





**Please cite the Published Version**

Taparia, Aditya , Bashir, Ali Kashif , Zhu, Yaodong , Gdekallu, Thippa Reddy  and Nath, Keshab (2024) Transforming satellite imagery into vector maps using modified GANs. Alexandria Engineering Journal, 109. pp. 792-806. ISSN 1110-0168

**DOI:** <https://doi.org/10.1016/j.aej.2024.09.074>

**Publisher:** Elsevier

**Version:** Published Version

**Downloaded from:** <https://e-space.mmu.ac.uk/636234/>

**Usage rights:**  [Creative Commons: Attribution 4.0](https://creativecommons.org/licenses/by/4.0/)

**Additional Information:** This is an open access article published in Alexandria Engineering Journal, by Elsevier.

**Data Access Statement:** We have used publicly available maps dataset in this study. This data can be found here: <http://efrogans.eecs.berkeley.edu/pix2pix/datasets/maps.tar.gz>.

**Enquiries:**

If you have questions about this document, contact [openresearch@mmu.ac.uk](mailto:openresearch@mmu.ac.uk). Please include the URL of the record in e-space. If you believe that your, or a third party's rights have been compromised through this document please see our Take Down policy (available from <https://www.mmu.ac.uk/library/using-the-library/policies-and-guidelines>)



Original article



# Transforming satellite imagery into vector maps using modified GANs

Aditya Taparia<sup>a</sup>, Ali Kashif Bashir<sup>b,g,h</sup>, Yaodong Zhu<sup>c,\*</sup>, Thippa Reddy Gdekallu<sup>d,e,g</sup>, Keshab Nath<sup>f</sup>

<sup>a</sup> Department of Computer Science and Engineering, Indian Institute of Information Technology, Kottayam, India

<sup>b</sup> Department of Computing and Mathematics, Manchester Metropolitan University, UK

<sup>c</sup> Jiaxing University School of Information Science and Engineering, Jiaxing Zhejiang 314001, China

<sup>d</sup> The College of Mathematics and Computer Science, Zhejiang A&F University, Hangzhou 311300, China

<sup>e</sup> Division of Research and Development, Lovely Professional University, Phagwara, India

<sup>f</sup> Department of Computer Science and Engineering, Bhattadev University, Bajali, India

<sup>g</sup> Center of Research Impact and Outcome, Chitkara University, Punjab, India

<sup>h</sup> Department of Computer Science and Mathematics, Lebanese American University, Beirut, Lebanon

## ARTICLE INFO

Dataset link: <http://efrogans.eecs.berkeley.edu/pix2pix/datasets/maps.tar.gz>

### Keywords:

Image-to-image translation  
Generative adversarial networks  
Hierarchical feature learning  
Road intersections mapping  
Building footprint clustering

## ABSTRACT

Vector maps find widespread utility across diverse domains due to their capacity to not only store but also represent discrete data boundaries such as building footprints, disaster impact analysis, digitization, urban planning, location points, transport links, and more. Although extensive research exists on identifying building footprints and road types from satellite imagery, the generation of vector maps from such imagery remains an area with limited exploration. Furthermore, conventional map generation techniques rely on labor-intensive manual feature extraction or rule-based approaches, which impose inherent limitations. To surmount these limitations, we propose a novel method called **HPix**, which utilizes modified Generative Adversarial Networks (GANs) to generate vector tile map from satellite images. HPix incorporates two hierarchical frameworks: one operating at the global level and the other at the local level, resulting in a comprehensive model. Through empirical evaluations, our proposed approach showcases its effectiveness in producing highly accurate and visually captivating vector tile maps derived from satellite images. We achieved a pixel-level accuracy of 61.04% and an SSIM score of 0.75, outperforming all other existing methods. We further extend our study's application to include mapping of road intersections and building footprints cluster based on their area. We also show usability of our proposed architecture as a general-purpose solutions in other tasks like edges-to-photo, BW-to-color, or labels-to-street scene. **GitHub:** <https://github.com/aditya-taparia/Satellite-Image-to-Vector-Map>.

## 1. Introduction

Vector maps offer a modern and versatile representation of geographical data, surpassing traditional maps in terms of precision and functionality. These maps accurately store and depict discrete data boundaries, including building footprints, disaster impacts, and transportation links, making them indispensable tools across various domains such as urban planning, disaster management, and transportation logistics.

Traditional approaches to generating vector maps from satellite images rely on labor-intensive processes involving manual feature extraction or rule-based methodologies. These techniques, while useful, impose inherent limitations, particularly in terms of scalability, accuracy, and efficiency [1,2]. The advent of deep learning, particularly

Convolutional Neural Networks (CNNs) [3–5] and Generative Adversarial Networks (GANs) [6,7], has revolutionized the field by automating the feature extraction process and enabling the generation of highly detailed vector maps. However, even with these advancements, existing methods often struggle with capturing the full complexity of satellite imagery, especially when dealing with large-scale or highly detailed features.

To address these challenges, we propose a novel approach termed **HierarchicalPix (HPix)**, which utilizes a hierarchical Generative Adversarial Network (GAN) framework. This approach is designed to overcome the limitations of traditional and single-level GAN methods by operating at both global and local levels. The hierarchical structure allows the model to capture the overall layout and structure of the map

\* Corresponding author.

E-mail addresses: [aditya2019@iiitkottayam.ac.in](mailto:aditya2019@iiitkottayam.ac.in) (A. Taparia), [dr.alikashif.b@ieee.org](mailto:dr.alikashif.b@ieee.org) (A.K. Bashir), [nuaazyd@mail.zjxu.edu.cn](mailto:nuaazyd@mail.zjxu.edu.cn) (Y. Zhu), [thippareddy@ieee.org](mailto:thippareddy@ieee.org) (T.R. Gdekallu), [keshabnath@bhattadevuniversity.ac.in](mailto:keshabnath@bhattadevuniversity.ac.in) (K. Nath).

<https://doi.org/10.1016/j.aej.2024.09.074>

Received 1 March 2024; Received in revised form 15 August 2024; Accepted 21 September 2024

Available online 1 October 2024

1110-0168/© 2024 The Authors. Published by Elsevier B.V. on behalf of Faculty of Engineering, Alexandria University. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

at a global level while simultaneously refining finer details at a local level. This two-tiered approach not only enhances the accuracy of the generated vector maps but also significantly reduces the occurrence of artifacts, which are common in single-level GAN outputs.

### 1.1. Motivation for the hierarchical approach

The rationale behind adopting a hierarchical GAN approach is rooted in the need to balance the trade-off between capturing global structures and refining local details. Traditional GAN-based methods often face challenges in simultaneously maintaining the integrity of large-scale features (such as the overall road network) and the accuracy of small-scale details (such as individual building footprints). A single-level GAN model might either overlook finer details due to its focus on global patterns or, conversely, generate detailed local features while losing sight of the broader context.

The hierarchical approach in HPix addresses these issues by decomposing the map generation process into two stages. The **global generator** focuses on producing a coarse but structurally coherent representation of the entire map, ensuring that large-scale features are accurately captured. Subsequently, the **local generator** refines this coarse map by focusing on smaller regions, enhancing the detail and accuracy of specific features such as building edges and road intersections. This two-stage process allows HPix to produce vector maps that are not only accurate at a macro level but also rich in detail at a micro level.

Moreover, the hierarchical structure inherently supports better handling of diverse geographical data, as it allows the model to adjust its focus depending on the level of detail required. This makes HPix particularly suitable for applications where both high-level structural accuracy and fine-grained detail are critical, such as disaster response planning or urban infrastructure development.

The proposed HPix model offers several distinct advantages over existing algorithms, particularly in the context of satellite image to vector map generation. By employing a hierarchical GAN framework, HPix effectively captures both global structures and local details, leading to higher overall accuracy. The inclusion of both global and local generators helps reduce artifacts commonly seen in single-level GAN approaches, resulting in more visually coherent and structurally accurate vector maps. Moreover, HPix demonstrates versatility in handling various image-to-image translation tasks, making it a robust tool for diverse applications such as urban planning, disaster response, and infrastructure management.

### 1.2. Key contributions

This paper makes the following significant contributions:

- We propose a novel hierarchical GAN architecture, termed HierarchicalPix (HPix), which comprises two GAN frameworks—one operating at a global level and the other at a local level. This hierarchical approach effectively captures both the overall structure and fine-grained details of satellite imagery in the generated vector maps.
- Through extensive empirical evaluations, we demonstrate that HPix outperforms existing methods such as Pix2Pix, CycleGAN, MapGen-GAN, and CscGAN in generating vector maps, achieving a pixel-level accuracy of 61.04% and an SSIM score of 0.75, which are superior to those reported by previous methods.
- We extend the application of HPix beyond vector map generation to include tasks such as mapping road intersections and classifying building clusters based on their area. Furthermore, we showcase the potential of HPix as a general-purpose solution for other image-to-image translation tasks, such as edges-to-photo, BW-to-color, or labels-to-street scenes.
- The hierarchical design of HPix, particularly the use of a local generator to refine outputs produced by the global generator, significantly reduces the occurrence of artifacts, resulting in higher quality and more visually coherent vector maps.

## 2. Related work

Recent research in vector map generation from satellite imagery has significantly advanced due to deep learning and computer vision technologies [8,9]. Convolutional Neural Networks (CNNs) have been particularly transformative, offering automated feature extraction that surpasses the accuracy of manual methods, reducing human error and labor intensity. CNNs, such as those developed by Iino et al. (2018) and Hormese et al. (2016), excel in identifying and segmenting complex spatial features like building footprints and road networks. These models learn hierarchical data representations, enabling detailed and accurate vector maps essential for urban planning, disaster management, and infrastructure development.

Despite these advances, the process is not without its limitations. Traditional vector map generation techniques—often manual or rule-based—are time-consuming and prone to inaccuracies when scaling to large datasets or complex urban topographies. Such methods also require substantial domain knowledge, which limits their adaptability and flexibility across varying geographical datasets.

Moreover, while deep learning models offer improved efficiency, they depend heavily on the availability of high-quality, labeled training data. The lack of such data can hinder the performance of these models. Furthermore, these models often struggle with overfitting to training data specifics, which can degrade their performance on unseen images or diverse conditions.

Generative Adversarial Networks (GANs) have introduced capabilities to generate more detailed and visually appealing vector maps by mimicking real-world data patterns. However, they too face challenges such as training stability and the need for careful hyperparameter tuning to avoid issues like mode collapse, where the model fails to capture the diversity of the input data.

Innovative solutions, such as the HPix model introduced by Taparia et al. leverage modified GANs to address some of these issues by using a hierarchical approach that processes images at both global and local scales for enhanced detail and accuracy. This method shows promise in overcoming some of the traditional barriers but continues to face challenges related to computational demands and the complexity of integrating multiple neural network architectures.

### 2.1. Image-to-image translation in satellite imagery

A substantial amount of work has been done in the field of image-to-image translation, particularly with the advent of deep learning algorithms that have outperformed traditional machine learning approaches. Among these, Variational Auto-encoders (VAEs) and Generative Adversarial Networks (GANs) have been at the forefront [10]. While VAEs provided more stable training compared to GANs, they faced several unsolved practical and theoretical challenges, which led to the increased adoption of GANs in various image-to-image tasks [10]. Since the introduction of GANs by Ian Goodfellow in 2014 [11], significant progress has been made in developing and improving different types of GANs tailored for specific image translation tasks [12].

One of the most influential derivatives of GANs is the Conditional GAN (cGAN), introduced by Mirza and Osindero [13]. cGANs enhanced the ability to control the output of GANs by incorporating additional information alongside a random vector as input. This approach was further advanced by Isola et al. [14], who introduced the Pix2Pix model, which has become a seminal work in the field of image-to-image translation. Pix2Pix employed a U-Net-inspired architecture for the generator and a convolutional Patch-GAN classifier for the discriminator, showing promising results in generating images with both high visual fidelity and structural accuracy.

In 2017, Zhu et al. expanded on the capabilities of cGANs by introducing CycleGAN [15], a model designed for unpaired image-to-image translation. CycleGAN addressed the challenge of lacking paired

training data by introducing cycle-consistency loss, which allowed the model to learn mappings between two domains without requiring paired examples. This innovation significantly broadened the applicability of image-to-image translation models, particularly in domains where acquiring paired datasets is difficult.

Subsequent research has focused on enhancing the performance of these models in specific applications, such as satellite image processing. For instance, Ingale et al. [16] proposed a modified version of Pix2Pix tailored for generating vector maps from satellite images. However, while these approaches have demonstrated substantial improvements, they often struggle with accurately representing complex urban environments and preserving fine-grained details in the generated maps.

To address these limitations, more recent models like MapGen-GAN [17] and CscGAN [18] have been developed. MapGen-GAN employs an unsupervised adversarial learning approach, reducing the dependency on large labeled datasets, while CscGAN introduces conditional scale-consistent generation to maintain structural integrity across different scales. Despite these advancements, challenges remain in capturing the full complexity of satellite imagery, particularly in representing small-scale features like narrow roads and building edges.

For our model comparison, we consider Pix2Pix [14] and CycleGAN [15] as baseline models and include the latest approaches like MapGen-GAN [17] and CscGAN [18]. We evaluate these models using Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), and pixel-level accuracy as metrics. These metrics have been proven effective for comparing the quality of images generated for this problem statement [10,17,18]. Detailed explanations of these metrics will be provided in the experiment section of this paper.

## 2.2. Feature extraction from satellite images

In addition to generating vector maps, satellite images are a valuable resource for extracting critical information such as road networks, building footprints, and clusters. This information can be utilized in various applications, including urban planning, disaster response, and infrastructure management [19]. The use of artificial neural networks (ANNs) for extracting features from satellite images was first proposed by Mokhtarzade and Zoj in 2007 [20]. Since then, the field has seen significant advancements, with many methods based on Convolutional Neural Networks (CNNs), Fully Convolutional Networks (FCNs), and U-Net architectures emerging as leading approaches for satellite image segmentation and feature extraction.

One of the most influential models in the field of image segmentation is the U-Net architecture, introduced by Ronneberger et al. in 2015 [21]. Originally designed for biomedical image segmentation, U-Net's symmetric encoder-decoder structure and skip connections have made it highly effective in capturing both local and global features in images. This architecture has been widely adopted and adapted for satellite image segmentation tasks, such as road network extraction and building footprint identification. However, despite its effectiveness, U-Net can sometimes struggle with the large variability and high resolution typical of satellite imagery, leading to inaccuracies in feature extraction.

To address these challenges, researchers have explored enhancements and alternatives to the U-Net architecture. For instance, D-LinkNet, as used by Zhou et al. [22], incorporates residual blocks and skip connections within an encoder-decoder framework to improve the extraction of high-level information from satellite images. This architecture has shown promising results in road network extraction, particularly when combined with advanced loss functions like Dice loss and the Jaccard index (IoU).

Another significant development is the use of ResNet-based U-Net architectures, which combine the strengths of U-Net with the deep feature learning capabilities of ResNet. For example, in a study by Alsabhan and Alotaiby [23], a U-Net architecture with a ResNet50

backbone was shown to achieve superior results in building footprint extraction compared to other state-of-the-art models like Deeplabv3. This combination of deep residual networks and U-Net has proven effective in improving the model's understanding of complex structures within satellite images, leading to more accurate segmentation results.

Despite these advancements, challenges remain in effectively capturing fine details in satellite imagery, particularly in high-resolution images with complex urban environments. To improve segmentation accuracy, researchers have increasingly focused on optimizing loss functions. Region-based loss functions, such as Dice loss and Jaccard index (IoU), have been found to outperform traditional distribution-based loss functions, particularly in tasks involving the segmentation of satellite images [22,23].

Our proposed work builds on these foundational architectures and techniques, integrating their strengths into a novel framework tailored for satellite imagery. U-Net++ [24], a further evolution of U-Net, introduces nested skip connections to enhance the model's representational power, allowing for more precise feature extraction at multiple scales. In our HPix model, we incorporate U-Net++ within the global generator to ensure robust structural coherence, particularly for large-scale features like road networks and building clusters. This global output is then refined by a local generator based on a modified Pix2Pix architecture [14], which enhances fine details such as the shapes of buildings and the layout of small roads.

Additionally, we adopt the use of advanced loss functions, leveraging Dice loss during training and validating our model with the IoU score. This approach is consistent with the findings from previous studies that have demonstrated the effectiveness of these metrics in improving model performance in satellite image segmentation tasks. Our research also utilizes the Massachusetts road and building dataset [25], which has been widely used in the literature for training, testing, and comparing segmentation models.

## 3. Methodology

The proposed approach comprises three parts: generating a base vector map using proposed architecture (HPix), identifying the road intersections and classifying building clusters based on size, and generating interactive vector map.

### 3.1. Hpix (HierarchicalPix)

In this paper, we propose a novel architecture for translating satellite images to vector maps termed HierarchicalPix. This architecture comprises two GAN frameworks, one at global level and other at local level, together forming a hierarchical model, as shown in Fig. 1. In this approach, the generator at global level would generate a coarse version of the vector map from the satellite image, capturing the overall layout and structure of the map. Then the generator at local level would then take the coarse map and the satellite image as input, and generate a more detailed version of the vector map, capturing fine-grained details and features. The local level generator also helps in reducing the artifact formation in the generated image.

The global GAN architecture comprises two components, generator and discriminator. The generator at global level comprises a complex network of encoder and decoder inspired from U-Net++ architecture [24], while for discriminator we are using PatchGAN, introduced in [14], with slight modification in its CNN Block. The local GAN architecture also comprises two components, a generator and a discriminator. While the local discriminator is identical as global discriminator, for local generator we are using modified Pix2Pix architecture [14] which takes our original image along with global generated image as input to give final generated image. More detail about generator and discriminator are explained in the following subsection.



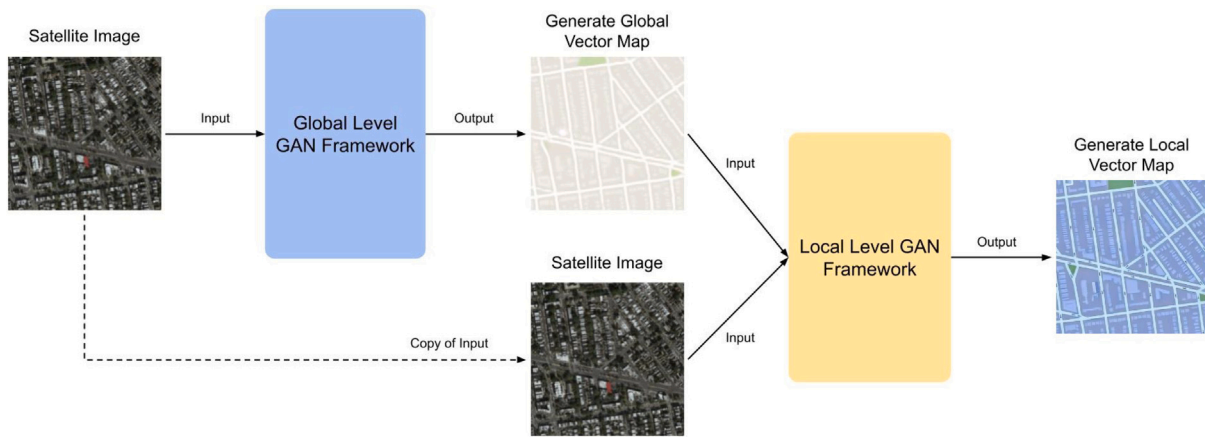


Fig. 1. HierarchicalPix architecture.

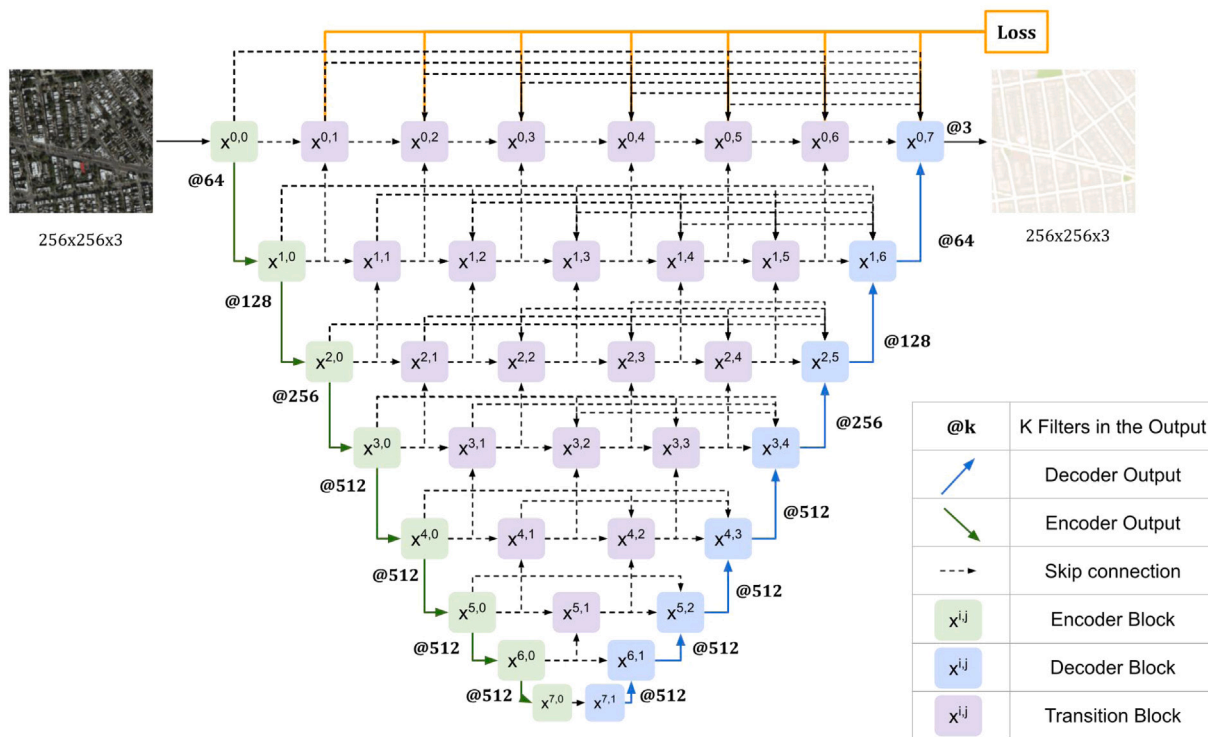


Fig. 2. Global generator architecture with nested skip connection network.

### 3.1.1. Global generator

Authors of the paper [14] explained how the use of skip connection, inspiring from Unet [26], improved the output of their generator model and with that intuition we worked on improving the connection network of the generator. This generator design was inspired from Unet++ architecture [24]. In this architecture apart from our standard encoder–decoder network we have introduced transition blocks which take encoded data from lower level and decode them and combines that information with information from other blocks at the same level and encodes it again before passing that information further. We have also applied deep supervision to further stabilize the output of the model.

Fig. 2 describes the architecture of the global generator used in HierarchicalPix. It also describes the number of feature channels in the output of each encoder and decoder node. The number of features in the transition block is the same as the encoder block on that level. Fig. 3 describes the architecture of encoder, decoder and transition

blocks used in the network. The encoder block comprises a Conv-InstanceNorm-LeakyReLU layer. The first encoder block ( $x_{0,0}$ ) does not apply InstanceNorm to its convoluted output and the bottleneck encoder block ( $x_{7,0}$ ) does not apply InstanceNorm and LeakyReLU to its convoluted output. Our decoder block comprises a ConvTranspose-InstanceNorm-ReLu layer which is followed by a dropout layer with 50% probability. The final decoder block ( $x_{0,7}$ ) does not apply InstanceNorm and LeakyReLU to its convoluted output, instead just applies Tanh. We have used skip connections to pass the feature map information between encoder, decoder and transition blocks. While applying deep supervision, output from blocks  $x_{0,1}$ ,  $x_{0,2}$ ,  $x_{0,3}$ ,  $x_{0,4}$ ,  $x_{0,5}$  and  $x_{0,6}$  are first passed through a convolution layer followed by Tanh and are then considered for calculating loss.

In this architecture, we have used an instance norm layer rather than a batch norm layer and applied reflection padding to reduce the appearance of artifacts in the generated output [15].

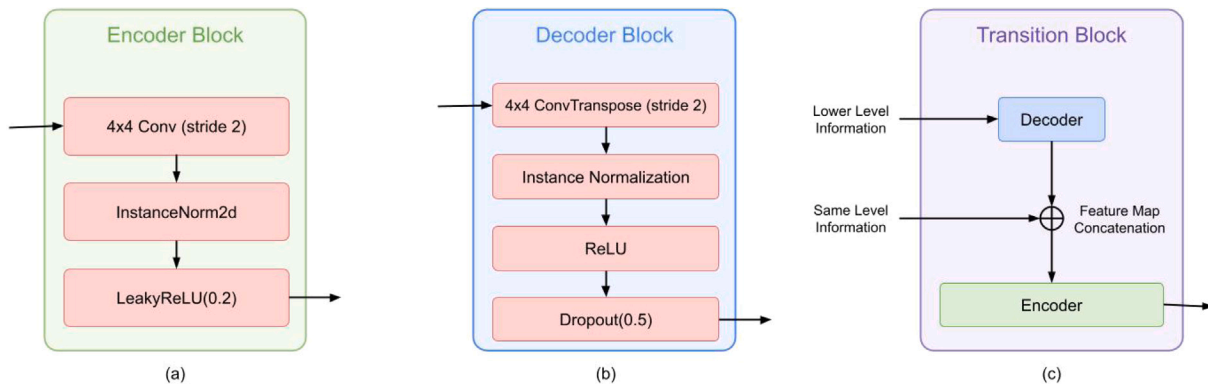


Fig. 3. Architecture of (a) Encoder, (b) Decoder and (c) Transition Blocks used in generators.

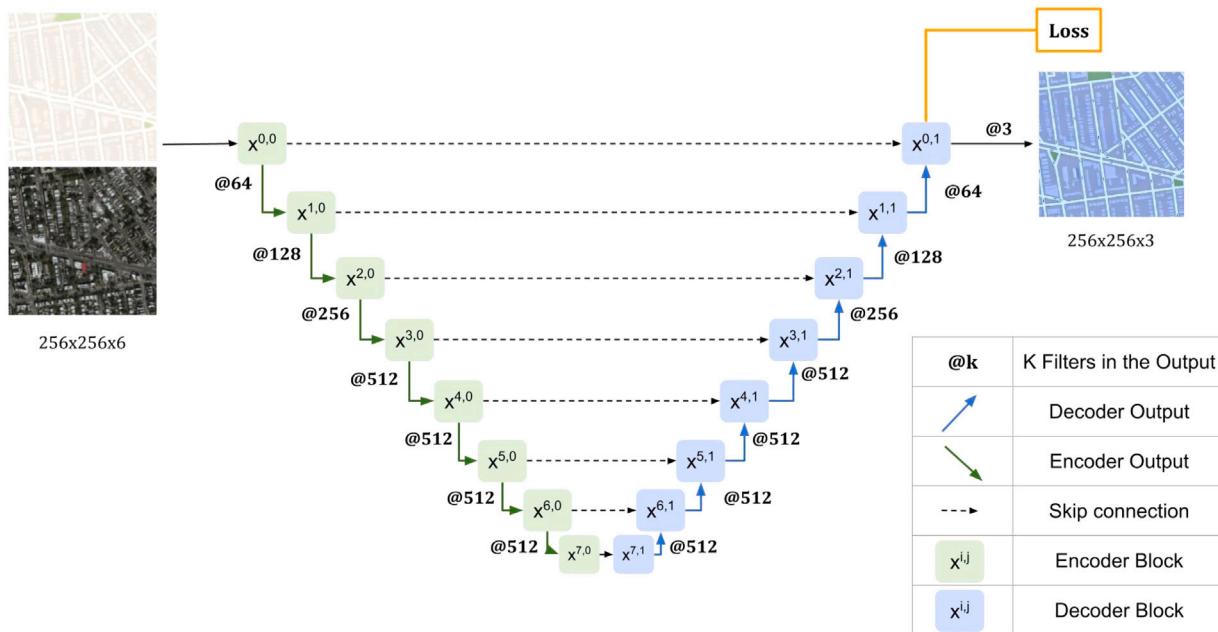


Fig. 4. Local generator architecture.

### 3.1.2. Local generator

The local generator of HierarchicalPix follows a modified architecture of Pix2Pix [14]. It takes two inputs, a generated output of the global generator and our original input (satellite image). We identified that the use of a local generator helps in repatching some of the artifacts formed by the global generator thus improving the output quality.

Fig. 4 describes the architecture of the local generator used in HierarchicalPix. It also describes the number of feature channels in the output of each encoder and decoder node. The encoder and decoder blocks are similar to the one used in the global generator. The first encoder block ( $x_{0,0}$ ) does not apply InstanceNorm to its convoluted output and the bottleneck encoder block ( $x_{7,0}$ ) does not apply InstanceNorm and LeakyReLU to its convoluted output. The final decoder block ( $x_{0,1}$ ) does not apply InstanceNorm and LeakyReLU to its convoluted output, instead just applies Tanh. We have used skip connections to pass the feature map information between encoder, decoder.

Using both the global and local generators simultaneously in the HPix model offers several key advantages:

1. The global generator provides a coherent structure, while the local generator adds necessary detail. This balance ensures that the final output is both globally consistent and locally accurate.

2. The hierarchical approach allows the model to correct errors and refine details that may have been overlooked or inaccurately represented by the global generator alone. This leads to a reduction in artifacts commonly seen in single-level GAN approaches.
3. The combination of global and local processing results in higher overall accuracy, as the model can capture both the macro and micro aspects of the map. This is particularly important for applications where both broad structural integrity and fine detail are required.
4. The hierarchical design optimizes computational resources by first processing the image at a global level, thereby reducing the complexity that the local generator needs to handle. This staged approach makes the model more efficient without compromising quality.

### 3.1.3. Global and local discriminator

For the discriminator network, we used a  $26 \times 26$  PatchGAN [14]. Both the discriminator networks are identical to each other and are used to identify real and fake images for global and local generator respectively. Fig. 5 shows the architecture of the discriminators used in HierarchicalPix and Fig. 6 gives the structure of discriminator's CNNBlocks used in the network. The discriminator blocks consist of Convolution-InstanceNorm-LeakyReLU layers. The first layer (block 1)

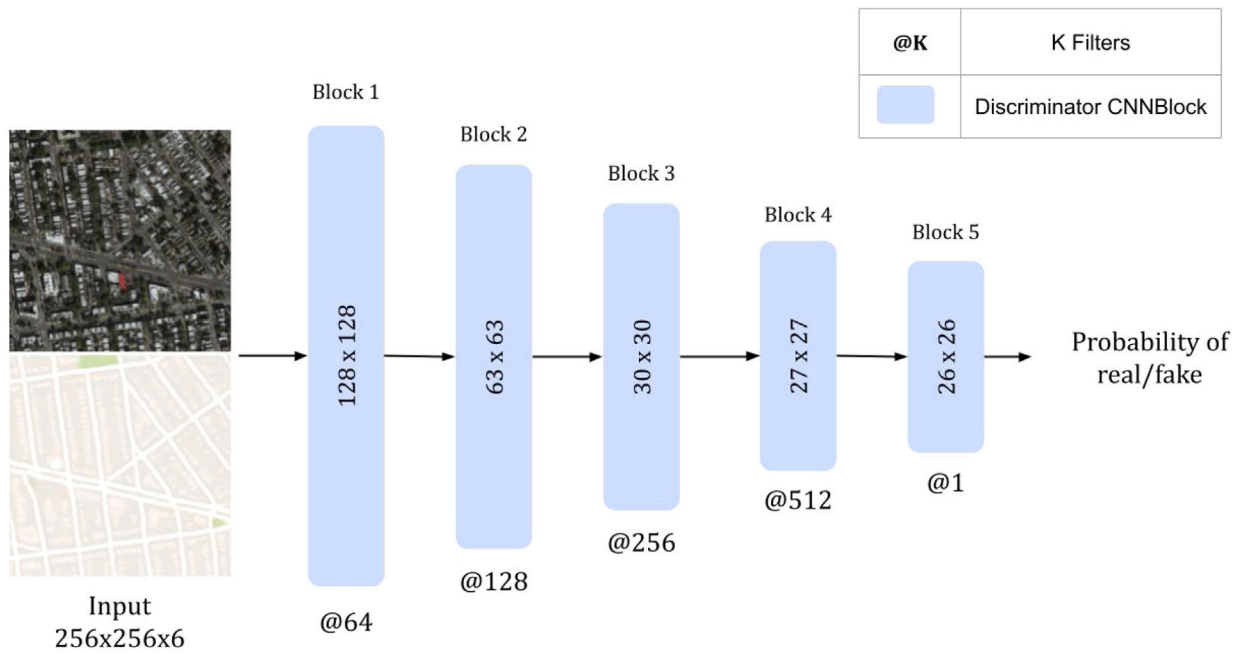


Fig. 5. PatchGAN discriminator architecture.

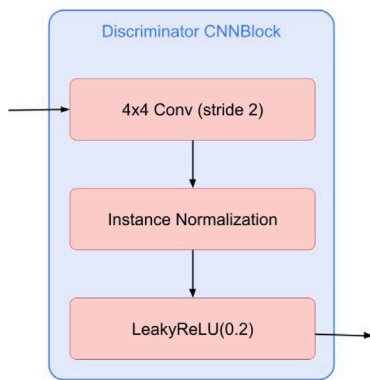


Fig. 6. Discriminator CNNBlock architecture.

does not have an InstanceNorm layer and we used LeakyReLU with a slope of 0.2. The last block, block 5, does not have an InstanceNorm and LeakyReLU layer and applies convolution with stride 1. The output from block 5 is a single channel output.

### 3.1.4. Objective function

In this approach, both our GAN models are conditional GANs and the objective of a conditional GAN can be expressed as

$$\mathcal{L}_{cGAN}(G, D) = \mathbb{E}_{x,y}[\log D(x, y)] + \mathbb{E}_{x,z}[1 - \log D(x, G(x, z))] \quad (1)$$

where G tries to minimize this objective against an adversarial D and D tries to maximize this objective against an adversarial G. So, for GAN at global level the objective function can be formulated as

$$\mathcal{L}_{global}(G, D_G) = \mathbb{E}_{x,y}[\log D_G(x, y)] + \mathbb{E}_{x,z}[1 - \log D_G(x, G(x, z))] \quad (2)$$

where G is the generator at global level,  $D_G$  is the discriminator at global level,  $x$  is the input image,  $y$  is the ground truth or target image and  $z$  is the random noise vector. And for GAN at local level the objective function can be formulated as

$$\mathcal{L}_{local}(H, D_H) = \mathbb{E}_{x,y}[\log D_H(x, y)] + \mathbb{E}_{x,z}[1 - \log D_H(x, H(x, G(x, z), z))] \quad (3)$$

where H is the generator at local level, G is the generator at global level,  $D_H$  is the discriminator at local level,  $x$  is the input image,  $y$  is the ground truth or target image and  $z$  is the random noise vector.

Pix2Pix [14] authors found that mixing the generator objective with a traditional loss like L1 distance helps generator to not just fool the discriminator but also bring the generated output near to ground truth and also generates less blurry output:

$$\mathcal{L}_{L1}(G) = \mathbb{E}_{x,y,z}[\|y - G(x, z)\|] \quad (4)$$

$$\mathcal{L}_{L1}(H) = \mathbb{E}_{x,y,z}[\|y - H(x, G(x, z), z)\|] \quad (5)$$

Our final objective functions are:

$$\mathcal{L}_{global}^*(G, D_G) = \arg \min_G \max_{D_G} \mathcal{L}_{global}(G, D_G) + \lambda \mathcal{L}_{L1}(G) \quad (6)$$

$$\mathcal{L}_{local}^*(H, D_H) = \arg \min_H \max_{D_H} \mathcal{L}_{local}(H, D_H) + \lambda \mathcal{L}_{L1}(H) \quad (7)$$

## 3.2. Feature extraction

In our research work, we are extracting two features, namely the road network and building clusters, from the satellite image. Subsequently, we utilize this data to identify road intersections and building clusters, which are then presented alongside a generated vector map to produce an interactive vector map. Further information regarding these tasks is elaborated in the subsequent subsection.

### 3.2.1. Road network and intersection

In this task, we would be extracting the road network from the satellite image and then using it to identify road intersections. For extracting road network we are a pretrained DLinkNet model trained on DeepGlobe Road Extraction Dataset [27]. The generated binary segmented map of the road network is then used with Algorithm 1 for identifying road intersections.

**Algorithm 1** Road Intersection Detection

```

1: procedure ROADINTERSECTION(Map)      ▷ Map: 3D array of pixels
2:   GrayscaleMap ← ConverttoGrayscale(Map)
3:   BinaryMap ← ConverttoBinary(GrayscaleMap)
4:   GaussianMap ← GaussianBlur(BinaryMap, kernel = (31 × 31))
5:   DilatedMap ← GaussianMap
6:   for i ← 0 to 5 do
7:     DilatedMap ← Dilate(DilatedMap, kernel = (3 × 3))
8:   end for
9:   ThresholdedMap ← Threshold(DilatedMap, threshold = 25)
10:  ErodedMap ← ThresholdedMap
11:  for i ← 0 to 5 do
12:    ErodedMap ← Erode(ErodedMap, kernel = (3 × 3))
13:  end for
14:  SkeletonMap ← Skeletonize(ErodedMap)
15:  CornerMap ← FindCorners(SkeletonMap)
16:  IntersectionMap ← DrawCorners(Map, CornerMap)
17:  return IntersectionMap ▷ IntersectionMap: 2D array of pixels
18: end procedure

```

3.2.2. Building clusters and classification based on size

In this task, we would be extracting building clusters from the satellite image and then use it to classify them based on sizes. For extracting building clusters, we would be training some of the popular segmentation models including Unet [26], Unet++ [24] and LinkNet [28] to generate a binary segmented map of building clusters and then use Algorithm 2 for classifying buildings in that segmented map. We choose these architectures because they have been proven to work well for this problem.

3.3. Generating interactive vector map

After identifying road intersections and classifying building clusters, this information is integrated with the base vector map generated from the satellite image. The process begins with the satellite image being passed through the trained HPix model, which outlines the primary road networks and major building clusters, providing a structural framework for the vector map. This initial output serves as the foundation for further refinement. The same satellite image is then processed through trained segmentation models to generate detailed road networks and building clusters. The local generator within HPix refines the coarse map by adding precise details, such as the exact shapes of buildings, the layout of small roads, and other intricate features. Algorithms 1 and 2 are subsequently employed to identify road intersections and classify building clusters. These refined details are then overlaid on the base vector map, resulting in an accurate and interactive vector map, suitable for high-stakes applications such as disaster response planning. Fig. 7 displays the general flow for generating interactive vector maps from the given satellite image.

4. Experiments and analysis

The conducted experiments were categorized into three segments. Initially, we successfully employed the HPix architecture to generate the vector tile map. The subsequent phase involved extracting various features, such as road networks and clusters of buildings, from the satellite image. This data was then utilized to identify road intersections and classify buildings based on their respective areas. Lastly, we amalgamated all the gathered information to create interactive vector maps. Further elaboration on each experiment will be provided in the upcoming sections.

4.1. Step1: Generating vector tile map using HPix

The first step can be further divided into four parts, dataset acquisition and preprocessing, evaluation metrics, experimental conditions, and performance evaluation and experimental results.

- **Dataset Acquisition and Preprocessing:** In this paper, for generating vector maps from GAN architectures we have conducted the experiments using publicly available maps dataset by Pix2Pix authors [14], which was later used by authors of [15,17,18] for their research. This dataset was collected from Google Maps and contains 1096 paired satellite and vector tile map images for training and 1098 paired satellite and vector tile map images for testing. The size of each satellite map and vector tile map image is 600 × 600. This dataset was also used by authors of [14,15,17,18] for training and testing their approach. Some examples of the maps dataset are shown in Fig. 8. For training and testing we resize the satellite and vector map image from 600 × 600 to 256 × 256 image. For training we also applied random jittering by first resizing the image to 286 × 286, then random cropping back to 256 × 256 sized image followed by horizontal flipping with a 50% probability. We have also normalized both satellite and vector map images before training and testing.
- **Evaluation Metrics:** To effectively compare our approach with existing models, we tested and compared output of our model on validation set using PSNR score, SSIM score and pixel level accuracy as metrics. We choose these metrics because they have been proven to be an effective way of comparing the quality of images being generated for this problem statement in [10,17,18].

1. **PSNR score:** PSNR stands for Peak Signal Noise Ratio and it is the most widely used quality metrics used for comparing two images and identifying how close they are based on intensity. It is generally used to check the quality of the generated image compared to its ground truth, and a large PSNR score means the generated image has similar intensity as the ground truth. PSNR is formulated as:

$$PSNR = 10 \times \log_{10} \left( \frac{MAX_I^2}{MSE} \right) \tag{8}$$

where:  $MAX_I$  maximum possible pixel value of the image  
 $MSE$  mean squared error between pixels of both images

Let there be two images,  $G$  for generated and  $A$  for actual, of size  $(m \times n)$  then MSE between these images can be formulated as:

$$MSE = \frac{1}{m \times n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} (A_{i,j} - G_{i,j})^2 \tag{9}$$

2. **SSIM score:** SSIM stands for Structural Similarity Index Measurement and is used to compare images based on structural similarity between them. It is used to calculate perceptual distance between the generated image ( $G$ ) and original image ( $A$ ) based on luminosity (mean), contrast (variance) and structure (covariance) of both images. It lies between 0 and 1, and two same images have a SSIM score of 1. SSIM is formulated as:

$$SSIM(A, G) = l(A, G) \cdot c(A, G) \cdot s(A, G) \\ = \left( \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \right) \cdot \left( \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2} \right) \cdot \left( \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3} \right) \tag{10}$$

where:  $\mu_x$  Mean of pixel values of  $x$   
 $\mu_y$  Mean of pixel values of  $y$



**Algorithm 2** Building Classification

```

1: procedure BUILDINGCLASSIFIER(Map, Resolution, Thresholds, ColorLabels) ▷ Map: 2D array of pixels, Resolution: meters per pixel, Thresholds:
   small, medium, and large building sizes, ColorLabels: small, medium, and large building colors
2:   Counters ← FindContours(Map)
3:   Areas ← CalculateAreas(Counters, Resolution)
4:   for i ← 0 to len(Areas) do
5:     if Areas[i] < SmallThreshold then
6:       Labels ← DrawContours(Map, Counters[i],1)
7:     else if Areas[i] < MediumThreshold then
8:       Labels ← DrawContours(Map, Counters[i],2)
9:     else
10:      Labels ← DrawContours(Map, Counters[i],3)
11:    end if
12:  end for
13:  ClassifiedMap ← RecolorMap(Labels, ColorLabels)
14:  return ClassifiedMap
15: end procedure

```

▷ *ClassifiedMap*: 2D array of pixels

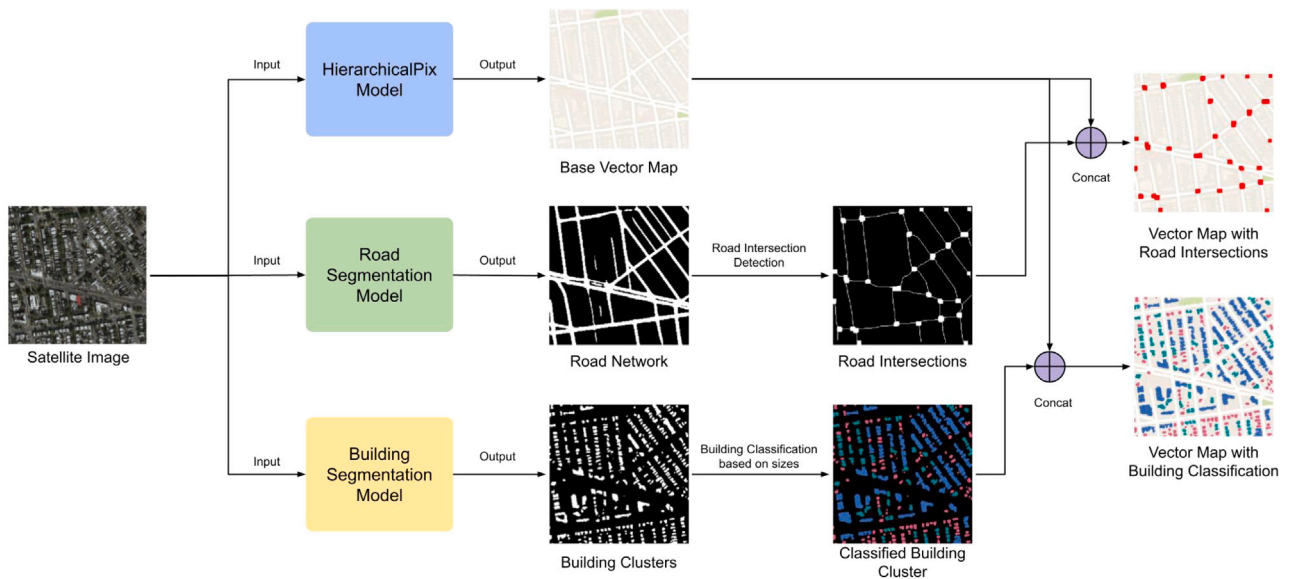


Fig. 7. Workflow of our approach.

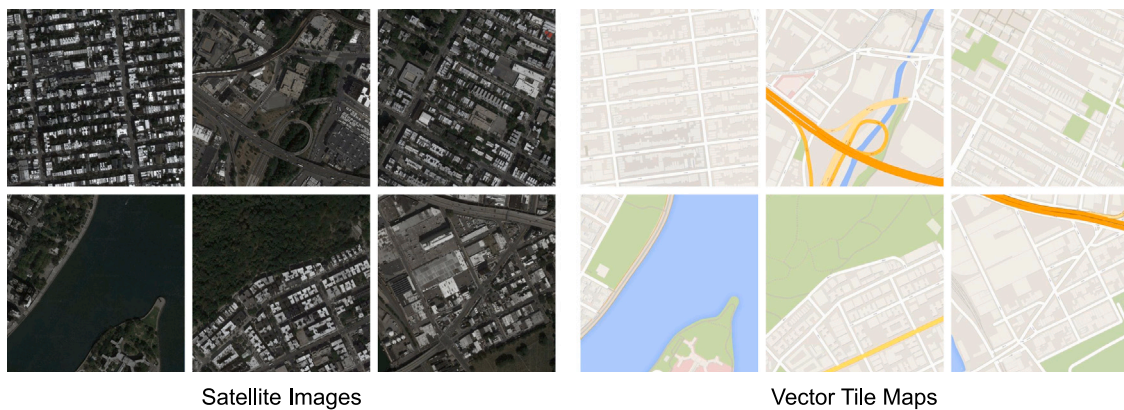


Fig. 8. Some samples from maps dataset.

**Table 1**  
Comparative analysis of different models.

Model name	Pixel level accuracy	SSIM score	PSNR score
Pix2Pix	42.09%	0.64	25.29
CycleGAN	36.47%	0.63	24.05
MapGen-GAN <sup>a</sup>	38.54%	0.64	24.64
CscGAN <sup>a</sup>	46.86%	0.73	<b>27.20</b>
HierarchicalPix (Proposed)	<b>61.04%</b>	<b>0.75</b>	26.98

Best results are highlighted in bold.

<sup>a</sup> Results for this model are obtained from their respective research paper.

- $\sigma_{x^2}$  Variance of  $x$
- $\sigma_{y^2}$  Variance of  $y$
- $\sigma_{xy}$  Covariance of  $x$  and  $y$
- $C_1, C_2, C_3$  Constants

3. **Pixel level accuracy:** This metric is used to find the accuracy of generated output by finding the accurate number of correctly translated pixels in the generated output with respect to ground truth and then taking its average. This metric was used by authors of [17,18,29] to find their model accuracy. Let us say a pixel from a generated image can be represented as  $G(r_i, b_i, g_i)$  and a pixel from the ground truth is represented as  $A(R_i, B_i, G_i)$  then pixel level accuracy can be formulated as:

$$pixelacc = \begin{cases} 1 & \text{if } \max(|R_i - r_i|, |G_i - g_i|, |B_i - b_i|) \leq \gamma \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Authors of [17,18,29] have used  $\gamma$  as 5 because colors may seem similar but may vary slightly at pixel level and this strategy can effectively counter that problem.

- **Experimental setups:** The training of models were performed on the Kaggle platform by using community-available two Nvidia Tesla T4 GPUs with 13 GB RAM and 2 CPU cores. It took around 16 h to train the model and around 10 min to validate the trained model. The code was written using Pytorch library in python. While training HierarchicalPix, we have trained both generators simultaneously so that they could learn and generalize the problem together. We used Adam optimizer for both the generators and discriminators with a learning rate of 0.0002 and beta1 and beta2 as 0.5 and 0.999. We trained the model on objective function defined in methodology for 200 epochs.
- **Performance Evaluation:** For comparing our model performance, we are considering Pix2Pix model [14], CycleGAN model [15] as our baseline. We are also considering some of the latest implementations including CscGAN [18] and MapGen-GAN [17]. These models have been proven to perform better than their predecessors. The comparison of our approach is shown in Table 1. From the experimental results we can conclude that our approach gave better performance on most of the metrics when compared with other models and the PSNR score of our approach was almost comparable to CscGAN (current best). In this research, we have trained Pix2Pix and CycleGAN from scratch under the same environment along with our proposed method and compared its performance. We have also used the results of CscGAN [18] and MapGen-GAN [17] to further compare our model with their approach. Fig. 9 displays the comparison between output generated by PixPix, CycleGAN and our approach and our model generates better results compared to them. Fig. 10 displays how use of the local generator helped in patching up the artifacts generated by the global generator and improving the output quality.

#### 4.2. Step2: Extracting features from the satellite image

We have also performed two feature-extracting tasks from the satellite image, road network extraction and building cluster extraction. We have used road network data to identify road intersections and used building cluster data to classify buildings based on the area covered. To extract the road network, we have used a pre-trained DLinkNet network while to extract building clusters we have trained and compared different segmentation models from scratch.

Features extraction is divided into four subsections, dataset acquisition and preprocessing, evaluation metrics, experimental conditions, and performance evaluation and experimental results. The first three subsections provides details about building extraction task while the final subsection provides results about both tasks. More details about the experiment will be explained in the following subsections.

- **Dataset acquisition and preprocessing:** For generating the binary segmentation map of the road network and building cluster map from the satellite image, we have used the ability of deep learning models to generalize the solution of a problem. While we are using pretrained model to extract road network, we have trained different popular segmentation models on the Massachusetts building dataset provided by authors of [25] and have compared them to choose the best model for building cluster identification. The Massachusetts building dataset consists of 151 aerial images of the Boston area split as 137 images for training, 10 images for testing and 4 images for validation. The size of images in both dataset is  $1500 \times 1500$ . Samples of images from these dataset are displayed in Fig. 11.
- For maintaining the similarity in size between the generated vector map and building cluster map, we have first randomly cropped the image from  $1500 \times 1500$  size to  $600 \times 600$  size and then resized it to  $256 \times 256$  for training. While testing, we have cropped a  $600 \times 600$  from the upper left corner and then resized it to  $256 \times 256$ .
- **Evaluation metrics:** For testing and comparing the output of our trained models we have used two metrics, IoU score and dice score.

1. **IoU score:** IoU score stands for Intersection over Union score and it is a very popular evaluation metric used to measure degree of overlap between the predicted image and actual image. The range of this score is [0, 1] with 1 indicating a perfect overlap and 0 indicating no overlap. The IoU score can be formulated as:

$$IoU\ score = \frac{TP}{TP + FP + FN} \quad (12)$$

where, TP, FP and FN represent True Positive, False Positive, and False Negative respectively. Here, TP, FP and FN are calculated based on the number of correct pixels marked in predicted segmentation.

2. **Dice score:** It is a commonly used metric for measuring the similarity between two images based on the overlap between the predicted image and actual image. It is a very popular metric which provides a good balance between sensitivity and specificity. This score lies between [0, 1], with a score of 1 indicating a perfect overlap and 0 indicating no overlap. The dice score can be formulated as:

$$Dicescore = \frac{2TP}{2TP + FP + FN} \quad (13)$$

where, TP, FP and FN represent True Positive, False Positive, and False Negative respectively.

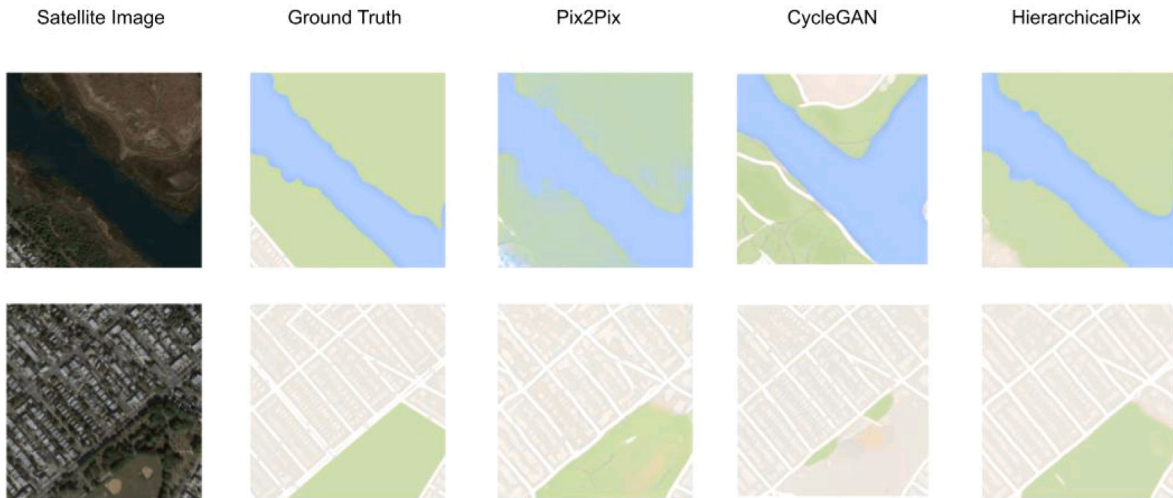


Fig. 9. Visualization of image generated by different methods on maps dataset.

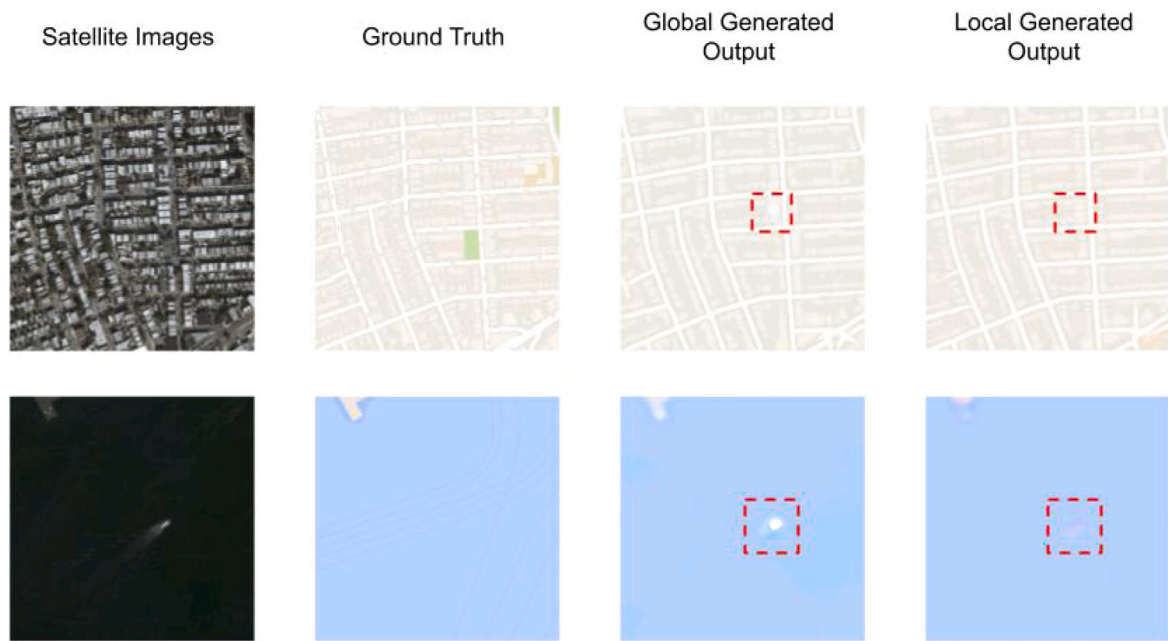


Fig. 10. Visualization of image generated by the global and local generator and how local generator helps in patching some of the artifacts generated by global generator.

- *Experimental setups:* We have used Unet [26], Unet++ [24] and LinkNet [28] architectures for building segmentation task and these implementation were carried out using Pytorch library in python. The training of these models were carried out on the kaggle platform by using community-available two Nvidia Tesla T4 GPUs with 13 GB RAM and 2 CPU cores. We have trained each model for 200 epochs and save the best model on validation set and it took around 3 to 4 h to train each model. The models were trained on a batch size of 8 and validated on batch size of 1. We have used Adam optimizer with learning rate of 0.0002 and binary cross entropy with logits loss to train the model.
- *Performance evaluation:* We have trained and tested Unet, Unet++ and LinkNet models for building clusters identification, and choose the best among them for our task of separating buildings

based on sizes. The comparison between these models is shown in Table 2. We have used Unet++ model and the pretrained model with the algorithm described in the methodology and Figs. 12 and 13 shows our experiment results. In Fig. 12, we have identified road network using pretrained model and highlighted the road intersection points with the general connectivity of the road network in the final representation. In Fig. 13, we have identified building clusters using the trained Unet++ model and separated the buildings identified based on area. We have considered the satellite image to be as 1 meter per pixel resolution and highlighted building with area between 0 and 250 square meters as red, area between 250 and 500 square meters as green and any building with area above 500 square meters as blue.



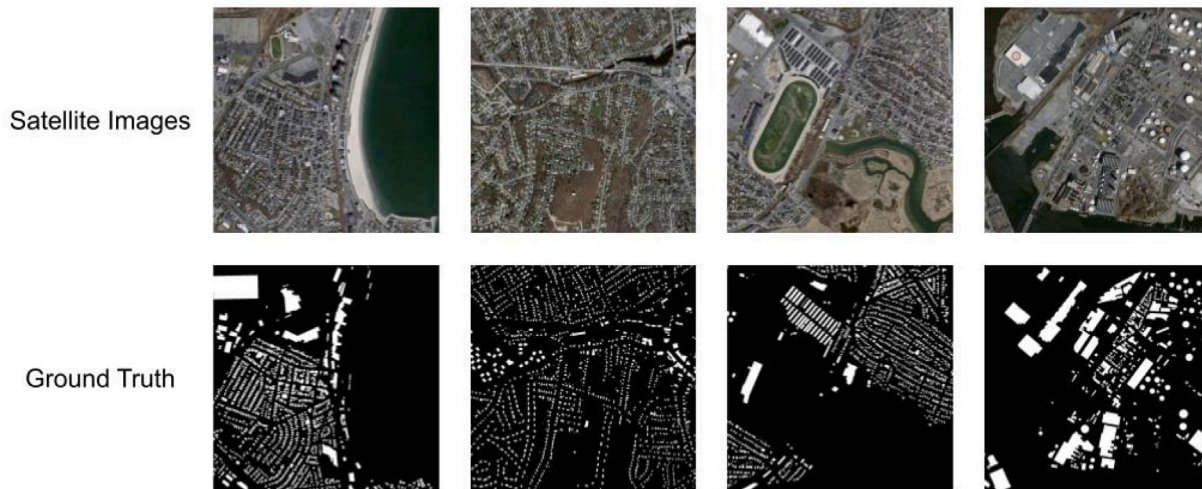


Fig. 11. Some samples from Massachusetts building dataset.

**Table 2**  
Comparative analysis of different models for building cluster.

Model name	IoU score	Dice score
Unet	0.5301	0.6929
Unet++	<b>0.5820</b>	<b>0.7358</b>
LinkNet	0.5201	0.6843

Best results are highlighted in bold.

**Table 3**  
Comparative Analysis of Different Models Across Multiple Datasets.

Model	PSNR (dB)	SSIM	IoU (Roads)	IoU (Buildings)
Pix2Pix	25.29	0.64	0.52	0.69
CycleGAN	24.05	0.63	0.49	0.65
MapGen-GAN	24.64	0.64	0.51	0.68
CscGAN	27.20	0.73	0.54	0.72
U-Net	28.15	0.74	0.56	0.75
DenseGAN	27.80	0.76	0.55	0.73
ResNet-based GAN	28.50	0.77	0.57	0.76
<b>HPix (Proposed)</b>	<b>29.10</b>	<b>0.78</b>	<b>0.60</b>	<b>0.80</b>

### 4.3. Generating interactive vector map

Following the identification of road intersections and the classification of building clusters according to their areas, we integrated this data with the generated vector map. The resulting interactive vector maps are showcased in Fig. 14, illustrating road intersections and the categorized buildings on the generated vector map.

To ensure the effectiveness of our proposed model, we have done a in depth comprehensive analysis of the performance of our proposed HPix model with respect to various state-of-the-art models. We used several datasets that cover diverse geographic features and complexities:

- **Massachusetts Buildings Dataset**[30]: Used for evaluating building footprint extraction and classification.
- **DeepGlobe Road Extraction Dataset**[31]: Focused on assessing road network extraction and mapping accuracy (see Table 3).

The following table summarizes the performance of HPix compared to other models across the different datasets:

The results indicate that HPix outperforms other models across all metrics and datasets. HPix achieves the highest PSNR, indicating that the vector maps it creates have the best image quality overall. Additionally, HPix has the top SSIM score. This indicates that the similarity in structure between its generated maps and the actual ground truth is better than that of other models. Moreover HPix performs exceptionally well in IoU scores for road networks and building footprints. This highlights its effectiveness in accurately capturing specific features in satellite images.

The key reason for HPix’s outstanding performance is its hierarchical design. It uses both global & local generators. The global generator captures large-scale structures accurately, while the local generator focuses on refining smaller details, which helps reduce artifacts and boost the overall quality of the created maps. This two-tiered design strikes a perfect balance between high-level structural integrity and detailed precision—both are crucial for important areas like urban planning and disaster response.

### 4.4. Ablation study

To check how well each part of the proposed HPix model works, an ablation study was carried out. This study’s aim is to understand the effect of each generator within the HPix architecture. We focused on the Global Generator and the Local Generator—with a specific eye on how they influence the quality of the final vector map. Also, we looked at how deep supervision and advanced loss functions help boost model performance.

We designed the ablation study by incrementally disabling or simplifying different parts of the HPix model. Then we observed the performance changes in accuracy, structural similarity, & computational efficiency. The following configurations were tested:

- **Full HPix Model:** The complete HPix structure incorporates both the Global & Local Generators. The Global Generator employs U-Net++, while the Local Generator utilizes a modified Pix2Pix architecture. Also included deep supervision techniques and advanced loss functions (Dice Loss and IoU).
- **Without Local Generator:** A simplified version of HPix, where just the Global Generator works. No refinement is done by the Local Generator in this setup.
- **Without Global Generator:** In this case, only the Local Generator operates. It takes the original satellite image directly as input, without the coarse map typically supplied by the Global Generator.



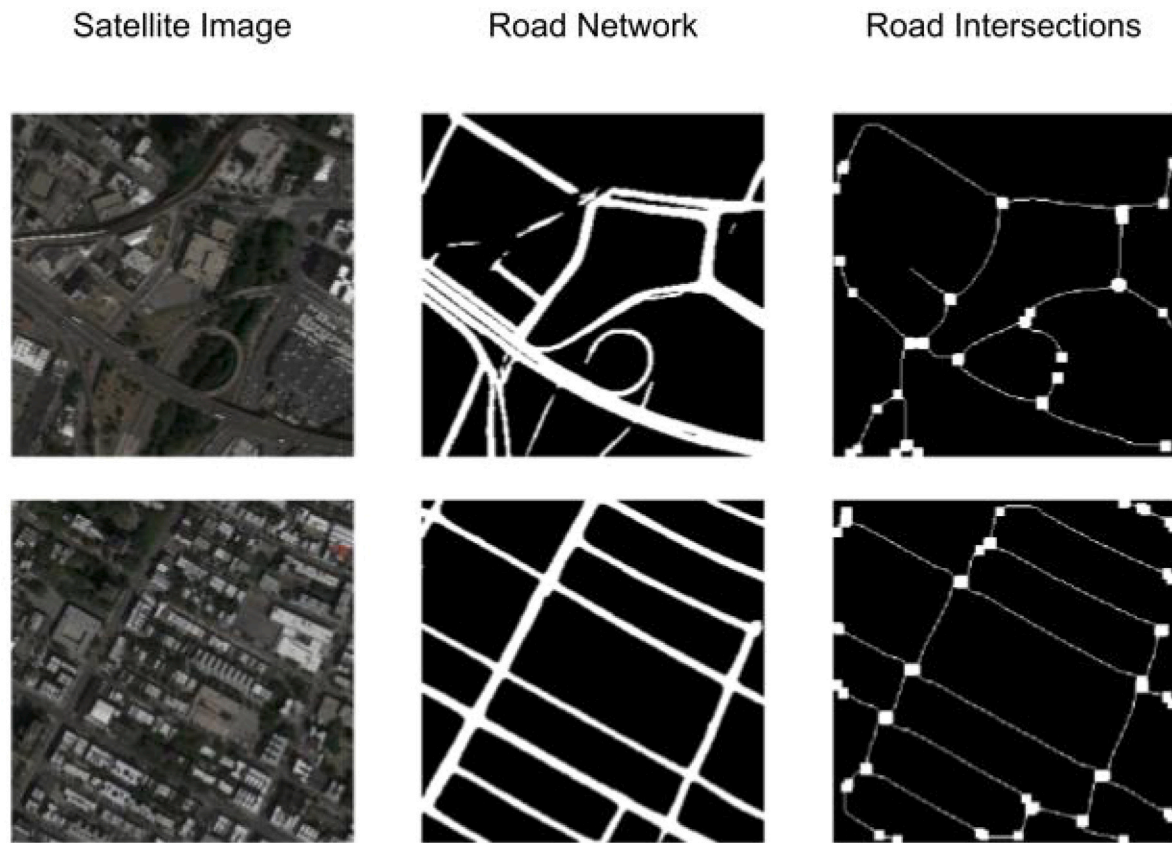


Fig. 12. Road network using pretrained model and intersection sample output.

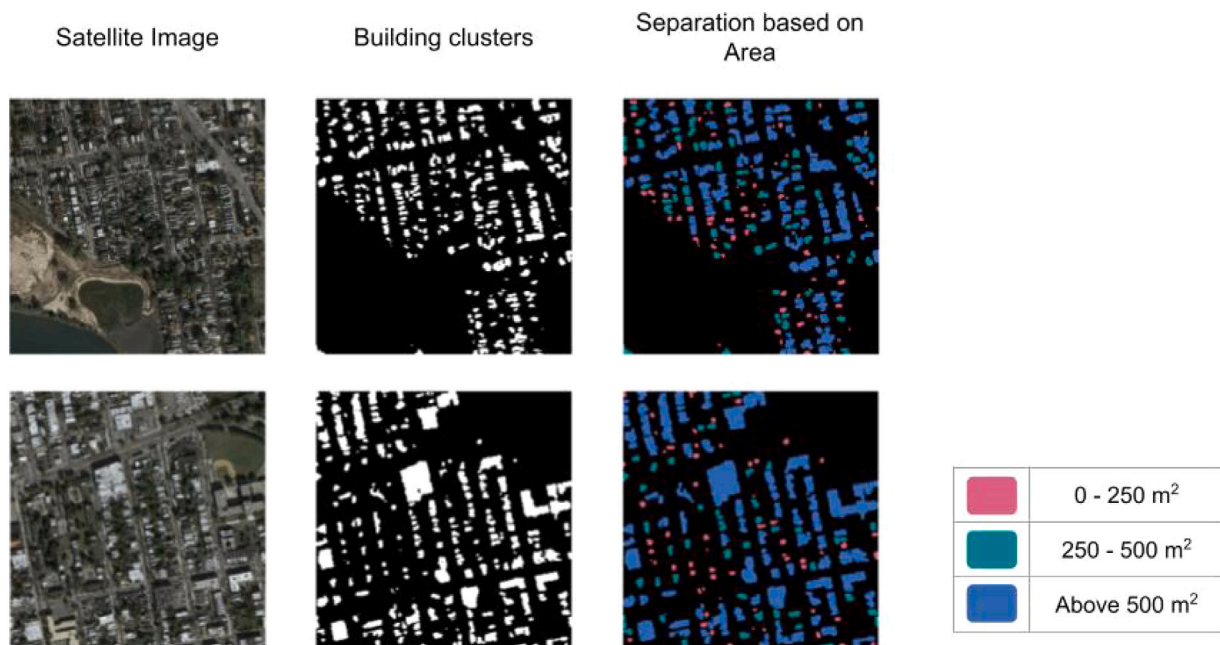


Fig. 13. Building cluster identification using trained Unet++ model and classification based on area.

- **Without Deep Supervision:** The full HPix model but without deep supervision applied in the Global Generator.
- **Without Advanced Loss Functions:** The full HPix model but using a basic cross-entropy loss function instead of the Dice Loss and IoU.

#### 4.4.1. Results and analysis

The results of the ablation study are summarized in Table 4. This table compares how each model configuration performed using three key metrics: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Measure (SSIM), & Intersection over Union (IoU) for building footprints and road networks.



Fig. 14. Some samples of generated vector map with marked road intersection and classified building clusters.

Table 4

Ablation Study Results: Performance of Different HPix Configurations.

Model configuration	PSNR (dB)	SSIM	IoU
Full HPix Model	29.10	0.78	0.80
Without Local Generator	27.85	0.72	0.75
Without Global Generator	27.10	0.70	0.72
Without Deep Supervision	28.30	0.74	0.77
Without Advanced Loss Functions	27.65	0.71	0.73

The ablation study reveals several important insights:

- **Importance of the Local Generator:** Eliminating the Local Generator leads to a noticeable decrease in performance across every metric. This is especially true for IoU, which underscores how vital it is for enhancing the coarse map created by the Global Generator.
- **Role of the Global Generator:** The Global Generator is key in maintaining the structural integrity of the vector map. when the Local Generator operates separately from the Global Generator, there is a significant drop in performance. This suggests that the coarse map gives important context that aids the Local Generator during its refinement process.
- **Effect of Deep Supervision:** The lack of deep supervision in the Global Generator results in a slight dip in the model’s accuracy. This shows that deep supervision plays a significant role in stabilizing the learning process as well as boosting overall performance.
- **Impact of Advanced Loss Functions:** When we use basic cross-entropy loss than Dice Loss & IoU results in a noticeable decrease in IoU scores. This finding highlights that advanced loss functions are crucial for securing high accuracy in segmentation tasks, especially within complex urban settings.

Our ablation study proves how effective each part of the HPix model is. The Global & Local Generators are vital for creating high-quality vector maps. The Local Generator adds necessary detail refinement, while the Global Generator makes sure the structure remains coherent. Furthermore, deep supervision along with advanced loss functions is essential for improving the model’s performance, particularly regarding accuracy and detail retention.

#### 4.5. Computational complexity

The computational complexity of the HPix model is a critical aspect that determines its feasibility for real-world applications, particularly in scenarios requiring the processing of large-scale satellite imagery.

##### 4.5.1. Time complexity

The time complexity of the HPix model can be analyzed by considering the two main components: the global generator and the local generator.

- **Global Generator:** The global generator is built on a U-Net++ architecture, which incorporates a sequence of convolutions, pooling, and upsampling processes. The time complexity for each convolutional layer is  $O(K \times K \times C_{in} \times C_{out} \times W \times H)$ , where  $K$  denotes the kernel size,  $C_{in}$  and  $C_{out}$  represent the input and output channels, and  $W$  and  $H$  indicate the width and height of the input feature map. Due to the hierarchical and nested design of U-Net++, the overall time complexity is heightened by deep supervision and dense inter-layer connections; however, this is balanced by improvements in accuracy and feature representation.
- **Local Generator:** The local generator, which uses a modified Pix2Pix architecture, has a complexity structure that is similar in time. It focuses on smaller, localized areas, so the computation needed is generally less than what the global generator requires. The local generator’s complexity mainly hinges on how many fine details need processing. Fortunately, it benefits from a hierarchical approach. This means that the coarse map made by the global generator helps narrow down the search for the local generator.

##### 4.5.2. Comparisons to other models

In contrast to traditional models like U-Net, Pix2Pix, and CycleGAN, HPix shows greater computational complexity. This is primarily due to its hierarchical design & the use of advanced architectures such as U-Net++. Still, this rise in complexity is well worth it because it leads to notable gains in accuracy and better feature representation.

- **U-Net:** The standard U-Net is known for its lower computational complexity. However, it faces with large-scale satellite imagery. This happens because it struggles to capture fine details across different scales effectively. HPix addresses these issues by using a dual approach, but this comes with increased complexity.
- **Pix2Pix and CycleGAN:** These models are quite efficient when it comes to general image-to-image translation tasks. Still, they fall short compared to HPix due to the lack of a hierarchical structure. This absence means they do not perform as well in scenarios needing both global structural integrity and detailed precision. On the other hand, HPix employs a global as well as local generator which does add to its computational demands but leads to much better performance in creating and accurate vector maps.

#### 4.5.3. Trade-offs and practical considerations

HPix's greater computational complexity could be viewed as a downside. Yet, this is a required trade-off to gain better accuracy & detail in the vector maps it generates. When it comes to tasks needing very high precision—think urban planning or disaster management—the advantages of HPix clearly outweigh the extra computational costs. Nevertheless, for tasks that are not as demanding, or in situations with scarce computational resources, simpler models like U-Net or Pix2Pix may be a better fit.

## 5. Conclusion

In this paper, we have proposed a novel method for generating vector tile map from satellite image termed HierarchicalPix (HPix). This architecture comprises of two generators, global and local, for identifying complex features in the input image and map it with ground truth. We have also found that using local level generator helps in reducing the number of artifacts in the generated output, thus improving the generated output quality. The experimental results show that our model HPix outperforms existing algorithms by employing a hierarchical GAN framework that captures both global structures and local details, resulting in higher accuracy and fewer artifacts. Its versatility across various image-to-image translation tasks makes it a robust tool for applications like urban planning and disaster response.

### Code availability

The code is available at the GitHub repository. [Link: <https://github.com/aditya-taparia/Satellite-Image-to-Vector-Map>]

### CRedit authorship contribution statement

**Aditya Taparia:** Conceptualization, Software, Writing – original draft. **Ali Kashif Bashir:** Investigation, Visualization, Writing – original draft. **Yaodong Zhu:** Data curation, Project administration, Supervision, Writing – review & editing. **Thippa Reddy Gdekallu:** Resources, Supervision, Visualization, Writing – review & editing. **Keshab Nath:** Conceptualization, Methodology, Writing – original draft.

### Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Data availability

We have used publicly available maps dataset in this study. This data can be found here: <http://efrogans.eecs.berkeley.edu/pix2pix/datasets/maps.tar.gz>.

## Acknowledgment

The work by Yaodong Zhu is supported by Zhejiang Key Research and Development Project under Grant 2017C01043.

## References

- [1] J. Xu, G. Zhou, S. Su, Q. Cao, Z. Tian, The development of a rigorous model for bathymetric mapping from multispectral satellite-images, *Remote Sens.* 14 (10) (2022) 2495.
- [2] Q. Chen, L. Yang, Y. Zhao, Y. Wang, H. Zhou, X. Chen, Shortest path in LEO satellite constellation networks: An explicit analytic approach, *IEEE J. Sel. Areas Commun.* (2024).
- [3] S. Iino, R. Ito, K. Doi, T. Imaizumi, S. Hikosaka, CNN-based generation of high-accuracy urban distribution maps utilising SAR satellite imagery for short-term change monitoring, *Int. J. Imag. Data Fusion* 9 (4) (2018) 302–318.
- [4] J. Hormese, C. Saravanan, Automated road extraction from high resolution satellite images, *Proc. Technol.* 24 (2016) 1460–1467.
- [5] M. Sahu, A. Ohri, Vector map generation from aerial imagery using deep learning, *ISPRS Ann. Photogramm. Remot. Sens. Spatial Inf. Sci.* 4 (2019) 157–162.
- [6] S. Ganguli, P. Garzon, N. Glaser, Geogan: A conditional GAN with reconstruction and style loss to generate standard layer of maps from satellite images, 2019, arXiv preprint arXiv:1902.05611.
- [7] H. Mansourifar, A. Moskowitz, B. Klingensmith, D. Mintas, S.J. Simske, GAN-based satellite imaging: A survey on techniques and applications, *IEEE Access* (2022).
- [8] H. Xu, Q. Li, J. Chen, Highlight removal from a single grayscale image using attentive GAN, *Appl. Artif. Intell.* 36 (1) (2022) 1988441.
- [9] J. Chen, Y. Song, D. Li, X. Lin, S. Zhou, W. Xu, Specular removal of industrial metal objects without changing lighting configuration, *IEEE Trans. Ind. Inform.* (2023).
- [10] Y. Pang, J. Lin, T. Qin, Z. Chen, Image-to-image translation: Methods and applications, *IEEE Trans. Multimed.* 24 (2021) 3859–3881.
- [11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks, *Commun. ACM* 63 (11) (2020) 139–144.
- [12] L. Yin, L. Wang, J. Li, S. Lu, J. Tian, Z. Yin, S. Liu, W. Zheng, YOLOV4\_CSPBi: enhanced land target detection model, *Land* 12 (9) (2023) 1813.
- [13] M. Mirza, S. Osindero, Conditional generative adversarial nets, 2014, Preprint at <https://arxiv.org/abs/1411.1784>.
- [14] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134.
- [15] J.-Y. Zhu, T. Park, P. Isola, A.A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, in: *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2223–2232.
- [16] V. Ingale, R. Singh, P. Patwal, Image to image translation: Generating maps from satellite images, 2021, Preprint at <https://arxiv.org/abs/2105.09253>.
- [17] J. Song, J. Li, H. Chen, J. Wu, MapGen-GAN: a fast translator for remote sensing image to map via unsupervised adversarial learning, *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* 14 (2021) 2341–2357.
- [18] Y. Liu, W. Wang, F. Fang, L. Zhou, C. Sun, Y. Zheng, Z. Chen, Cscgan: Conditional scale-consistent generation network for multi-level remote sensing image to map translation, *Remote Sens.* 13 (10) (2021) 1936.
- [19] Z. Li, Y. Wang, R. Zhang, F. Ding, C. Wei, J.-G. Lu, A lidar-OpenStreetMap matching method for vehicle global position initialization based on boundary directional feature extraction, *IEEE Trans. Intell. Veh.* (2024).
- [20] M. Mokhtarzade, M.V. Zojer, Road detection from high-resolution satellite images using artificial neural networks, *Int. J. Appl. Earth Obs. Geoinf.* 9 (1) (2007) 32–40.
- [21] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: *Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III* 18, Springer, 2015, pp. 234–241.
- [22] H. Xu, H. He, Y. Zhang, L. Ma, J. Li, A comparative study of loss functions for road segmentation in remotely sensed road datasets, *Int. J. Appl. Earth Obs. Geoinf.* 116 (2023) 103159.
- [23] W. Alsabhan, T. Alotaiby, Automatic building extraction on satellite images using unet and resnet50, *Comput. Intell. Neurosci.* 2022 (2022).
- [24] Z. Zhou, M.M. Rahman Siddiquee, N. Tajbakhsh, J. Liang, Unet++: A nested u-net architecture for medical image segmentation, in: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: 4th International Workshop, DLMIA 2018, and 8th International Workshop, ML-CDS 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 20, 2018, Proceedings 4*, Springer, 2018, pp. 3–11.
- [25] V. Mnih, *Machine Learning for Aerial Image Labeling*, University of Toronto (Canada), 2013.



- [26] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5–9, 2015, Proceedings, Part III 18, Springer, 2015, pp. 234–241.
- [27] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, R. Raskar, DeepGlobe 2018: A challenge to parse the earth through satellite images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2018, pp. 172–181.
- [28] A. Chaurasia, E. Culurciello, Linknet: Exploiting encoder representations for efficient semantic segmentation, in: 2017 IEEE Visual Communications and Image Processing (VCIP), IEEE, 2017, pp. 1–4.
- [29] H. Fu, M. Gong, C. Wang, K. Batmanghelich, K. Zhang, D. Tao, Geometry-consistent generative adversarial networks for one-sided unsupervised domain mapping, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2427–2436.
- [30] V. Mnih, Machine Learning for Aerial Image Labeling (Ph.D. thesis), University of Toronto, 2013.
- [31] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, R. Raskar, DeepGlobe 2018: A challenge to parse the earth through satellite images, in: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2018.