Conv-Attentional Image Transformers. Xu, Y., Zhang, Q., Wei, Q., Bai, S., Zhang, C., Sato, I., Sugiyama, M. (2021). *International Conference on Computer Vision (ICCV)*.

CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. Chen, C.-F., Fan, Q., Panda, R. (2021). *International Conference on Computer Vision (ICCV)*.

CvT: Introducing Convolutions to Vision Transformers. Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., Zhang, L. (2021). *International Conference on Computer Vision (ICCV)*.

DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. (2018). *IEEE TPAMI*.

DeepLabv3+: Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation. Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H. (2018).

Fully Convolutional Networks for Semantic Segmentation. Long, J., Shelhamer, E., Darrell, T. (2015).

LeViT: A Vision Transformer in ConvNet's Clothing for Faster Inference. Graham, B., El-Nouby, A., Touvron, H., Stock, P., Joulin, A., Jegou, H., Douze, M. (2021). *ICCV*.

LocalViT: Bringing Locality to Vision Transformers. Li, Z., Zhang, H., Hu, X., Yang, J. (2021). *ICCV*.

Mask R-CNN Based Automated Identification and Extraction of Oil Well Sites. He, H., Xu, H., Zhang, Y., Gao, K., Li, H., Ma, L., Li, J. (2022). *International Journal of Applied Earth Observation and Geoinformation*, 112.

Pyramid Vision Transformer: A Versatile Backbone for Dense Prediction without Convolutions. Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L. (2021). *ICCV*.

Remote Sensing Image Registration Methodology: Review and Discussion. Tondewad, P. S., Dale, M. P. (2020). *Procedia Computer Science*, 171, 2390–2399.

Rethinking Attention with Performers. Choromanski, K., Likhosherstov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., Belanger, D., Colwell, L., Weller, A. (2020). *ICLR*.

SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., Luo, P. (2021). *NeurIPS*.

SETR: Semantic Segmentation with Transformers. Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P. H., Zhang, L. (2021).

SERNet-Former: Semantic Segmentation by Efficient Residual Network with Attention-Boosting Gates and Attention-Fusion Networks. Erişen, S. (2024).

State of the Art on Automatic Road Extraction for GIS Update: A Novel Classification. Mena, J. B. (2003). *Pattern Recognition Letters*, 24(16), 3037–3058.

Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B. (2021). *ICCV*.

Tokens-to-Token ViT: Training Vision Transformers from Scratch on ImageNet. Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z., Tay, F. E., Feng, J., Yan, S. (2021). *ICCV*.

Training Data-Efficient Image Transformers & Distillation Through Attention. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H. (2021). *ICML*.

Transformer in Transformer. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y. (2021). *NeurIPS*.

Twins: Revisiting the Design of Spatial Attention in Vision Transformers. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C., Wang, L. (2021). *NeurIPS*.

U-Net: Convolutional Networks for Biomedical Image Segmentation. Ronneberger, O., Fischer, P., Brox, T. (2015). *MICCAI*.
Retrieved from https://paperswithcode.com/method/u-net

Your ViT is Secretly an Image Segmentation Model. Kerssies, T., Cavagnero, N., Hermans, A., Norouzi, N., Averta, G., Leibe, B., Dubbelman, G., Geus, D. (2025). *CVPR*.

A ConvNet for the 2020s. Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., Xie, S. (2022). *CVPR*.
(Also known as ConvNeXt)

A Comparative Study of Loss Functions for Road Segmentation in Remotely Sensed Road Datasets. Xu, H., He, H., Zhang, Y., Ma, L., Li, J. (2023).

Heuristic Comparison of Vision Transformers Against Convolutional Neural Networks for Semantic Segmentation on Remote Sensing Imagery. Dahal, A., Akbar Murad, S., Rahimi, N. (2024).

Seeing the Roads Through the Trees: A Benchmark for Modeling Spatial Dependencies with Aerial Imagery. Robinson, C., Corley, I., Ortiz, A., Dodhia, R., Lavista Ferres, J. M., Najafirad, P. (2024).

RS-Mamba for Large Remote Sensing Image Dense Prediction. Zhao, S., Chen, H., Zhang, X., Xiao, P., Bai, L., Ouyang, W. (2024).

Hugging Face Transformers Documentation. (n.d.). SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. https://huggingface.co/docs/transformers/en/model_doc/segformer

CS-433 Road Segmentation Project. (n.d.). GitHub Repository. https://github.com/CS-433/Road_Segmentation

satellite-road-segmentation-dataset. Laborie, T. (2023). Retrieved from https://www.kaggle.com/datasets/timothlaborie/roadsegmentation-boston-losangeles?select=images

An Improved Road and Building Detector on VHR Images. Simler, C. (2011). *Proceedings of IGARSS*, 507–510.