IUST
Iran University of Science and Technology

Iranian Journal of Electrical and Electronic Engineering

Journal Homepage: ijeee.iust.ac.ir

Enhancing privacy by large mask inpainting and fusion-based segmentation in street view imagery

M. Khouri Shandiz*, A. Amirkhani*(C.A.)

Abstract: Protecting privacy in street view imagery is a critical challenge in urban analytics, requiring comprehensive and scalable solutions beyond localized obfuscation techniques such as face or license plate blurring. To address this, we propose a novel framework that automatically detects and removes sensitive objects, such as pedestrians and vehicles, ensuring robust privacy preservation while maintaining the visual integrity of the images. Our approach integrates semantic segmentation with 2D priors and multimodal data from cameras and LiDAR to achieve precise object detection in complex urban scenes. Detected regions are seamlessly filled using a large-mask inpainting technique based on fast Fourier convolutions (FFC), enabling efficient generalization to high-resolution imagery. Evaluated on the SemanticKITTI dataset, our method achieves a mean Intersection over Union (mIoU) of 64.9%, surpassing state-of-the-art benchmarks. Despite its reliance on accurate sensor calibration and multimodal data availability, the proposed framework offers a scalable solution for privacy-sensitive applications such as urban mapping, and virtual tourism, delivering high-quality anonymized imagery with minimal artifacts.

Keywords: Privacy Protection, Street View Imagery, Large Mask Inpainting, Semantic Segmentation, Multi-modality, Lidar.

1 Introduction

Street view images (SVI), which are obtained from sources such as Google Street View (GSV), Here Map Street View, Baidu Street View, Mapillary Street View, Tencent Street View, etc., are vital tools for studying and understanding different regions of the world. Among these sources, GSV has a wide coverage in 114 countries, which broadly displays the different regions of these countries. By providing comprehensive images from different parts of the world, this street view provides unique information for researchers, travelers, and those interested in spaces and cities [1]. In recent years, GSV has been the largest and perhaps the most well-known collection of street-level images collected. The GSV service was first rolled out in 2007 as an experiment in some cities in the United States [2]. After that, it was developed more widely around the world in the following years. This service provides the possibility to view panoramic images of different streets and passages and is now available in many cities and regions around the world. This service enables users to efficiently search for and locate their points of interest. Additionally, it facilitates virtual tours of the street-level environment, facilitating a diverse range of applications such as real estate search, virtual tourism, travel planning, driving routes, and more [3]. GSV is a useful and highly popular service. However, it raises significant privacy concerns.

Iranian Journal of Electrical & Electronic Engineering, YYYY. Paper first received DD MONTH YYYY and accepted DD MONTH YYYY.
* The authors are with the School of Automotive Engineering, Iran University of Science and Technology (IUST), Tehran 16846-13114, Iran.
E-mails: amirkhani@iust.ac.ir.
Corresponding Author: Abdollah Amirkhani

The images captured from the street level contain numerous personally identifiable features, such as faces and license plates [4]. To tackle this challenge, Google has introduced a sliding window-based system aimed at automatically blurring faces and license plates in street view images [3]. Although this work reduces the concerns related to the disclosure of information and the privacy of people, still information such as clothing items, colors, patterns, body shape, height size, geographic location information, and also vehicles in the images that information such as reveals the type and color of the vehicle, etc. To address this issue, we propose an automated method to remove all pedestrians and vehicles from street view images. Our proposed method uses semantic segmentation to detect vehicles and pedestrians, and after detection, it removes them and fills the large-scale gaps. For semantic segmentation, we suggest employing 2D Priors (2DPASS) [5], a method tailored to improve 3D LiDAR semantic segmentation by integrating insights from 2D priors derived from cameras. For semantic segmentation, 2DPASS combines multi-modal information into a single point cloud. Additionally, given the presence of significant masks in the images, we recommend employing large mask inpainting (LaMa) [6]. The method is suggested for image inpainting, highlighting its capability to generalize to high-resolution images despite being trained solely on low-resolution data.

In sum, we make four key claims, which are supported by our experimental findings: (i) Our framework integrates multimodal data from LiDAR and cameras, enhanced by a knowledge distillation strategy, to achieve state-of-the-art segmentation accuracy for pedestrians and vehicles in challenging urban environments with complex lighting and occlusions. (ii) Using an inpainting method based on fast Fourier Convolutions, our approach effectively fills large-scale gaps created by object removal, achieving high-quality reconstructions in high-resolution street view images while maintaining computational efficiency. (iii) Unlike prior methods that focus on localized obfuscation (e.g., face or license plate blurring), our method provides a comprehensive solution by removing all sensitive objects, ensuring robust privacy protection while preserving the visual integrity of the images. (iv) The proposed algorithm is scalable to large-scale datasets, including street view imagery, and achieves an average inference time of less than 600 milliseconds per image. It operates efficiently with minimal computational overhead on standard hardware, such as an NVIDIA Tesla T4 GPU.

2 Related work

2.1 Detection and removal of objects

To remove unwanted areas, it is necessary to identify them first. This operation is known as region of interest (ROI) detection [7]. Following the detection of regions of interest, the subsequent task involves their removal and background filling. Recent strides in deep learning-based object detection methods have showcased strong performance in ROI detection, affirming their effectiveness in this domain. Multi-layer neural networks such as CNN are designed to directly recognize visual patterns in pixels, and powerful architectures such as ResNet [8] and Xception [9] have been developed. At the same time, with the advancement of CNN technology, object detection algorithms based on CNN have also been developed and models like You Only Look Once (YOLO) [10] and DeepLab v3+ [11] have been presented. In these methodologies, the initial step involves identifying desired objects and areas based on their contours and distinctive features through image recognition and segmentation algorithms. However, with CNN-based object detection for ROI determination, there's no requirement to define a specific target object policy for the ROI. Instead, we're dealing with a mask that completely covers the desired subject. However, acquiring background images devoid of moving obstacles presents a challenge in practical projects. For this purpose, efforts are made to obtain accurate and flawless background images by using various data and algorithms. Also, in the field of segmentation of objects, by using several sensors like LiDAR and cameras, efforts are made to combine the information and benefit from the advantages of each, so that the accuracy and efficiency in segmentation are improved [12]. The RGBAL method [12] involves converting images from RGB color format into a polar grid representation. It then employs fusion strategies at both early and mid-level stages to design them. PointPainting [13] utilizes image segmentation logits, transferring them to LiDAR space to enhance the performance of the LiDAR network, utilizing structures such as a spherical view or a bird's eye view (BEV). The PMF approach [14] utilizes the joint integration of two techniques within the camera coordinates. Nevertheless, at both training and inference stages, these techniques rely on multi-sensor inputs. In addition, Multimodal data are usually computationally compressed. At the global level, [15] has introduced an extensive framework from fine to coarse, which includes two networks. This approach involves the network initially prioritizing the completion of the coarse global structures, while the second network uses it as a guide to enhance the finer local details. In recent research, two-step approaches that adhere to the concept of structure-texture decomposition [16] have gained popularity. Some researches [17, 18] modify the framework in such a way that the resulting components are generated concurrently instead of sequentially. Furthermore, various studies have put forward two-stage approaches employing the completion of different types of structures as an intermediate step. For instance, in [19],

the focus is on salient edges, while [20] tackles semantic segmentation maps, [21] deals with object foreground contours, [22] addresses gradient maps, and [23] focuses on smooth edge-preserved images. Point-based methods face a challenge known as overhead, stemming from their costly random memory access, particularly noticeable in large-scale outdoor scenes. Voxel-based methods offer a solution by employing thin convolution techniques. As a follow-up, SPVNAS [24] introduces the concept of sparse point-voxel complexity. To mitigate the challenge posed by imbalanced point distribution, Cylinder3D [25] introduces a methodology based on cylindrical partitioning and integrates a 3D convolution network for structural enhancement. RPVNet exploits three diverse point representations and amalgamates them into a cohesive network. To improve the network training performance, 2DPASS [5] and PVKD [26] use knowledge distillation strategies to enhance the network. Knowledge distillation (KD) originates from the pioneering work of G. Hinton et al [27]. Its primary aim is to transfer hidden insights from a teacher model with excessive parameters to a more streamlined student model. A plethora of approaches have been proposed, encompassing diverse forms of knowledge transfer, including intermediate features [28], visual attention maps [29], cross-sample similarity scores [30], region-level affinity scores [31], among others. Inspired by [5], we reach very strong results in semantic segmentation using knowledge distillation of 2D and 3D information.

2.2 Adversarial inpainting

Inpainting is an image processing process used to be utilized for reconstructing images that have suffered from loss or damage of information caused by factors like occlusion, blurring, or transmission interference. This process assimilates information regarding the absent segments of the image, grasps the holistic structure of the image, and integrates other pertinent details to ensure precise reconstruction. Inpainting serves various functions and finds applications in numerous image processing scenarios, such as removing unwanted objects, repairing damage, and eliminating occluded areas on objects. The multi-image fusion technique for occlusion-free texture was introduced by Böhm et al. [32]. This method utilizes a process akin to background subtraction. Within a set of captured images, pixels sharing similar RGB values are clustered together, and outliers are subsequently filtered out. The background pixel is determined by selecting the pixel with the highest number of "votes" from the remaining clustered pixels. The early methods for performing inpainting operations inside the image, it was mainly based on data. These approaches included the use of patch-based methods and nearest neighbors. In the era of deep learning, an early approach in indoor inpainting involved employing a convolutional neural network architecture featuring an encoder-decoder structure, coupled with adversarial training, to complete missing elements [33]. This approach is known as one of the common methods for deep inpainting. In the nascent phases of deep learning research, Pathak et al [34] introduced an encoder-decoder architecture, trained using a combination of pixel-based adversarial loss and reconstruction loss. For the enhancement of image completion stability, Iizuka et al. [35] introduced global and local context discriminators into the training of a fully convolutional completion network. Their primary focus lies in discriminator design, complemented by the utilization of a simple encoder-decoder network as the generator. Additionally, [36] introduced an improved patch-based discrimination approach, which later gained traction among researchers. Furthermore, [37] proposed an innovative approach involving patch-based discrimination. Subsequently, they implemented a Partial convolution operation, followed by an automated mask update step, aimed at enhancing the filling of irregular holes. [38] introduced an encoder network for image completion, leveraging a pyramid-context and attention transfer. Meanwhile, to simultaneously recover both structure and texture, Liu et al. [39] integrated texture features and structure features through feature equalization. Additionally, [40] proposed mask-aware convolution along with point normalization, catering to the dynamic concept of image inpainting. Wang et al. [41] proposed regional composite normalization and migratable convolution modules to improve the utilization of valid pixels throughout the inpainting procedure. In a similar vein, Zhu et al. [42] utilized a semantic segmentation map to guide the inpainting process of mixed scenes, requiring supplementary semantic segmentation annotations during the training phase. These methods sometimes face significant challenges due to the lack of sufficient constraints, significant artifacts such as smooth textures, and false semantics. Yu et al. [15] refined a generative approach for inpainting by integrating both coarse and refined grids. In the refinement network, they implemented contextual attention mechanisms to capture extensive correlations spanning longer distances within the input data. Nazeri et al. [43] introduced a two-step edge-guided approach for image inpainting. In their approach, they first reconstructed the edge map of the occluded area and then combined it with the incomplete image to form the input for the subsequent inpainting stage. Architectures based on U-Net [44] are also popular choices in image completion. Fruh et al. [45] introduced an automatic method for creating textured 3D models of urban environments This method uses a vehicle equipped with cameras and laser scanners on city streets and creates 3D

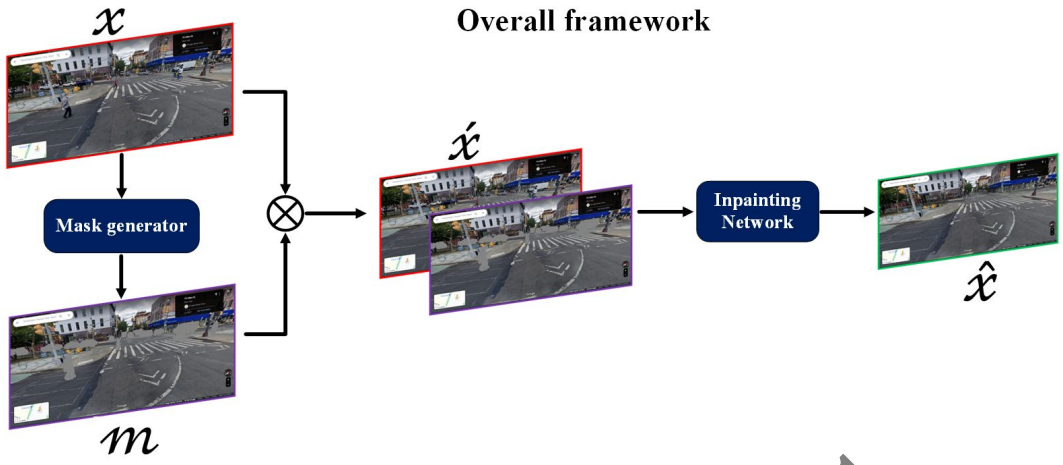

Fig. 1 Overall framework of our proposed method. x represents the input image, m denotes the mask produced by the mask generator block, ẋ signifies the input image with the mask and x̂ indicates the output inpainted image.

point clouds. Then, useful information is extracted by analyzing the images and removing the pixels of the foreground objects. Finally, textured 3D models are created by filling holes using methods such as cut-and-paste and interpolation. One of the main challenges in this field is the proper understanding of the local and global context. To improve this issue, [35] proposed an approach that uses incorporated dilated convolutions [46] to expand the knowledge domain of the network in image completion. Additionally, work on maintaining global and local consistency led to the introduction of two discriminators. In a study by [47], they proposed to combine branches in the complement network, each with different receiving contexts. The image inpainting model leverages stacked generative networks to ensure seamless texture and color coherence between the generated regions and their surrounding context. Furthermore, the integration of the contextual attention model into the networks allows for the borrowing of detailed information from distant spatial locations. To address the discrepancy between open-source datasets and facade inpainting content, this approach emphasizes training on custom datasets gathered from street facade images [15]. Another method, a new mechanism based on FFC introduced by [6]. Additionally, this approach is aligned with the utilization of transformers in computer vision [48], while also considering the Fourier transform as a lightweight substitute for self-attention [49].

3 Our method

Our proposed approach introduces an end-to-end method designed to detect and eliminate undesired objects from images automatically. The removal process is done by using the most accurate labels and creating masks on unwanted objects. To begin with, we used the SemanticKITTI dataset and trained the network with 18 labels for semantic segmentation. This type of semantic segmentation is pivotal for understanding vast outdoor environments and holds widespread applications in fields such as robotics and autonomous driving. After removing the unwanted images, large holes are created in the image, to fill them we used the internal inpainting technique of the image. We did this using a simple one-step network called LAMA (Large Mask Inpainting). The implementation steps of our method are illustrated in Fig. 1. Initially, the input image is processed by the mask generator block. Subsequently, the resulting mask is passed to the inpainting block along with the input image, facilitating the filling of the void created by the mask.

3.1 Semantic segmentation

In recent years, the research community has been deeply engaged in enhancing the understanding of natural scenes, leveraging camera images [50, 51] or LiDAR point clouds [24, 25, 52] as input sources. However, single-sensor methods have often faced problems in complex environments, these problems are caused by their inherent limitations. The input sensors clearly specify that the cameras provide accurate texture and fine-texture information, but do not perform reliably in detecting depth features, which are usually vaguely shaped and in low-light conditions. In contrast, extensive depth information regardless of light variance and LiDAR provide accurate, but only record thin information without texture information. Given the complementary nature of cameras and LiDAR, utilizing both types of sensors are advantageous for understanding the surrounding environment. Recently, numerous commercial vehicles have been outfitted with both LiDAR and cameras systems, enabling them to capture street effectively. This trend has spurred research endeavors aimed at improving semantic segmentation by amalgamating insights from

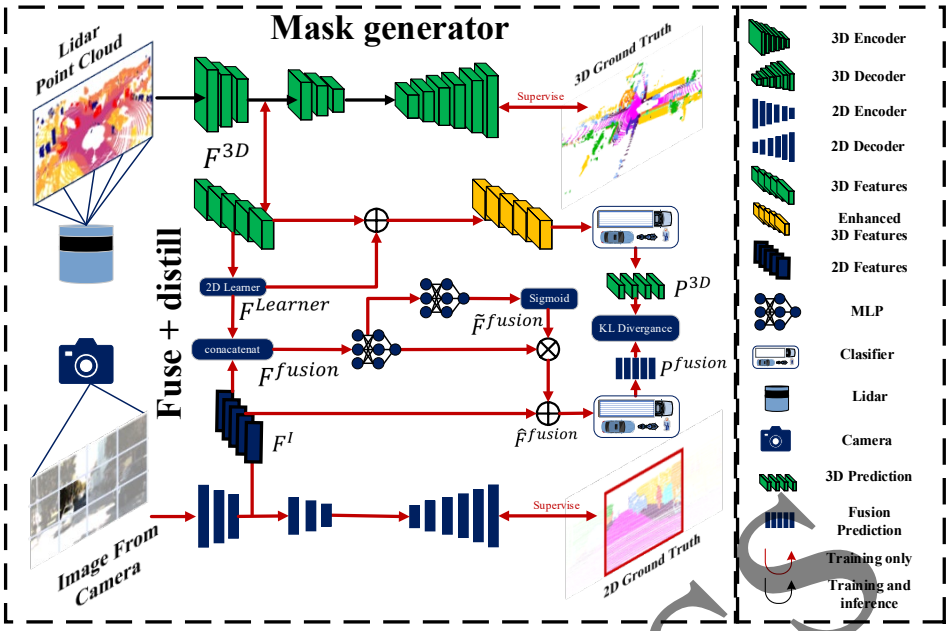

Fig. 2 Mask generator block using two-dimensional priors. Leveraging the 2D information from the camera image, a small segment is initially extracted from the original image. Subsequently, this extracted segment, along with the LiDAR point cloud, undergoes independent processing through 3D and 2D encoders simultaneously to generate multi-scale features in parallel.

two complementary sensors [12-14]. These approaches first create a 3D map between points using sensor calibration and then generate. 2D pixels using point clouds on image planes. Through point-to-pixel mapping, models merge pertinent features in images with point features, computed to derive final semantic metrics. However, fusion-based methods exhibit the following limitations:

a. Point-to-pixel mapping is impractical for points outside the image due to the differences in field of view (FOV) between LiDAR and cameras. Fusion-based methods are severely constrained by the fact that the field of view of most LiDAR systems and cameras only overlap to a small extent.

b. Fusion-based methods necessitate greater computing resources as they concurrently process both point clouds and images during runtime, thereby substantially enhancing the performance of real-time applications. The employed network primarily focuses on improving the semantic segmentation of the LiDAR point cloud by aiming to assign a semantic label to each point. Figure 2 provides a visual representation of the workflow steps of the Mask Generator block, which utilizes 2D Priors. The operation of the network is that first, camera images that are large in size (for example, 1242 x 512) are impossible to send to the multi-modal pipeline due to their large size. Therefore, a small patch with the size 480 × 320 is randomly sampled from the original camera image, and training processing is performed with this small patch, in order to increase the execution speed. Following sampling, the LiDAR point cloud and cropped image patch undergo independent processing through separate 2D and 3D encoders. This entails extracting 2D and 3D features concurrently from two distinct backbones. Then, using multi-scale fusion-to-single KD, the 3D network is enhanced using multiple features. This integration includes texture information, color-sensitive 2D features, and 3D core knowledge preservation. Ultimately, all 2D and 3D features at each scale contribute to generating semantic segmentation predictions, which are supervised by 3D labels. During the inference stage, the corresponding two-dimensional branch can be omitted, thus effectively circumventing the additional computational overhead in real-world applications. This structure provides an improvement in actual performance speed compared to methods based on information fusion.

4 Architectures

4.1 Modal-specific architectures

As explained in Fig. 2, in this block, two different networks are used to independently encode the multiscale features of the 3D point cloud and the 2D image. These two networks work as follows:

2D Encoder: This network uses ResNet34 [8] as a two-dimensional network to encode two-dimensional image features and operates using two-dimensional convolution, and its task is to transform the image. It is two-dimensional with different features and different scales.

3D Encoder: For the 3D network, the concept of sparse convolution [53] is used to build the 3D network. One of the features of this type of operation is its inherent sparsity, which means that the operation is only applied to

voxels that have non-empty values. In other words, in this network, calculation operations are performed only on the voxels that have data.

ResNet bottleneck structure: at each scale (2D and 3D), a hierarchical encoder with a design similar to that of the decoder in [24] is used. Also, ReLU is replaced by Leaky ReLU [54]. These two 2D and 3D networks extract different features and scales from 3D point clouds and 2D images.

These features are extracted from different scales, known L as feature maps, and used to enhance the 3D network and use in semantic segmentation predictions. Then the two-dimensional and three-dimensional features of each scale are obtained and displayed as $\{F_l^{2D}\}_{l=1}^{L}$ and $\{F_l^{3D}\}_{l=1}^{L}$. These features are then used to enhance the 3D network and used in semantic segmentation predictions. This step encompasses multi-scale fusion-to-single knowledge distillation, leveraging multi-modal features to enhance the performance of the 3D network. This integration involves incorporating texture information and color-aware 2D features, while also preserving the original 3D knowledge. In the final analysis, these features are harnessed across multiple scales to generate semantic segmentation predictions. These predictions are then overseen by unadulterated 3D labels. This approach allows the network to have different features and different scales to interpret and distinguish different components of scenes and objects. Within the two-dimensional network, the FCN decoder [55] has been implemented to extract features from each encoder layer. This indicates that the feature map D_l^{2D} from the lth decoder layer can be obtained by sampling the feature map from the encoder layer at position $(L - l + 1) - th$. This sampling operation is performed sequentially from lower layers to higher layers, and the sampled feature maps are combined through the merge operation. Finally, for the semantic segmentation process in the 2D network, the combined feature map is obtained through a linear classification. This linear classification contributes to the final image of semantic segmentation prediction. In the 3D network, the U-Net decoder isn't utilized; In contrast, the approach involves sampling features from different scales to match the original size, followed by binning them before feeding into the classifier. During knowledge distillation, the features of point clouds and images are initially merged, ensuring that the information conveyed by image features is amalgamated with the existing information from point clouds. Following this, an alignment process is conducted in a unidirectional manner between the fused features and those extracted from the point cloud. This means determining the weight that is assigned to image features and point cloud so that each information source determines its participation and contribution in producing the final output. In this way, finally, by combining image features and point cloud and performing unidirectional alignment, a compact and high-quality knowledge model is obtained for use in subsequent processes, such as image completion or pattern recognition. In this method, the fusion accurately preserves the complete information of the multivariate data. Furthermore, through unidirectional alignment, fusion yields enhanced point cloud features while safeguarding modality-specific information integrity. Regarding modality fusion, directly fusing the raw 3D features $\hat{F}_l^{3D}$ to their 2D counterparts $\hat{F}_l^{2D}$, for each scale is inefficient, given the disparity in 2D and 3D feature representations stemming from different backbone architectures. So, First, we convert $\hat{F}_l^{3D}$to $\hat{F}_l^{learner}$ through a 2D MLP learner, aiming to mitigate the feature gap. Subsequently, $\hat{F}_l^{learner}$ not only proceeds to the next concatenation with the 2D features $\hat{F}_l^{2D}$ to obtain the combined features $\hat{F}_l^{2D3D}$ through another MLP but also through a skip connection to the original dimension features to enhance the three-dimensional features of $\hat{F}_l^{3De}$. In addition, similar to the attention mechanism, the final augmented composite features $\hat{F}_l^{2D3De}$ with:

$$\hat{F}_l^{2D3De} = \hat{F}_l^{2D} + \sigma\left(\mathrm{MLP}\left(\hat{F}_l^{2D3D}\right)\right) \odot \hat{F}_l^{2D3D}, \qquad (1)$$

it will be obtained; where σ represents the sigmoid activation function.

Using a knowledge distillation scheme in this framework has several advantages. Firstly, it combines the 2D learner and fused distillation into single, rich texture information, which enhances the learning of 3D features without losing any modal information. This approach furnishes detailed information in three dimensions. Moreover, during the training phase, the fusion branch operates exclusively, indicating that the advanced model can be deployed with minimal additional computational overhead during inference. This structure allows the network to better learn hierarchical information while making predictions more efficiently. Ultimately, this structure may provide the best use of the information contained in the features and contribute to the accuracy of the predictions.

4.2 Point-to-Pixel mapping

Transferring information directly from one mode to another poses a challenge due to the representation of 2D and 3D features as pixels and points, respectively. The primary objective of this section is to generate paired features using two distinct methods. The process of generating these coupled features in both modes is detailed in Fig. 3. For instance, the generation process of 2D features is depicted in row b of Fig. 3. The process entails extracting a small patch I from the original image,

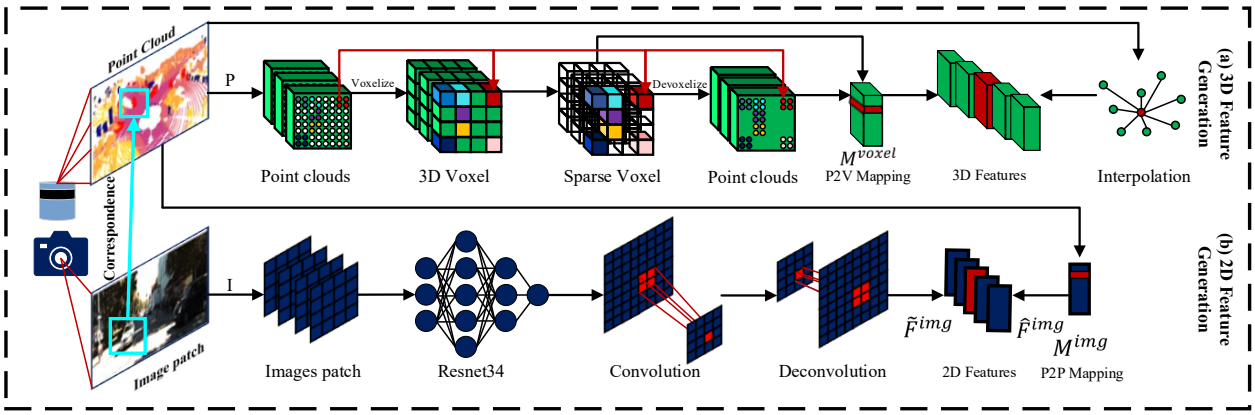

Fig. 3 Generation of 2D and 3D features. Part (a) depicts the generation of 3D features, where point-to-voxel (P2V) mapping is easily obtained and voxel features are interpolated onto the point cloud. Part (b) showcases the 2D feature generation, where the point cloud is initially projected onto the image segment, creating a point-to-pixel (P2P) mapping. Subsequently, the 2D feature map is transferred to 2D point features according to the P2P mapping.

which belongs to $\mathbb{R}^{H\times W\times 3}$, and then processing it through a 2D grid. As a result, multiscale features with varying resolutions are extracted from the hidden layers. As an illustration, let's consider the feature map F_l^{2D} from layer l, where $F_l^{2D} \in \mathbb{R}^{H_l\times W_l\times D_l}$. Initially, a deconvolution operation is conducted to enhance the resolution, resulting in the original $\tilde{F}_l^{2D}$. In line with recent advancements in multi-sensor methodologies [14], point clouds and images are computed utilizing perspective projection and point-to-pixel mapping techniques. To elaborate, within a LiDAR point cloud $P = \{p_i\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}$, each 3D point $p_i = (x_i, y_i, z_i) \in R^3$ is projected onto a point $\hat{p}_l = (u_i, v_i) \in R^2$ on the image plane according to the following scheme:

$$[u_i, v_i, 1]^T = \frac{1}{z_i} \times K \times T \times [x_i, y_i, z_i, 1]^T \quad (2)$$

where the internal matrices $K \in R^{(3\times 4)}$ and the external matrices $T \in R^{(4\times 4)}$ represent cameras. These matrices K and T are directly provided in KITTI [56]. Mapping 3D features is relatively straightforward, as depicted in Fig. 3 row A. Specifically, for the point cloud $P = \{(x_i, y_i, z_i)\}_{i=1}^{N}$, a point-to-voxel mapping in the lth layer is executed through

$$M_l^{voxel} = \left\{\left(\left\lfloor\frac{x_i}{r_l}\right\rfloor, \left\lfloor\frac{y_i}{r_l}\right\rfloor, \left\lfloor\frac{zi}{r_l}\right\rfloor, \right)\right\}_{i=1}^{N} \in \mathbb{R}^{N\times 3}, \quad (3)$$

we get where r_l is the resolution of voxelization in layer lth. Then, according to the 3-dimensional feature $F_l^{3D} \in \mathbb{R}^{\acute{N}_l\times D_l}$ of a thin twist layer, we obtain a point. The 3-dimensional feature $\tilde{F}_l^{3D} \in \mathbb{R}^{N\times D_l}$ through the closest interpolation in the original F_l^{3D}map feature corresponds to M_l^{voxel}. Finally, points are filtered by discarding those that fall outside the field of view of the image:

$$\hat{F}_l^{3D} = \{f_i | f_i \in \tilde{F}_l^{3D}, M_{i,1}^{img} \le H, M_{i,2}^{img} \le W\}_{i=1}^{N} \in \mathbb{R}^{N^{img}\times D_l}, \quad (4)$$

According to the provided information, the 2D images are on the image screen corresponding to those images. In the next step, this predicted 2D ground truth is used as a monitoring criterion for 2D branch-related issues [5].

4.3 Inpainting image

We use the LAMA [6] model for inpainting the created holes. Our goal is to get the masks created by semantic segmentation as input to this data network and the output of Inpainting. During the early layers of the network, the decision regarding global temporal integration is pivotal, as it facilitates complex tasks like filling large masks. In such scenarios, an effective architecture should incorporate units with the widest possible receptive field in the primary layers. In the early layers, conventional models like ResNet may encounter challenges due to the slow expansion of the receptive field. This limitation arises because, particularly in the network's early layers, the receptive field may not grow as rapidly as desired, as they typically employ small convolution kernels (e.g., 3x3). In other words, since the convolution kernels in the initial layers have little spatial information, many of these layers will lack the global context, and this leads to a waste of computation and parameters. Additionally, for wide masks, it is possible for the entire receptive field of the generator to be located at a specific position within the mask and only observe the missing pixels. To address this issue, particularly prominent in high-resolution images, an architecture with units possessing a larger receptive field and enhanced spatial perception capability is required. In recent times, fast Fourier convolution [48] has emerged as a solution enabling the incorporation of global context in early layers. Utilizing a channel fast Fourier transform (FFT), FFC extends its reach to encompass a receptive field that spans the entirety of the image dimensions. The two branches divided in parallel

in this operator are as follows:

- The local branch employs regular convolutions for processing.
- The global branch employs the real FFT to compute the global field.
- The key characteristic of the real Fast Fourier Transform is its applicability solely to real-valued signals.

Moreover, it utilizes the inverse real FFT to ensure real-valued output. The FFC performance involves several steps:

a) Applying real FFT2d on an input tensor

$$\text{Real FFT2d: } \mathbb{R}^{H\times W\times C} \rightarrow \mathbb{C}^{H\times \frac{W}{2}\times C}, \quad (5)$$

and connecting the and imaginary real parts together

$$\text{ComplexToReal: } \mathbb{C}^{H\times \frac{W}{2}\times C} \rightarrow \mathbb{R}^{H\times \frac{W}{2}\times 2C}, \quad (6)$$

b) applying a convolution block within the frequency domain

$$\text{ReLU} \circ \text{BN} \circ \text{Conv1}\times 1 : \mathbb{R}^{H\times \frac{W}{2}\times 2C} \rightarrow \mathbb{R}^{H\times \frac{W}{2}\times 2C}, \quad (7)$$

c) Restoring the spatial structure involves performing an inverse transformation.

$$\text{RealToComplex: } \mathbb{R}^{H\times \frac{W}{2}\times 2C} \rightarrow \mathbb{C}^{H\times \frac{W}{2}\times C}, \quad (8)$$

$$\text{Inverse Real FFT2d: } \mathbb{C}^{H\times \frac{W}{2}\times C} \rightarrow \mathbb{R}^{H\times W\times C}, \quad (9)$$

To sum up, integration of the outputs from both the local (a) and global (b) branches occurs. Fig. 4 illustrates the architecture of the inpainting network. LaMa integrates the subsequent methodologies to boost performance and efficiency in large mask inpainting tasks:

FFC: Recently introduced, this method facilitates the integration of global context in primary layers. FFC harnesses the channel FFT, resulting in a substantial receptive field that encompasses the entirety of the image. Multi-component loss: To enhance the quality of inpainting for large masks, this aspect of the approach integrates both perceptual and adversarial loss, resulting in a broad receptive field.

Training method - mask enlargement time: Serving as a training strategy for mask enlargement, this component enables the model to accurately produce large masks. The efficacy of FFC is particularly crucial in this aspect.

These operators are fully differentiable and easily replace conventional convolutions in deep networks. Due to the provision of a wide receptive field, algorithms are able to access global information through elementary layers. This issue is very important for high-resolution images, because these types of images require more accuracy and variety of information for accurate and high-quality reconstruction. From the beginning, the FFC network is able to provide a wide receiving field and take advantage of the global context to achieve the best results in tasks such as inpainting high-resolution images.

4.4 Inpainting loss functions

The problem of inpainting (filling in missing items in an image) is ambiguous in nature, as many valid alternatives can be provided for regions where information is missing, especially when the "holes" are wider. Adversarial Loss is employed to address this issue and ensure the natural creation of local details. Here, a discriminator $D_\xi(\cdot)$ is defined that operates at the local segment level [57] and differentiates between "real" and "fake" segments. "fake" segments are labeled "fake" only for pieces of the image that intersect with the masked region. This means that the discriminator only considers the areas that need to be filled. By employing the perceptual loss of HRF (High-Resolution Features), this approach facilitates rapid learning to replicate known sections of the input image. Subsequently, these labeled "real" parts are utilized as local details in production images. Overall, leveraging non-saturating adversarial loss enables the model to excel in the inpainting task by producing valid and high-quality images to fill in the missing regions.

$$L_D = -\mathbb{E}_x[\log D_\xi(x)] - \mathbb{E}_{x,m}[\log D_\xi(\hat{x}) \odot m] - \mathbb{E}_{x,m}[\log(1 - D_\xi(\hat{x})) \odot (1 - m)] \quad (10)$$

$$L_G = -\mathbb{E}_{x,m}[\log D_\xi(\hat{x})] \quad (11)$$

$$L_{Adv} = sg_\theta(L_D) + sg_\xi(L_G) \rightarrow \min_{\theta,\xi} \quad (12)$$

where in

x: This variable is an instance of the dataset that is given as input to the model.

m: The m mask indicates the areas of the image that need to be filled.

$\hat{x} = f_\theta(\acute{x})$: The result of inpainting is the image $\acute{x}$. f_θ is a function that transforms the image $\acute{x}$ into an image with new colors according to the input mask m.

$\acute{x} = \text{stack}(x \odot m, m)$: Considered as input to f_θ model. This input is created using mask m to combine the original image x with the mask.

sg_{var}: This indicates the gradients (Gradient) relative to the var variable. Gradients signify the alterations of the loss function concerning the desired variable.

L_{Adv}: This value represents the Adversarial Loss function. This loss function quantifies the disparity between the reconstructed images($\hat{x}$) and the actual images (x).

This loss function is employed to quantify the disparity between the generated images ($\hat{x}$) and the real images (x).

The purpose of this loss function is to encourage the model to produce images closer to real images.

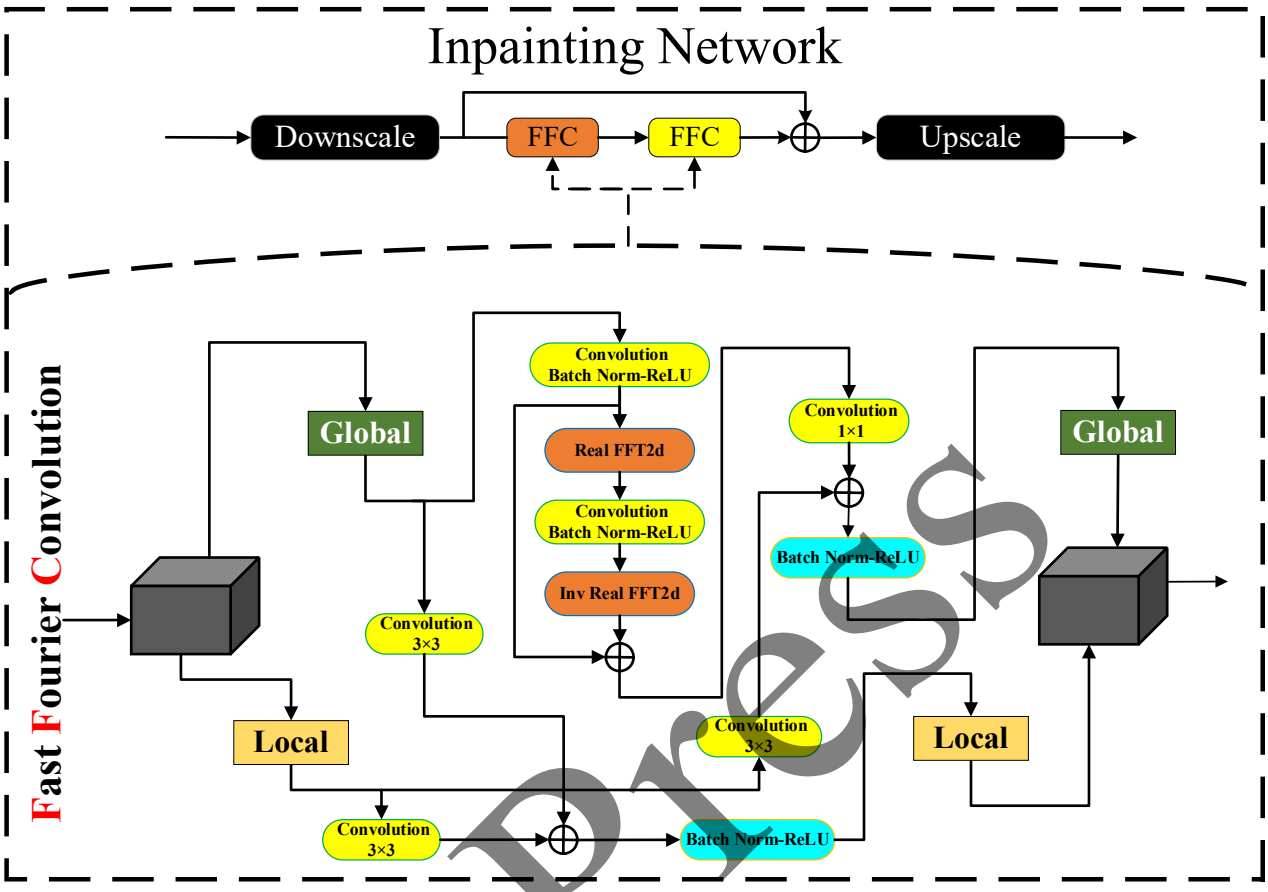

Fig. 4 Large mask inpainting. This network is designed to combine multi-component, adversarial, high receptive field perceptual loss and mask generation process based on an internal FFC network during training.

4.5 Final loss function

The ultimate loss function serves as a metric to evaluate the quality and efficacy of the model in generating inpainted images. Also $R_1 = E_x||\nabla D_\xi(x)||^2$ gradient penalty [58], and perceptual loss based on differentiation or so-called feature matching loss have been used. This final loss function is defined as follows:

$$L_{final} = \alpha L_{HRFPL} + \beta L_{DiscPL} + \gamma R_1 + \kappa L_{Adv} \quad (13)$$

L_HRFPL corresponds to the monitored signal and captures the global structure of the image, while L_DiscPL, a discriminative loss function, is utilized to differentiate between generated and real images. R_1 is Gradient Penalty, which is used to set the number of gradient changes in the optimization process. $\kappa, \alpha, \beta, \gamma$ give weight to different values of final loss function. By using the final loss function and proper setting of parameters, the model improves the quality of inpainting images and has better performance in performing tasks.

5 Results

5.1 Semantic Segmentation settings

This section delves into exploring the properties of the selected segmentation model, examining its suitability for evaluation within 360-degree space using the SemanticKITTI dataset. SemanticKITTI delivers meticulous semantic annotations, ensuring comprehensive coverage for scans in sequences 10-00 of the KITTI dataset [56]. Formally, sequence 08 serves as the validation set, while the remaining sequences are utilized for training the model. Furthermore, the test set comprises sequences 11–21 from the KITTI dataset. SemanticKITTI comprises sequences of 3D scans of street environments. These sequences are collected with great care and contain spatial shape and semantic information of objects in the environment. One of the unique features of SemanticKITTI is the detailed semantic annotation for each scan in the sequences. This annotation means categorizing the objects in each scan and assigning semantic labels to three-dimensional points

in the environment. To enhance the accuracy assessment of the segmentation model, we utilize the mean Intersection over Union metric. This metric calculates the average IoU overlap across all classes. The mIoU measure means the average correspondence of objects with units of objects in images. To compute this metric, we first determine the intersection over union and the union size between each class or object in the image and the unit of objects. Then the IoU is calculated for each object and the average of these IoU values for all objects is reported as mIoU. Mathematically, the mIoU measure is computed as follows:

$$\mathrm{mIoU} = \frac{1}{\mathrm{N}} \sum_{\mathrm{i}=1}^{\mathrm{N}} \frac{\mathrm{TP_i}}{\mathrm{TP_i} + \mathrm{FP_i} + \mathrm{FN_i}} \tag{14}$$

Where N represents the number of classes or objects in the image, $\mathrm{TP_i}$ denotes the number of actual matching points of objects with the unit of objects for class i, $\mathrm{FP_i}$ represents the number of points of incorrect matching of objects with the unit of objects for class i, and $\mathrm{FN_i}$ denotes the number of points of non-matching of objects with the unit of objects for class i.

This criterion serves as an indicator of the model's performance quality in matching with various objects. In addition, we report two other measures. First, we calculate the overall accuracy for each class. To implement the model, we use a ResNet34 encoder with 2D complexity. In this method, features are generated after each down sampling layer in order to extract 2D features. Also, to enhance the speed of the network in the 3D model, we use a modified SPVCNN [24] with a voxel size of 0.1 and less parameters. The hidden dimension of this network is determined to be 64 for the SemanticKITTI dataset. The utilization of L layers is also critical for amalgamating the collective knowledge. In the context of the SemanticKITTI dataset, L is defined as 4. Throughout each stage of knowledge transfer, both 3D and 2D features undergo adjustment to 64 dimensions via recurrent processing or multilayer neural networks. Similarly, the hidden size of multi-layer networks and 2D learner in hybrid knowledge fusion is also set to 64.

5.2 The results of the semantic segmentation training

Within the context of semantic segmentation section, both the cross-entropy error function and the Lovasz error function [59] are employed. Also, for the knowledge transfer process, we have set the ratio of detection error value to KL deviation as 1 to 0.05. In addition, in the test phase, we use the technique of increasing the data of the test time. Training of the network was conducted using the NVIDIA Tesla T4 GPU equipped with 16 GB GDDR6 memory and 2,560 CUDA cores. A batch size of 6 was employed, and approximately 208 hours were invested in model training.

Table 1 presents the outcomes of our training utilizing the 2D pass network, showcasing class accuracy and evaluation results. While results for all classes are included to demonstrate the model's robustness, the primary focus of this study is on privacy-sensitive objects, particularly persons and vehicles. For the person category, which combines person (mIoU: 76.6%), and bicyclists (mIoU: 87.9%), the model demonstrates strong segmentation performance, effectively detecting individuals across diverse scenarios. For vehicles, the results span multiple subcategories, including car (mIoU: 96.5%), truck (mIoU: 75.3%), motorcycle (mIoU: 63.3%), and bicycle (mIoU: 45.8%). These results highlight the model's robustness in segmenting sensitive

Table 1. Evaluation results of semantic segmentation in different classes

Method	mIoU	road	sidewalk	parking	other-gro.	building	car	truck	bicycle	motorcycle	vegetation	trunk	terrain	person	bicyclist	motorcyclist	fence	pole	Traffic sign
RangeNet53++ [17]	52.2	91.8	75.2	65.0	27.8	87.4	91.4	25.7	25.7	34.4	80.5	55.1	64.6	38.3	38.8	4.8	58.6	47.9	55.9
Meta-RangeSeg [60]	61.0	90.7	74.6	64.3	29.2	91.1	93.9	43.9	53.1	43.8	82.6	65.5	65.5	63.7	53.1	18.7	64.7	56.3	**64.2**
CNN-LSTM [61]	56.9	90.7	75.7	23.3	17.6	90.0	92.6	48.6	**74.6**	49.6	87.1	60.8	75.4	53.8	74.6	9.2	51.3	63.9	41.5
3D-MiniNet [62]	55.8	91.6	74.5	64.2	25.4	89.4	90.5	28.5	42.3	42.1	82.8	60.8	66.7	47.8	44.1	14.5	60.8	48.0	56.6
GAF-NET [63]	58.8	91.0	74.6	61.9	24.2	89.5	94.7	34.2	33.5	33.6	84.2	65.3	68.4	48.8	50.5	-	61.3	52.2	53.3
TransRVNet [64]	64.8	91.9	76.5	**68.5**	29.9	91.0	92.7	43.4	51.2	50.3	84.4	67.6	70.2	62.1	55.5	-	**67.6**	59.2	62.5
NAPL [65]	61.6	89.6	73.7	67.1	**31.2**	**91.9**	**96.6**	47.3	32.3	43.6	84.8	69.8	68.8	51.1	53.9	**36.5**	67.4	59.1	59.2
SqueezeSegV3 [53]	55.9	91.7	74.8	63.4	26.4	89.0	92.5	29.6	38.7	36.5	82.0	58.7	65.4	45.6	46.2	20.1	59.4	49.6	58.9
CGGC-Net [66]	60.8	86.9	73.7	59.0	15.7	91.3	94.5	50.8	35.2	40.8	83.9	64.9	68.2	58.8	57.6	-	62.8	52.5	53.3
MASNet [67]	64.6	**96.5**	**80.0**	46.6	0.6	88.5	95.3	**83.3**	46.3	62.9	87.6	70.3	74.1	73.7	78.1	-	63.5	63.4	42.4
Ours	**64.9**	92.7	79.6	42.7	2.6	89.9	96.5	75.3	45.8	**63.3**	**89.4**	**72.1**	**77.2**	**76.6**	**87.9**	0.0	57.6	**64.2**	54.1

objects, even under challenging conditions such as occlusions and varying object scales. Including other classes in the evaluation ensures a holistic assessment of the model's overall performance, demonstrating its generalization ability without compromising accuracy for key classes

Table 2 compares the inference times of various segmentation algorithms. All evaluations were conducted on an NVIDIA Tesla T4 GPU. Our method outperforms several state-of-the-art approaches, demonstrating superior efficiency and minimal computational overhead, making it highly suitable for fast processing in large-scale applications.

Table 2. Comparison of Inference Time across Segmentation Algorithms.

Method	Infrance Time
RangeNet53++ [17]	83.3s
SqueezeSegV3 [53]	238ms
PointNet++ [68]	5900ms
TransRVNet [64]	38.4ms
RandLA-Net [69]	880ms
PolarNet [70]	62ms
Ours	**62ms**

5.3 Inpainting results

To achieve better results in inpainting images with large masks, we suggest using [6], which uses a variety of strong baselines at lower resolutions. This difference in performance and the ability to reveal more during the painting process is evident. For training both the image completion and discriminator models, we employed the Adam optimizer, utilizing fixed learning rates of 0.001 and 0.0001, respectively, for the networks. Furthermore, all models undergo training for 1 million iterations with a size of 30. Across all experimental stages, hyperparameters are meticulously chosen through the coordinate beam search strategy. We used the pre-trained LaMa model for inpainting. Utilizing the coordinate beam search strategy, the following weight settings were attained: $\alpha = 30$, $\beta = 100$, $\gamma = 0.001$, $\kappa = 10$. It's noteworthy that the hyperparameter search is consistently executed on a distinct subset of the validation data across all cases. For model tests, the Places2 [71] dataset have been used as input data. In the design of the models, the established method presented in the recent image2image literature has been followed. To evaluate the performance of the models, well-established metrics like the learned perceptual image patch similarity (LPIPS) and the initial Frechet inception distance (FID) have been employed. These metrics are juxtaposed with the L1 and L2 distances at the pixel level for comparison. These two criteria, when several natural finishes are acceptable, are recognized as more suitable criteria for evaluating the performance of masks in the inpainting process. We then compare our proposed approach with several strong baselines, as shown in Table 3. For this evaluation, the performance of various inpainting methods is assessed across three different mask sizes, with FID and LPIPS as the primary metrics. The results demonstrate that LAMA consistently outperforms most of the baselines, delivering the best performance across all mask sizes. LAMA achieves superior results in both FID and LPIPS, particularly in the 0.01%-20% and 20%-40% ranges, where it maintains a clear advantage over other methods.

Table 3. We present a quantitative evaluation of inpainting results based on FID and LPIPS metrics across three different mask sizes: 0.01%-20%, 20%-40%, and 40%-60%. Our experiments show that LAMA consistently achieves superior performance compared to a broad set of baseline methods, delivering more accurate inpainting results that better preserve perceptual quality and align with the true distribution of real-world images.

Method	0.01%-20%		20%-40%		40%-60%	
	FID	**LPIPS**	**FID**	**LPIPS**	**FID**	**LPIPS**
Deep Fill v2 [34]	23.6854	0.0446	27.3259	0.1362	36.5458	0.2891
CTSDG [72]	24.9852	0.0458	29.2158	0.1429	37.4251	02712
WaveFill [73]	30.4259	0.0519	39.8519	0.1365	56.7527	0.3395
LDM [74]	-	-	-	-	27.3619	0.2675
WNet [75]	20.4925	0.0387	24.7436	0.1136	32.6729	0.2416
MISF [76]	21.7526	0.0357	30.5499	0.1183	44.4778	0.2278
CMT [77]	22.1841	0.0364	32.0184	0.1184	35.1688	0.2378
MxT [78]	15.3980	**0.0334**	23.7109	0.1106	26.9155	0.2372
LAMA [6]	**14.7288**	0.0354	**22.9381**	**0.1079**	**25.9436**	**0.2124**

Table 4 compares the inference times of eight image inpainting algorithms, all evaluated on an NVIDIA Tesla T4 GPU. AOT-GAN [80] achieves the fastest performance with an inference time of 0.05 seconds, while LaMa [6], the method employed in this study, provides competitive efficiency at 0.5 seconds. DDRM [65] follows with a fast inference time of 1.5 seconds. In contrast, Score-SDE [79] takes 25 seconds, RePAINT [81] requires 150 seconds, and ICT [82] takes 110 seconds. DSI [83] has an inference time of 30 seconds. IAGAN [84] exhibits the longest inference time at 350

seconds. These differences highlight the trade-offs between computational efficiency and model complexity.

Table 4. Comparison of Inference Time across Inpainting Algorithms.

Method	Infrance Time
Score-SDE [79]	25S
AOT-GAN [80]	0.05S
DDRM [65]	1.5S
RePAINT [81]	150S
ICT [82]	110S
DSI [83]	30S
IAGAN [84]	350S
LaMa [6]	**0.5S**

6 Final results

We used Google Street View images to evaluate our method. GSV images have special challenges for segmentation and inpainting due to the wide range of imaging angles that provide 360-degree information. The features of these images include the following:

- 360-degree coverage: These images provide information from different angles and completely in 360 degrees, which challenges segmentation and inpainting every part of this image.
- Spatial and temporal changes: Spatial and temporal changes in a 360-degree image can reduce the stability and accuracy of segmentation and inpainting because objects and objects may be seen in different positions and times.
- Differences in lighting and shadows: Due to differences in lighting during the day and in different locations, it may become difficult to distinguish boundaries and separate objects in the image.
- Changes in scale and distance: Objects in GSV images may be seen at different distances and scales, which can complicate their detection, segmentation, and inpainting.

As depicted in Fig. 5, our method effectively overcomes these challenges and achieves superior results.

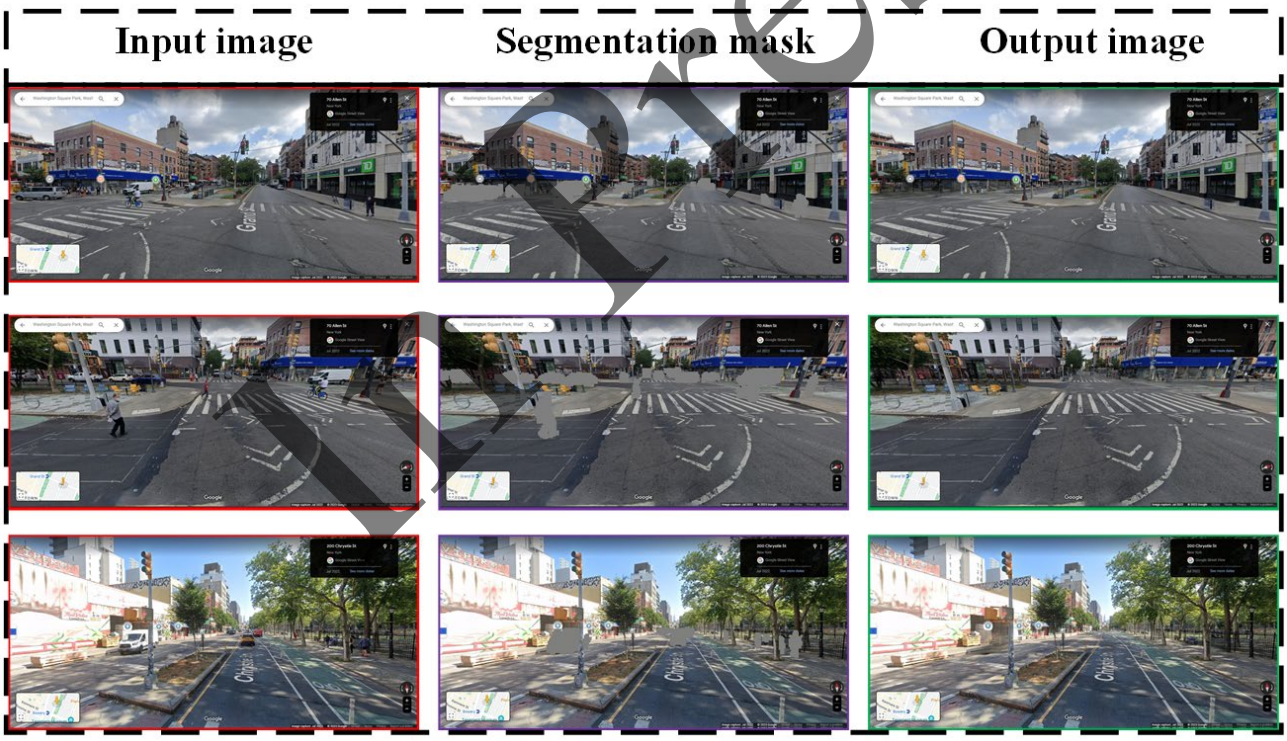

Fig. 5 The results obtained from our method. the left column portrays the input image, the middle column illustrates the segmented image and the right column showcases the inpainting output.

7 Conclusion

This paper introduces a novel approach that yields substantial benefits by removing people and vehicles from street view images and subsequently employing inpainting techniques to fill the resulting holes. One of the basic advantages of this method is to increase the privacy of people in the images. Due to the fact that street image monitoring technologies are becoming more popular day by day, people's privacy is at risk. This method allows us to remove people and vehicles from images while keeping

the images attractive. This research utilizes 2DPASS semantic segmentation, providing an extensive training framework to enhance LIDAR point cloud semantic segmentation performance through the integration of prior knowledge. By leveraging semantic modeling, 2DPASS extracts comprehensive semantic and structural insights from multimodal data, thereby enhancing the effectiveness of a 3D network. Furthermore, in inpainting tasks, a straightforward and single-step approach for addressing large masks has been explored. Fast Fourier convolutions enable our method to generalize effectively to higher resolutions while maintaining more efficient parameterization compared to baseline techniques. This method creates a significant improvement in the clarity and cleanliness of images after removing people and vehicles. By removing unwanted objects and applying inpainting techniques, images are displayed without interference. This helps users to better access the important information of the images. In addition, this method can be effective in reducing the interference and occupation of unwanted objects (from the 18 classes we taught) in the images and display the images more perfectly.

References

[1] F. Xu, A. Jin, X. Chen, and G. Li, "New Data, Integrated Methods and Multiple Applications: A Review of Urban Studies based on Street View Images," in *2021 IEEE International Geoscience and Remote Sensing Symposium IGARSS*, 2021: IEEE, pp. 6532-6535.

[2] L. Vincent, "Taking online maps down to street level," *Computer*, vol. 40, no. 12, pp. 118-120, 2007.

[3] A. Frome, G. Cheung, A. Abdulkader, M. Zennaro, B.Wu, A. Bissacco, H. Adam, H. Neven, and Luc Vincent, "Large-scale privacy protection in google street view," in *2009 IEEE 12th International Conference on Computer Vision*, 2009: IEEE, pp. 2373-2380.

[4] A. Flores and S. Belongie, "Removing pedestrians from google street view images," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops*, 2010: IEEE, pp. 53-58.

[5] X. Yan, J. Gao, C. Zheng, C. Zheng, R. Zhang, S. Cui, Z. Li, "2dpass: 2d priors assisted semantic segmentation on lidar point clouds," in *European Conference on Computer Vision*, 2022: Springer, pp. 677-695.

[6] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, and V. Lempitsky "Resolution-robust large mask inpainting with fourier convolutions," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision,* 2022, pp. 2149-2159.

[7] Y. Sun, H. Zhu, F. Zhuang, J. Gu, and Q. He, "Exploring the urban region-of-interest through the analysis of online map search queries," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 2269-2278.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770-778.

[9] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2017, pp. 1251-1258.

[10] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 779-788.

[11] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 801-818.

[12] K. Madawi, H. Rashed, A. Sallab, O. Nasr, H. Kamel, and S. Yogamani, "Rgb and lidar fusion based 3d semantic segmentation for autonomous driving," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019: IEEE, pp. 7-12.

[13] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "Pointpainting: Sequential fusion for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 2020, pp. 4604-4612.

[14] Z. Zhuang, R. Li, K. Jia, Q. Wang, Y. Li, and M. Tan, "Perception-aware multi-sensor fusion for 3d lidar semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 16280-16290.

[15] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5505-5514.

[16] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Transactions on Image Processing*, vol. 12, no. 8, pp. 882-889, 2003.

[17] M.-c. Sagong, Y.-g. Shin, S.-w. Kim, S. Park, and S.-j. Ko, "Pepsi: Fast image inpainting with parallel decoding network," in *Proceedings of the IEEE/CVF*

Conference on Computer Vision and Pattern Recognition, 2019, pp. 11360-11368.

[18] Y.-G. Shin, M.-C. Sagong, Y.-J. Yeo, S.-W. Kim, and S.-J. Ko, "Pepsi++: Fast and lightweight network for image inpainting," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 252-265, 2020.

[19] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "Edgeconnect: Generative image inpainting with adversarial edge learning," *arXiv preprint* arXiv:1901.00212, 2019.

[20] Y. Song, C. Yang, Y. Shen, P. Wang, Q. Huang, and C.-C. J. Kuo, "Spg-net: Segmentation prediction and guidance network for image inpainting," *arXiv preprint* arXiv:1805.03356, 2018.

[21] W. Xiong, J. Yu, Z. Lin, J. Yang, X. Lu, C. Barnes and J. Luo, "Foreground-aware image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5840-5848.

[22] J. Yang, Z. Qi, and Y. Shi, "Learning to incorporate structure knowledge for image inpainting," in *Proceedings of the AAAI Conference on Artificial Intelligence, 2020*, vol. 34, no. 07, pp. 12605-12612.

[23] Y. Ren, X. Yu, R. Zhang, T. H. Li, S. Liu, and G. Li, "Structureflow: Image inpainting via structure-aware appearance flow," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 181-190.

[24] H. Tang, Z. Liu, S. Zhao, Y. Lin, J. Lin, H. Wang and S. Han, "Searching efficient 3d architectures with sparse point-voxel convolution," in *European Conference on Computer Vision*, 2020: Springer, pp. 685-702.

[25] X. Zhu, H. Zhou, T. Wang, F. Hong, W. Li, Y. Ma, H. Li, R. Yang, and D. Lin, "Cylindrical and asymmetrical 3d convolution networks for lidar segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9939-9948.

[26] Y. Hou, X. Zhu, Y. Ma, C. C. Loy, and Y. Li, "Point-to-voxel knowledge distillation for lidar semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 8479-8488.

[27] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *arXiv preprint* arXiv:1503.02531, 2015.

[28] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning to steer by mimicking features from heterogeneous auxiliary networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, vol. 33, no. 01, pp. 8433-8440.

[29] Y. Hou, Z. Ma, C. Liu, and C. C. Loy, "Learning lightweight lane detection cnns by self attention distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1013-1021.

[30] F. Tung and G. Mori, "Similarity-preserving knowledge distillation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1365-1374.

[31] Y. Hou, Z. Ma, C. Liu, T.-W. Hui, and C. C. Loy, "Inter-region affinity distillation for road marking segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12486-12495.

[32] J. Böhm, "Multi-image fusion for occlusion-free façade texturing," *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 35, no. 5, pp. 867-872, 2004.

[33] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozai, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in Neural Information Processing Systems*, vol. 27, 2014.

[34] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536-2544.

[35] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1-14, 2017.

[36] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4471-4480.

[37] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 85-100.

[38] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Learning pyramid-context encoder network for high-quality image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1486-1494.

[39] H. Liu, B. Jiang, Y. Song, W. Huang, and C. Yang, "Rethinking image inpainting via a mutual encoder-decoder with feature equalizations," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow*, UK, August 23–28, 2020, Proceedings, Part II 16, 2020: Springer, pp. 725-741.

[40] M. Zhu, D. He, X. Li, C. Li, F. Li, X. Liu, E. Ding, and Z. Zhang, "Image inpainting by end-to-end cascaded refinement with mask awareness," *IEEE Transactions on Image Processing*, vol. 30, pp. 4855-4866, 2021.

[41] N. Wang, Y. Zhang, and L. Zhang, "Dynamic selection network for image inpainting," *IEEE Transactions on Image Processing*, vol. 30, pp. 1784-1798, 2021.

[42] L. Liao, J. Xiao, Z. Wang, C.-W. Lin, and S. i. Satoh, "Guidance and evaluation: Semantic-aware image inpainting for mixed scenes," in *Computer Vision–ECCV 2020: 16th European Conference*, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVII 16, 2020: Springer, pp. 683-700.

[43] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi, "Edgeconnect: Structure guided image inpainting using edge prediction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2019, pp. 0-0.

[44] O. Ronneberger, P. Fischer, and 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, 2015: Springer, pp. 234-241.

[45] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness," *arXiv preprint* arXiv:1811.12231, 2018.

[46] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint* arXiv:1511.07122, 2015.

[47] Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia, "Image inpainting via generative multi-column convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 31, 2018.

[48] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint* arXiv:2010.11929, 2020.

[49] J. Lee-Thorp, J. Ainslie, I. Eckstein, and S. Ontanon, "Fnet: Mixing tokens with fourier transforms," *arXiv preprint* arXiv:2105.03824, 2021.

[50] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 834-848, 2017.

[51] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint* arXiv:1706.05587, 2017.

[52] X. Yan, J. Gao, J. Li, R. Zhang, Z. Li, R. Huang, and S. Cui, "Sparse single sweep lidar point cloud segmentation via learning contextual shape priors from scene completion," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021, vol. 35, no. 4, pp. 3101-3109.

[53] B. Graham, M. Engelcke, and L. Van Der Maaten, "3d semantic segmentation with submanifold sparse convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 9224-9232.

[54] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. icml*, 2013, vol. 30, no. 1: Atlanta, GA, p. 3.

[55] H. Caesar, V. Bankiti, A. Lang, S. Vora, V. Erin Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11621-11631.

[56] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012: IEEE, pp. 3354-3361.

[57] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Rrecognition*, 2017, pp. 1125-1134.

[58] H. Drucker and Y. Le Cun, "Improving generalization performance using double backpropagation," *IEEE Transactions on Neural Networks*, vol. 3, no. 6, pp. 991-997, 1992.

[59] H. Zhou, X. Zhu, X. Song, Y. Ma, Z. Wang, H. Li, and D. Lin, "Cylinder3d: An effective 3d framework for driving-scene lidar semantic segmentation," *arXiv preprint* arXiv:2008.01550, 2020.

[60] W. Song, J. Zhu, and R. Zhang, "Meta-rangeseg: Lidar sequence semantic segmentation using multiple feature aggregation." in *IEEE Robotics and Automation Letters* 7, no. 4, 2022: 9739-9746.

[61] S. Wen, T. Wang, and S. Tao, "Hybrid cnn-lstm architecture for lidar point clouds semantic segmentation," in *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 5811–5818, 2022.

[62] I. Alonso, L. Riazuelo, L. Montesano, and A. C. Murillo, "3d-mininet: Learning a 2d representation from point clouds for fast and efficient 3d lidar semantic segmentation," *IEEE Robotics and*

Automation Letters, vol. 5, no. 4, pp. 5432-5439, 2020.

[63] Z. Ce, and Q. Ling, "GAF-Net: Geometric Contextual Feature Aggregation and Adaptive Fusion for Large-Scale Point Cloud Semantic Segmentation," in *IEEE Transactions on Geoscience and Remote Sensing* 61 2023: 1-15.

[64] C. Hui-Xian, X. Han, and G. Xiao, "TransRVNet: LiDAR semantic segmentation with transformer," in *IEEE Transactions on Intelligent Transportation Systems* 24, no. 6, 2023: 5895-5907.

[65] Z. Yangheng, J. Wang, X. Li, Y. Hu, C. Zhang, Y. Wang, and S. Chen. "Number-adaptive prototype learning for 3d point cloud semantic segmentation." In *European Conference on Computer Vision*, pp. 695-703. Cham: Springer Nature Switzerland, 2022.

[66] W. Xuzhe, J. Yang, Z. Kang, J. Du, Z. Tao, and D. Qiao, "A category-contrastive guided-graph convolutional network approach for the semantic segmentation of point clouds," in *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 16, 2023: 3715-3729.

[67] L. Xiaohang, and J. Zhou, "MASNet: Road semantic segmentation based on multi-scale modality fusion perception," in *IEEE Transactions on Instrumentation and Measurement,* 2023.

[68] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[69] Q. Hu, B. Yang, L. Xie, S. Rosa, Y. Guo, Z. Wang, N. Trigoni, and A. Markham, "Randla-net: Efficient semantic segmentation of large-scale point clouds," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11108-11117.

[70] Y. Zhang, Z. Zhou, P. David, X. Yue, Z. Xi, B. Gong, and H. Foroosh, "Polarnet: An improved grid representation for online lidar point clouds semantic segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9601-9610.

[71] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint* arXiv:1710.10196, 2017.

[72] X. Guo, H. Yang, and D. Huang, "Image inpainting via conditional texture and structure dual generation," In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* pages 14134–14143, 2021.

[73] Y. Yu, F. Zhan, S. Lu, J. Pan, F. Ma, X. Xie, and C. Miao, "Wavefill: A wavelet-based generation network for image inpainting," in *Proceedings of the IEEE/CVF international conference on computer vision, 2021,* pp. 14 114–14 123.

[74] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pages 10684– 10695, June 2022.

[75] R. Zhang, W. Quan, Y. Zhang, J. Wang, and D. Yan, "W-net: Structure and texture interaction for image inpainting," in *IEEE Transactions on Multimedia,* 2022.

[76] X. Li, Q. Guo, D. Lin, P. Li, W. Feng, and S. Wang, "Misf: Multilevel interactive siamese filtering for high-fidelity image inpainting," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* pages 1869– 1878, 2022.

[77] K. Ko and C. Kim, "Continuously masked transformer for image inpainting," In *Proceedings of the IEEE/CVF International Conference on Computer Vision,* pages 13169–13178, 2023.

[78] C. Shuang, A. Atapour-Abarghouei, H. Zhang, and H. Shum, "MxT: Mamba x Transformer for Image Inpainting." in *arXiv preprint arXiv:2407.16126,* 2024.

[79] Y. Song, J. Sohl-Dickstein, D.P. Kingma, A. Kumar, S. Ermon, and B. Poole, "Score-based generative modeling through stochastic differential equations," In *arXiv presprint arXiv:2011.13456*, 2020.

[80] Y. Zeng, J. Fu, H. Chao, and B. Guo. "Aggregated contextual transformations for high-resolution image inpainting," in *IEEE Transactions on Visualization and Computer Graphics*, 2022, pp.3266-3280.

[81] A. Lugmayr, , M. Danelljan, A. Romero, F. Yu, R. Timofte, and L. Van Gool, "Repaint: Inpainting using denoising diffusion probabilistic models," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 11461-11471.

[82] Z. Wan, J. Zhang, D. Chen, and J. Liao, "High-fidelity pluralistic image completion with transformers," In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4692-4701.

[83] J. Peng, D. Liu, S. Xu, and H. Li, "Generating diverse structure for image inpainting with hierarchical VQ-VAE," In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10775-10784.

[84] S.A. Hussein, T. Tirer, and R. Giryes, "Image-adaptive GAN based reconstruction," In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020, Vol. 34, No. 04, pp. 3121-3129.

Abdollah Amirkhani (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees (Hons.) in electrical engineering from the Iran University of Science and Technology (IUST), Tehran, in 2012 and 2017, respectively. He is currently an Associate Professor with the School of Automotive Engineering, IUST. He has been actively involved in several national research and development projects, related to the development of new methodologies and learning algorithms based on AI techniques. His research interests are in machine vision, fuzzy cognitive maps, autonomous vehicle, data mining, and machine learning. He earned the Outstanding Student Award form the First Vice President of Iran in 2015. In 2016, he was conferred award by the Ministry of Science, Research and Technology. He is an Associate Editor of the Engineering Science and Technology, an International Journal.

Mahdi Khouri Shandiz received the B.Sc. degree in electronic engineering from Technical and Vocational University (TVU), Mashhad, Iran, in 2019, and he is currently a M.Sc. student at Iran University of Science & Technology (IUST), Tehran, Iran. His research interests include machine vision, machine learning, deep learning, image processing, and simultaneous localization and mapping algorithm, with applications in intelligent transportation systems and autonomous vehicle.