RSBEV: Multi-view Collaborative Segmentation of 3D Remote Sensing Scenes with Bird's-Eye-View Representation

Baihong Lin, Zhengxia Zou, *Senior Member, IEEE* and Zhenwei Shi*, *Senior Member, IEEE*,

***Abstract*—Perception of 3D remote sensing scenes plays a crucial role in accurately recognizing and locating ground objects, as it enables a deeper understanding of complex environments by capturing scene geometry, object relationships, and occlusion patterns. Inspired by the powerful multi-sensor fusion capabilities in autonomous driving, we explore a new task in this paper: given a set of multi-view images of a 3D remote sensing scene, we aim to obtain bird's-eye view (BEV) scene information under the common view area in the world coordinate system. In this work, we focus on the task of semantic segmentation to demonstrate the feasibility of our approach and introduce a BEV modeling technique tailored for remote sensing scenes, which facilitates the projection of 3D scene details from multiple perspective views onto a bird's-eye view. We then utilize a dual-encoder structure based on the Vision Transformer (VIT) architecture to extract relevant spatial information using self-attention mechanisms. Within the decoder, we employ a Feature Pyramid Network (FPN) to integrate BEV patch encoding with spatial feature residuals, enabling fine-grained segmentation results at the original input resolution. Furthermore, we curated the LEVIR-MDS multi-drone segmentation dataset, comprising scenes from 10 community-level areas across 3 continents, totaling 243k images and their corresponding annotated BEV semantic maps, amounting to approximately 500GB. This dataset serves as a robust benchmark to assess the effectiveness and generalization capability of our proposed method. To our best knowledge, this is the first semantic segmentation dataset designed specifically for collaborative multi-drone applications. We further show that our method achieves a 12% improvement in mIoU, reaching 69.73%, compared to a pure convolutional network model.**

***Index Terms*—Multi-view collaborative segmentation, remote sensing, Bird's-Eye-View (BEV) representation, semantic segmentation**

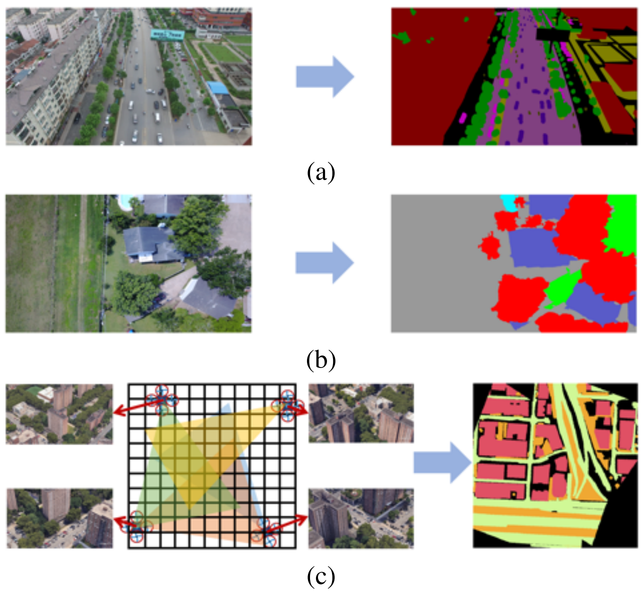
Fig. 1: Visual comparison between existing drone semantic segmentation framework and our proposed framework. (a) Captured from a side-view perspective and annotated using pixel-to-pixel labeling [1]. (b) Captured from a top-view perspective and annotated using pixel-to-pixel labeling [2]. (c) Our proposed framework, which learns scene 3D information from multiple side-view perspectives and performs semantic segmentation on BEV at absolute scale.

I. Introduction

MULTI-VIEW stereoscopic perception is vital in remote sensing tasks for applications such as disaster assessment [2, 3], resource monitoring [4, 5], and land surveying [6]. While drone-based remote sensing provides advantages over satellite imaging, including flexible path planning and fine-grained perception, most existing remote sensing approaches rely on pixel-based segmentation, overlooking the richness of multi-view and 3D information [7].

With the advancement of convolutional neural networks (CNNs), the performance of semantic segmentation has significantly improved. Existing drone-based remote sensing semantic segmentation can be categorized into two types based on the input perspective: semantic segmentation for side-view perspectives [1, 8] and semantic segmentation for top-view perspectives [2, 3, 9]. Side-view perspectives encompass more spatial information and perception coverage. However, they face challenges such as object occlusion at different heights, requiring semantic segmentation in a pixel-to-pixel space after perspective transformation, making it difficult to map to world coordinates. Top-view perspectives largely resolve occlusion issues and allow direct target localization in world coordi-

The work was supported by the National Key Research and Development Program of China under Grant 2022ZD0160401, the National Natural Science Foundation of China under Grant 62125102 and 62471014, the Beijing Natural Science Foundation under Grant JL23005, and the Fundamental Research Funds for the Central Universities. *(Corresponding author: Zhenwei Shi (e-mail: shizhenwei@buaa.edu.cn))*

Baihong Lin and Zhenwei Shi are with the Image Processing Center, School of Astronautics, Beihang University, and with the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University, Beijing 100191, China, and also with Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China. Zhengxia Zou is with Department of Guidance, Navigation and Control, School of Astronautics, Beihang University, Beijing 100191, China, and also with Shanghai Artificial Intelligence Laboratory, Shanghai 200232, China.

nates. However, their perception range is fixed, with a single viewpoint that limits capturing spatial scene information.

In recent years, within the realm of autonomous driving (e.g., Tesla [10]), pertinent methods have adeptly integrated multi side-view sensor data by creating a Bird's Eye View (BEV) representation. This method involves projecting and integrating information captured by side-view sensors around the vehicle into a top-down view. The process facilitates a comprehensive depiction of surrounding scenes in the world coordinate system, providing an absolute scale and leveraging 3D information perceived from side-view perspectives.

To project side-view information into BEV perspective, methods can be categorized into two approaches. Some methods directly map side-view features to BEV space, such as [11–14], by extracting 2D image features from side views, predicting depth distribution, and then projecting and integrating them into BEV space. Others like [15, 16] use depth estimation to predict side-view depth maps and extract features in 3D space. These methods share a common goal of recovering depth information lost in the 2D imaging process of side views and associating it with BEV space. Alternatively, some methods utilize camera models for inverse perspective mapping directly in BEV space. For example, [17] first utilized this method in 3D object detection, assuming targets lie on a plane in 3D space and querying point-by-point for corresponding features from various viewpoints. Furthermore, [18, 19] built mappings between BEV space points and side-view images based on a variational encoder-decoder and MLP architecture, while [20, 21] utilized transformers to directly construct BEV queries and employ cross-attention mechanisms to search for corresponding features in perspective images.

However, these methods build on assumptions made for autonomous driving tasks, where the objective is limited to perceiving the immediate surroundings of a vehicle. This allows for treating the ground as a nearly flat plane to facilitate straightforward projections between perspective views and the bird's-eye view. In drone scenarios, by contrast, the task generally involves a much broader scope of scene perception, rendering the simplistic flat plane assumption impractical. Moreover, unlike in autonomous driving, the extensive freedom in drone trajectories and the multiscale nature of scenes demand a specialized BEV-based framework designed for drone applications. Therefore, we introduce a robust network framework, RSBEV, tailored for multi-view collaborative segmentation. Accompanying this framework, we present a substantial dataset named LEVIR-MDS, which spans 10 community-level scenes across three continents, encompassing 243k images with corresponding annotated BEV semantic maps, totaling approximately 500GB.

In this paper, we apply Inverse Perspective Mapping (IPM) within BEV framework to comprehensively extract all relevant 2D data and employ self-attention mechanisms to meticulously filter and ascertain the accurate 3D information. Initially, we model the ground height under a normal distribution assumption using a multi-plane approach and extract BEV representation of the scene through weighted pooling. Subsequently, we propose a dual Encoder structure based on the Vision Transformer (VIT) [22] architecture to filter out irrelevant spatial information using self-attention mechanisms. Finally, in the decoder, we design a Feature Pyramid Network (FPN) [23] to connect BEV patch encoding with spatial feature residuals, enabling the acquisition of finely segmented bird's-eye view images at the input resolution.

The contributions of our work can be summarized as follows:

1) We propose a multi-view joint perception framework designed for 3D remote sensing scenes. Given a set of multi-view images, our aim is to accurately derive BEV scene information within the shared viewing area of the world coordinate system. Our method stands out by leveraging complex 3D scene data and drone collaboration, ensuring precise perception of the environment and we utilize the task of semantic segmentation to validate the efficacy of our framework.
2) A Novel Algorithm (RSBEV): We propose the RSBEV method to achieve multi-view collaborative semantic segmentation. Initially, we project 3D scene information from multiple perspectives into BEV space using BEV modeling. Additionally, we design a dual Encoder structure based on the Vision Transformer (VIT) architecture, utilizing self-attention mechanisms to extract relevant spatial information. In the decoder phase, we employ a Feature Pyramid Network (FPN) to connect BEV patch encoding with spatial feature residuals, enabling finely segmented bird's-eye view results at the input resolution. These innovative algorithmic designs enable our method to effectively address multi-view collaborative scene segmentation tasks with drones.
3) A New Dataset (LEVIR-MDS): We curate a comprehensive multi-view collaborative segmentation dataset, which includes scenes from 10 community-level areas across three continents, featuring a total of 243k images and their corresponding annotated bird's-eye view (BEV) semantic maps, totaling approximately 500GB. This dataset provides essential experimental resources and serves as a significant benchmark in the drone remote sensing field. It not only validates the effectiveness of our proposed RSBEV method but also fosters extensive research and development in this domain.

The rest of the paper is organized as follows. Section II reviews related work in BEV representation and semantic segmentation. Section III details our proposed method. Section IV presents the details of LEVIR-MDS. Section V provides experimental results demonstrating the effectiveness and rationale of our method. Finally, we conclude in Section VI.

II. Related Work

A. Vision-centric BEV Representation

Vision-centric BEV representation, with its capacity to seamlessly integrate 3D information at a real-world scale, has increasingly captured the attention of both industry and academia [24]. The earliest works in this area date back over 30 years, introducing methods that project side-view perspectives onto BEV under rigid plane assumptions [25].

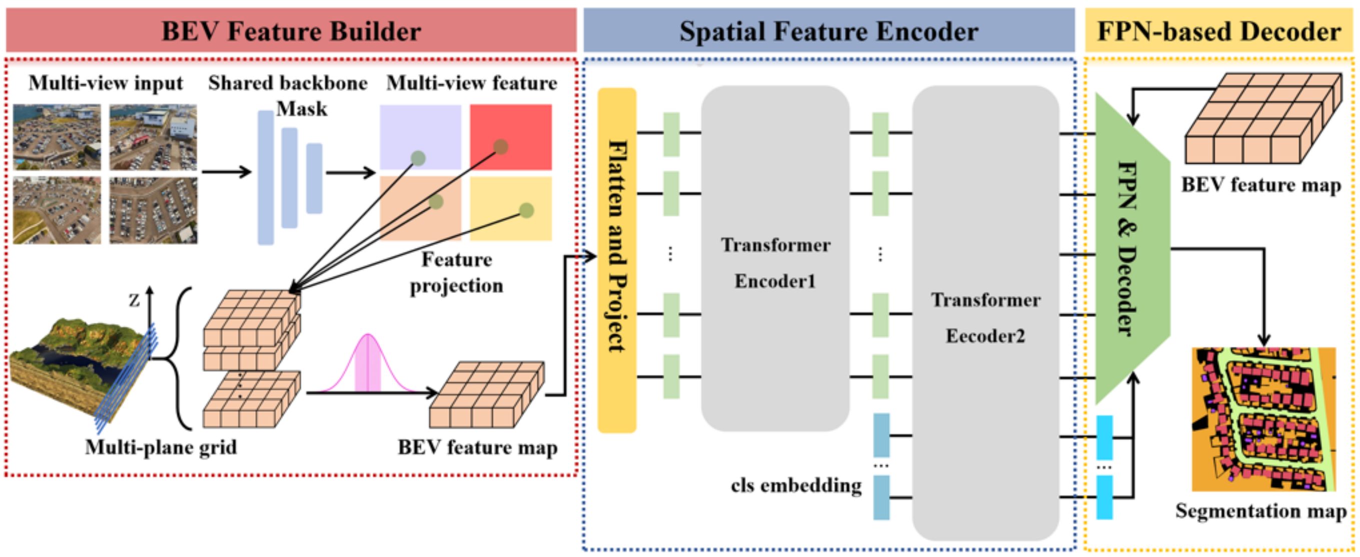

Fig. 2: An overview of our method. We take multi-view images as input. In the BEV Feature Builder, multi-view feature is projected onto a multi-plane grid within the world coordinates, followed by a weighted pooling to obtain the BEV feature map. Subsequently, in the Spatial Feature Encoder, the BEV feature map is used to learn spatial and category-related information. Finally, the FPN-based Decoder predicts the Segmentation map in BEV space.

With advancements in deep learning, several methods [7, 26–28] have utilized BEV representation to transform features from 2D perspective pixel coordinates to 3D world coordinates in a bird's-eye view.

To mitigate severe deformation and occlusion issues during rigid transformation processes, some methods have introduced Generative Adversarial Network (GAN) [29] to compensate for errors in projection. For instance, BriighGAN [30] takes three views as inputs: perspective view, homography view, and bird's-eye view. It employs multiple sub-models based on a multi-GAN architecture to address significant viewpoint differences and severe deformations between perspective views and bird's-eye view. MonoLayout [31] predicts BEV layouts, leveraging adversarial feature learning to hallucinate scene details for occluded image regions. This approach tackles the challenge of estimating layouts beyond visible information and compensates for 3D information loss due to projection.

However, these direct transformation methods still suffer from incomplete projections due to occlusions and errors caused by the planar assumption. To address these fundamental issues, [11, 12, 15, 16] leverage depth prediction or the probability distribution of depths from 2D side-view images to determine the position of feature vectors in the BEV space. Nonetheless, directly predicting missing depth information increases model complexity and hinders convergence. Alternatively, CaDNN [32] incorporates supervision using depth information obtained directly from lidar during depth estimation, leading to a more precise transformation from perspective view to BEV.

Meanwhile, some methods continue to pursue accurate transformation processes exclusively through visual cues.VED [18] employs learnable parameters to implicitly express the mapping from perspective view to BEV, utilizing a variational encoder-decoder and MLP architecture to extract semantic information in 3D space directly from monocular images.Building on this approach, [19, 21] integrate features from multiple perspective views into BEV space, aggregating spatial information from various viewpoints on the BEV feature map. While these methods effectively mitigate errors in BEV modeling of non-flat terrain such as mountains or buildings, they may encounter overfitting issues due to the network's requirement to fit additional scene-specific geometric details. Notably, HFT [33] devised a hybrid model that combines geometric transformation features with camera parameters and learnable features, surpassing previous methods in performance.However, further enhancements are necessary to optimize performance when integrating multi-view information [24].

The advent of Vision Transformer (VIT) [22] has brought remarkable advancements in applying transformer-based models to computer vision tasks. Certain transformer-based approaches have surpassed MLP-based methods and consistently demonstrate strong performance in multi-view fusion scenarios. In the field of autonomous driving, [20, 21, 34] leveraging cross-attention mechanisms effectively retrieve image features from perspective views to map onto the BEV. In such methods, the primary trade-off lies in the use of sparse queries [35–37], which sacrifices spatial geometric structure awareness, while dense queries [20, 38] require exceedingly high computational complexity.

In this article, addressing the characteristics of drone in large-scale scene perception. By employing inverse perspective mapping combined with a VIT-based architecture [22], we filter out inaccurately projected spatial information, achieving a balance between computational complexity and accuracy.

B. Semantic Segmentation Using Encoder-Decoder Architecture

The encoder-decoder architecture framework is widely employed in pixel-level segmentation tasks. In this framework, the encoder typically reduces the spatial scale of the input image and maps semantic information from the spatial domain to high-dimensional feature vectors. Meanwhile, the decoder performs upsampling to map these feature vectors back to category information. FPN [23] was first introduced using the VGG network as its encoder, where the original fully connected layers were replaced with a decoder capable of restoring resolution. To ensure multiscale spatial information, feature maps of different scales from the encoder were directly fused with corresponding layers of the decoder, preserving spatial features at various resolutions. Subsequently, more convolutional network architectures such as ResNet [39], HRNet [40], and ResNeSt [41] have also been used as encoder. However, due to pooling operations in encoder and continuous upsampling in decoder, it remains challenging to learn high-frequency details in images. U-Net [42] and its subsequent variants [43–46] introduced several shortcuts in the encoder-decoder structure, further preserving multiscale information, and found wide applications in image segmentation in medical and remote sensing domains [47]. A similar U-shaped network structure was adopted in [47], incorporating deconvolution layers in the decoder to extract roads in drone remote sensing scenes. Additionally, SegNet [48] systematically decodes encoder features layer by layer, preserving indices in max pooling to consider contextual information and maintain overall object-level consistency during segmentation. More recent research, Spatial-Spectral Masked Auto-encoder (MAE) [49], extends the encoder-decoder paradigm by focusing on masked image modeling, where certain portions of the input are masked, and the model is trained to reconstruct those hidden parts. This approach enhances the model's ability to capture spatial and spectral relationships, which is particularly valuable in remote sensing tasks [50]. MAE's masked reconstruction strategy leverages the rich spatial-spectral information in remote sensing data, contributing to improved scene classification and feature extraction.

The exceptional global modeling capabilities of Transformer [51] have been extended to encoder-decoder-based image segmentation frameworks in research. For example, SETR [52] follows the paradigm of VIT [22] in the encoder, splitting the image into patches and leveraging transformers to learn global information. The output tokens are then reassembled into feature maps and processed with upsampling and convolution in the decoder. On the other hand, Segmenter [53] is built entirely on transformer-based encoder-decoder structures.In decoding process, the authors replace convolutional architecture with transformer that includes a category token, and ultimately compute the similarity between output tokens and category tokens using dot products. Segformer [54] employs lightweight self-attention mechanisms and hierarchical feature maps in encoder, while leveraging MLP structures in decoder to effectively reduce computational complexity. More recently, methods such as [55] have embedded Swin Transformer [56] into U-Net [42], resulting in improved segmentation accuracy for small-scale objects.

III. PROPOSED METHOD

The overall process of our RSBEV method is illustrated in the flowchart shown in Fig. 2. This method comprises three main components: 1) **BEV Feature Builder**, which projects features from input side-view perspectives to obtain Bird's Eye View (BEV) representations. 2) **Spatial Feature Encoder**, which filters out irrelevant spatial information from the BEV representations and models the correlation between features and categories. 3) **FPN-based Decoder**, which decodes the output features by integrating multi-scale information to obtain category prediction results.

A. BEV Feature Builder

In this subsection, we efficiently projects image features from multiple perspectives into BEV space. First, we extract features from multi-perspective images using a shared backbone network. Subsequently, these features are explicitly projected onto a multi-plane grid in the world coordinates using the camera models. This process allows us to obtain the BEV feature map through weighted pooling.

We first process multi-view images $\mathbf{I} \in \mathbb{R}^{N\times 3\times H\times W}$ and utilize their corresponding camera parameters $\mathbf{M} \in \mathbb{R}^{N\times 4\times 4}$ with a ResNet-like [39] network (NET) to extract image features. To effectively integrate spatial information, a depth coordinate map $\mathbf{Z} \in \mathbb{R}^{N\times 1\times H\times W}$, derived from $\mathbf{M}$, is introduced. This map, representing the depth for each viewpoint, quantifies the distance from the observed scene to the world coordinate plane at $z = 0$. It is concatenated with the extracted features, and a convolutional operation is applied to generate a per-pixel mask. This mask assesses the relative importance of image features at specific locations. The feature extraction process can be formalized as:

$$\mathbf{F}_{2D} = \text{Net}(\mathbf{I}) \tag{1}$$

$$\mathbf{mask} = \text{conv}(\text{concat}(\mathbf{F}_{2D}, \mathbf{Z})) \tag{2}$$

$$\mathbf{F}_{\text{mask2D}} = \mathbf{F}_{2D} \odot \text{sigmoid}(\mathbf{mask}) \tag{3}$$

Where $\mathbf{F}_{2D} \in \mathbb{R}^{N\times \text{feature\_h}\times \text{feature\_w}\times C}$ denotes the image features directly obtained from the feature extraction network, and $\mathbf{F}_{\text{mask2D}} \in \mathbb{R}^{N\times \text{feature\_h}\times \text{feature\_w}\times C}$ represents the image features weighted by the viewpoint-specific masks, where C is the dimension of the image features.

It is noteworthy that a significant challenge arises due to the unavailability of ground truth elevation values at arbitrary points within the scene. Directly projecting features onto the BEV plane from ground level often results in inaccurate feature representations. To address this issue, we model the ground height distribution within the scene as a Gaussian distribution $\mathcal{N}(\mu, \sigma)$, where μ represents the average scene height z_{world}, and we set $\sigma = 1.0$. Specifically, we discretize the height into a series of bins, dividing the space into $2L - 1$ parallel planes, equally spaced with elevations: $\mathbf{z} = \{z_1, z_2, \ldots, z_L, \ldots, z_{2L-2}, z_{2L-1}\}$, where $z_L = \mu$. In our experiments, we set $L = 4$ with a height spacing of

5 meters, resulting in a total of 7 planes. For each height z_i, a grid-shaped BEV map is created. Using the camera projection model, we search for 2D image features that can be projected from each grid point onto the BEV plane. For a given BEV grid with coordinate (x, y, z), the projection process is mathematically described as follows:

$$\mathbf{p} = M_{\text{trans}}(\mathbf{M}, x, y, z) \tag{4}$$

Where $\mathbf{p} \in \mathbb{R}^{N \times 2}$ includes feature pixel coordinates of all cameras that can be projected from the coordinates (x, y, z). Further, we filter out points within the coordinate range of each camera's feature map, along with their corresponding features:

$$\mathbf{p}_{\text{valid}} = \{\mathbf{p}_i \in \mathbf{p} \mid \text{check\_bounds}(\mathbf{p}_i, H_{\text{feature}}, W_{\text{feature}})\} \tag{5}$$

$$\mathbf{F}_{3D}[x, y, z] = \mathbf{F}_{\text{mask2D}}(\mathbf{p}_{\text{valid}}) \tag{6}$$

The function check_bounds verifies if the coordinates $\mathbf{p}_i$ fall within the feature map dimensions $[0, H_{\text{feature}}) \times [0, W_{\text{feature}})$. After filtering, $\mathbf{p}_{\text{valid}} \in \mathbb{R}^{n_{\text{hit}} \times 2}$ contains only those coordinates that meet this criteria, where $n_{\text{hit}} \in [0, N)$ denotes the count of viewpoints that can project their features within this coordinate range by the grid point. If $n_{\text{hit}} = 0$, then $\mathbf{F}_{3D}[x, y, z]$ is a zero vector. Otherwise, $\mathbf{F}_{3D}[x, y, z] \in \mathbb{R}^{n_{\text{hit}} \times C}$ represents the projected feature set corresponding to this BEV grid.

In the above process, the transformation $M_{\text{trans}}(\mathbf{M}, x, y, z)$ adheres to the pinhole camera model, where the 3D point (x, y, z) in the world coordinate system is projected onto a 2D pixel plane through the intrinsic and extrinsic camera parameters incorporated within the matrix $\mathbf{M}$. The projection mechanism is defined by the following equation:

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \\ \mathbf{1} \end{pmatrix} = \mathbf{K} \begin{pmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & \mathbf{1} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \tag{7}$$

In this equation, $(\mathbf{u}, \mathbf{v}) \in \mathbb{R}^{N \times 2}$ corresponds to the previously defined $\mathbf{p}$, $\mathbf{K}$ signifies the cameras' intrinsic matrix, $\mathbf{R}$ the rotation matrix, and $\mathbf{t}$ the translation vector. These components collectively compose the cameras' extrinsic matrix. The complete matrix $\mathbf{M}$ is thus formulated as the product of $\mathbf{K}$ and the concatenated extrinsic matrix $\begin{pmatrix} \mathbf{R} & \mathbf{t} \end{pmatrix}$.

Once we have obtained the complete BEV multi-plane features $\mathbf{F}_{3D}(x, y, z) \in \mathbb{R}^{n_{\text{hit}} \times C}$, we compute the averaged features for each grid point in the BEV map as follows:

$$\mathbf{F}[x, y, z] = \begin{cases} \frac{1}{n_{\text{hit}}} \sum_{i=1}^{n_{\text{hit}}} \mathbf{F}_{3D}[x, y, z][i] & \text{if } n_{\text{hit}} > 0 \\ \mathbf{0} & \text{if } n_{\text{hit}} = 0 \end{cases} \tag{8}$$

Where $\mathbf{F} \in \mathbb{R}^{\text{bev\_h} \times \text{bev\_w} \times (2L-1) \times C}$ represents the BEV map across different layers, providing a comprehensive representation of the scene at various heights.

Finally, we employ weighted pooling to aggregate $\mathbf{F}$ according to the discrete Gaussian distribution probabilities, yielding the final BEV feature map $\mathbf{B} \in \mathbb{R}^{\text{bev\_h} \times \text{bev\_w} \times C}$:

$$\mathbf{B} = \sum_{i=0}^{2L-1} \frac{1}{\sigma\sqrt{2\pi}} \exp - \frac{(z_i - z_L)^2}{2\sigma^2} \cdot \mathbf{F}[:, :, i, :] \tag{9}$$

Algorithm 1 BEV Feature Building Process. The symbols used in this pseudocode follow the same naming conventions as in the previous formulas.

```
Input: Multi-view images $\mathbf{I}$, Camera parameters $\mathbf{M}$
Input: BEV grid dimensions $(\text{bev\_h}, \text{bev\_w}, L)$
Output: Final BEV feature map $\mathbf{B}$
1: Step 1: Extract multi-view features.
2: $\textbf{mask} \leftarrow \texttt{conv}(\texttt{concat}(\texttt{Net}(\mathbf{I}), \mathbf{Z}))$
3: // $\mathbf{Z}$: Depth map computed from $\mathbf{M}$
4: $\mathbf{F}_{\text{mask2D}} \leftarrow \texttt{Net}(\mathbf{I}) \odot \texttt{sigmoid}(\textbf{mask})$
5:
6: Step 2: For each grid point in the BEV map, calculate
   projected feature.
7: for each grid point $(x, y, z)$ in BEV grid do
8:     $\mathbf{p} \leftarrow M_{\text{trans}}(\mathbf{M}, x, y, z)$
9:     $\mathbf{p}_{\text{valid}} \leftarrow \texttt{check\_bounds}(\mathbf{p}, \mathbf{F}_{\text{mask2D}}\text{.shape()})$
10:    if $\mathbf{p}_{\text{valid}} \neq \emptyset$ then
11:        $\mathbf{F}_{3D}(x, y, z) \leftarrow \mathbf{F}_{mask2D}(\mathbf{p}_{\text{valid}})$
12:    else
13:        $\mathbf{F}_{3D}(x, y, z) \leftarrow \mathbf{0}$
14:    end if
15: end for
16:
17: Step 3: Feature fusion and weighted pooling.
18: for each grid point $(x, y, z)$ in BEV grid do
19:    if $n_{\text{hit}} > 0$ then
20:        $\mathbf{F}[x, y, z] \leftarrow \frac{1}{n_{\text{hit}}} \sum_{i=1}^{n_{\text{hit}}} \mathbf{F}_{3D}[x, y, z][i]$
21:    else
22:        $\mathbf{F}[x, y, z] \leftarrow \mathbf{0}$
23:    end if
24: end for
25: $\mathbf{B} \leftarrow \texttt{WeightedPooling}(\mathbf{F}$, Gaussian weights)
26: return $\mathbf{B}$
```

B. Spatial Feature Encoder

We observe that the BEV feature map $\mathbf{B}$ obtained through pooling contains overlapping features from different elevations. Consequently, directly applying convolutional architectures for feature encoding and decoding would struggle to capture effective contextual information due to the presence of excessive invalid information in the BEV feature map. Subsequent experiments also validate the presence of severe overfitting under such conditions. To address this issue, we propose a transformer-based architecture inspired by VIT [22], which enables global modeling of spatial information through self-attention mechanisms. The entire process is divided into two encoders, Encoder1 and Encoder2, utilizing self-attention modules with identical structures. Furthermore, considering that the pixel-wise operations during the projection of image features may negate the regularization effect induced by data augmentation in subsequent encoders, we incorporate additional data augmentation on the BEV feature map $\mathbf{B}$ following the approach proposed in BEVDet [13], involving operations such as rotation, cropping, resizing, and flipping.

In Encoder1, we divide the BEV map into a series of 8×8 patches, which are flattened and fed into a linear layer to

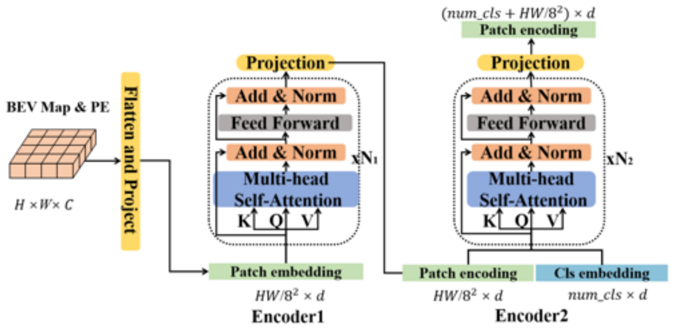

Fig. 3: Network architecture of the Spatial Feature Encoder. In Encoder1, the network receives patch embedding with positional embedding and utilizes self-attention mechanisms to filter out irrelevant spatial information in the input. Subsequently, the output is concatenated with class embedding and fed into Encoder2 to learn the dependencies between features and class information.

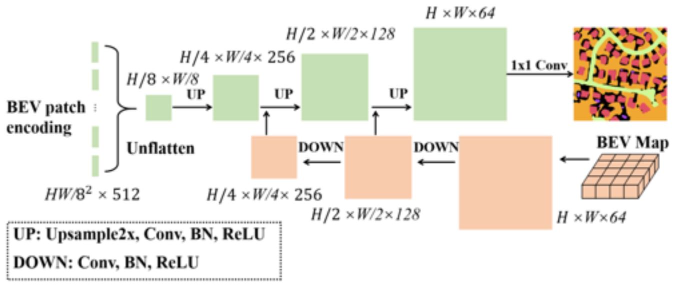

Fig. 4: Decoder Network Architecture Diagram. This diagram illustrates how the feature sequence processed by the encoder is rearranged into a 2D feature map and subsequently refined through three upsampling modules to produce the final segmentation results. During the upsampling process, feature maps of various resolutions are fused with the downsampled BEV feature map to enhance segmentation performance.

obtain BEV feature tokens. These tokens, along with positional embeddings, are then input to Encoder1, comprising N_1 self-attention modules aimed at filtering out irrelevant spatial information. The output dimensions remain the same as the input BEV patch encoding. Subsequently, these encodings are combined with category embeddings and fed into Encoder2, which consists of N_2 self-attention modules designed to learn the dependencies between BEV feature encoding and category information. Finally, Encoder2 outputs the BEV patch encoding corresponding to each patch.

As shown in Fig. 3, the same self-attention operation is utilized in different encoders. $\mathbf{Q}$, $\mathbf{K}$, $\mathbf{V}$ denote query, key and value, and input tokens ($\mathbf{Q}$, $\mathbf{K}$, V) are linearly projected using embedding tokens or encoding tokens T, represented as:

$$\mathbf{Q} = \mathbf{T}\mathbf{W}^q, \quad \mathbf{K} = \mathbf{T}\mathbf{W}^k, \quad \mathbf{V} = \mathbf{T}\mathbf{W}^v, \tag{10}$$

where $\mathbf{W}^q$, $\mathbf{W}^k$, $\mathbf{W}^v$ are the learnable parameters of the linear projection layer. In Encoder1, they have dimensions $HW/8^2 \times d$, and in Encoder2, their dimensions are $(HW/8^2 + num\_cls) \times d$. Moreover, num_cls represents the number of classes, and d denotes the feature dimension of the tokens. After obtaining $\mathbf{Q}$, $\mathbf{K}$, and$\mathbf{V}$, one attention head can be expressed as:

$$\text{SelfAttn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V} \tag{11}$$

C. FPN-based Decoder

Although our encoder contains intermediate features with precise spatial relations, the network struggles to capture information about small-scale objects. This is because each encoding vector at the output of the Spatial Feature Encoder describes only the category information within each patch. Directly upsampling and computing the inner product with the class encoding, as done in [53], results in the loss of small-scale information at this stage. To address this issue, we introduce FPN [23] into the segmentation output head to fuse multi-scale information. By employing both upscaling UP modules and downscaling DOWN modules, we fuse the BEV map before the Spatial Feature Encoder with the feature maps concatenated with BEV patch encodings, enabling multi-scale fusion. This ultimately yields category segmentation results in the BEV space at the original scale. The process is illustrated in Fig. 4.

D. Loss Function

Our loss function consists of three components: weighted cross-entropy loss L_{ce}, dice loss [57] L_{dice}, and inter-class diversity loss L_{cls}. Assuming C is the number of classes, N is the number of samples, y_{ij} denotes the ground truth label (0 or 1) of sample i belonging to class j, and p_{ij} is the probability predicted by the model that sample i belongs to class j.

L_{ce} can be represented as follows:

$$L_{ce} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C} w_j \cdot y_{ij} \cdot \log(p_{ij}) \tag{12}$$

Where w_j is the weight assigned to class j, which is set inversely proportional to the pixel count of each class to further balance the class distribution in the samples.

L_{dice}can be represented as follows:

$$L_{dice} = 1 - \frac{1}{C}\sum_{j=1}^{C} \frac{2 \cdot \sum_{i=1}^{N} p_{ij} \cdot y_{ij}}{\sum_{i=1}^{N} p_{ij} + \sum_{i=1}^{N} y_{ij}} \tag{13}$$

L_{cls} can be represented as follows:

$$L_{cls} = -\frac{1}{C(C-1)}\sum_{i=1}^{C}\sum_{j=1, j\neq i}^{C} \left| \frac{\mathbf{u}_i}{\|\mathbf{u}_j\|} - \frac{\mathbf{u}_i}{\|\mathbf{u}_j\|} \right| \tag{14}$$

Where $\mathbf{u}_i$ and $\mathbf{u}_j$ are both class encoding output by the encoder. Minimizing this ensures that features from different classes have distinct separability in the feature space, thereby enhancing the ability to distinguish between different classes. The final loss function can be represented as:

$$L = L_{ce} + L_{dice} + \lambda L_{cls} \tag{15}$$

The weight of the L_{cls}, denoted by λ, is set to 0.1 in this case.

E. Network Details

TABLE I: Detailed configuration of our shared backbone. Here, Conv-DOWN represents a convolutional layer followed by batch normalization, rectified linear unit (ReLU), and max pooling; Conv-UP represents a convolutional layer followed by ReLU and upsampling layer.

Layer	Config	Input Dim	Output Dim	Resolution
Layer0	Conv-Down	3	64	1/4
Layer1	Resnet.layer1	64	256	1/4
Layer2	Resnet.layer2	256	512	1/8
Layer3	Resnet.layer3	512	1024	1/16
Layer4	Resnet.layer4	1024	2048	1/32
Conv_FPN1	Conv-UP+Layer4	2048	1024	1/16
Conv_FPN2	Conv-UP+Layer3	1024	512	1/8
Conv_FPN3	Conv-UP+Layer2	512	256	1/4
Conv_FPN4	Conv-UP+Layer1	256	64	1/2
Conv_mask	Concat-Conv	64+1	1	1/2
Scalar_product	Conv_mask $\odot$ Conv_FPN4	64&1	64	1/2

1) BEV Feature Builder: As shown in Table I, we employ a ResNet-50-like [39] backbone network to extract image features. The average pooling and fully connected layers are removed and replaced with an upsampling network structured as a FPN [23], with corresponding feature maps being skip connected. The output image feature size is half of the input image size, with feature channels set to $C = 64$.

The BEV feature map $\mathbf{B}$ was set to have dimensions $512 \times 512 \times 64$, corresponding to a spatial resolution of $0.8\,m \times 0.8\,m$, effectively covering a perception range of $409.6\,m \times 409.6\,m$. In the multi-plane projection part of the BEV Feature Builder, we set $L = 4$ with a height spacing of 5 meters, resulting in a total of 7 planes, and the perception of information within a 30-meter height range.

2) Transformer Encoder: We divide the input image into a series of 8×8 patches, and then project each patch into a 256-dimensional feature vector using a linear transformation (with an 8×8 convolutional kernel and a stride of 8). After incorporating positional embedding, these feature vectors serve as the input to the transformer. Additionally, in TabVII, we obtain models with different parameter counts by modifying the settings in the encoder. For instance, in RSBEV-base, we set the number of layers in both Encoder1 and Encoder2 to 6, the number of attention heads to 4, and the dimension of the Feed Forward layer to 512.

3) Segmentation Head: We reorganize the patch encodings outputted by the encoder into a shape of$\frac{H}{8} \times \frac{W}{8} \times 256$. Then, leveraging the FPN [23] structure depicted in Fig. 4, we progressively fuse BEV features with patch encodings at different scales. Finally, employing a 1×1 convolution, we predict categories at the original scale of the BEV feature map.

IV. Dataset

Multi-view drone datasets have made significant progress in the field of object tracking [58, 59], yet there are still substantial gaps in the segmentation domain. In the realm of autonomous driving, several large-scale datasets such as nuScenes [60], KITTI [61], and the Waymo Open Dataset (WOD) [62] have been established, providing a wealth of data for various tasks. To validate the effectiveness of RSBEV, we drew inspiration from both dataset creation methods in the drone domain and dataset formats prevalent in the autonomous driving domain, and proposed a comprehensive dataset named **LEVIR-MDS**, specifically designed for multi-view collaborative segmentation.

This dataset leverages real-world 3D scenes generated from Google Earth Studio [63], comprising 10 extensive community-level scenes across 3 continents. Each scene is captured from four unique drone viewpoints, resulting in a comprehensive collection of 5800 frames per viewpoint. These frames have been rigorously annotated to categorize six distinct features: building, road, vegetation, water, boat, and other, demonstrating the substantial effort involved in dataset preparation. Representative images from the dataset are presented in Fig. 5, while the corresponding annotated bird's-eye view images are illustrated in Fig. 6.

Our LEVIR-MDS offers a robust foundation for advancing research in multi-view collaborative segmentation, bridging the existing gaps in the current drone segmentation datasets. By leveraging the high-resolution and detailed nature of the simulated 3D scenes, researchers can develop and validate new algorithms and methodologies aimed at enhancing segmentation accuracy and efficiency in multi-drone systems.

A. Trajectory Setup

For each scene measuring approximately $4\text{km} \times 4\text{km}$, the area is typically divided into 3 rows and 5 columns, and scanned along a serpentine trajectory through these subregions as illustrated in Fig. 7. During the scanning process, the camera centers of the four trajectory cameras in each frame converge on the same point, with the cameras facing forward, backward, leftward, and rightward, respectively. In our dataset, the BEV center position for each frame is determined by a meticulously designed set of trajectories, ensuring that the central position is as much as possible within the common view area of the four viewpoints.

Scanning is performed along the row direction with a 500-frame interval between two adjacent rows (3 rows in total) and along the column direction with a 200-frame interval between two adjacent columns (5 columns in total), all at a frame rate of 24 fps. These speeds were chosen to balance between detailed temporal resolution and practical data collection rates. This results in a sequence of 4×5800 temporal images for each scene, where each image measures 1600×900 pixels. The horizontal field of view (HFOV) for each image is $20°$.

The height and tilt angle of the drones vary within specific ranges to simulate realistic flight conditions. Specifically, the height ranges from 100 meters to 500 meters, and the tilt angle varies between 25 degrees and 50 degrees to capture different perspectives and enhance the dataset's diversity.

B. Data Pairing

For scene i, we only annotate to obtain the BEV ground truth map Mask_i. In frame t, the visible area of the ground truth map P^t in this scene is defined as:

$$P^t = \bigcup_{n=1}^{N} \left(\bigcup_{p \in \text{Mask}_i} p = (u, v) \middle| M_{\text{trans}}(p) \in \text{grid}(I_n) \right) \quad (16)$$

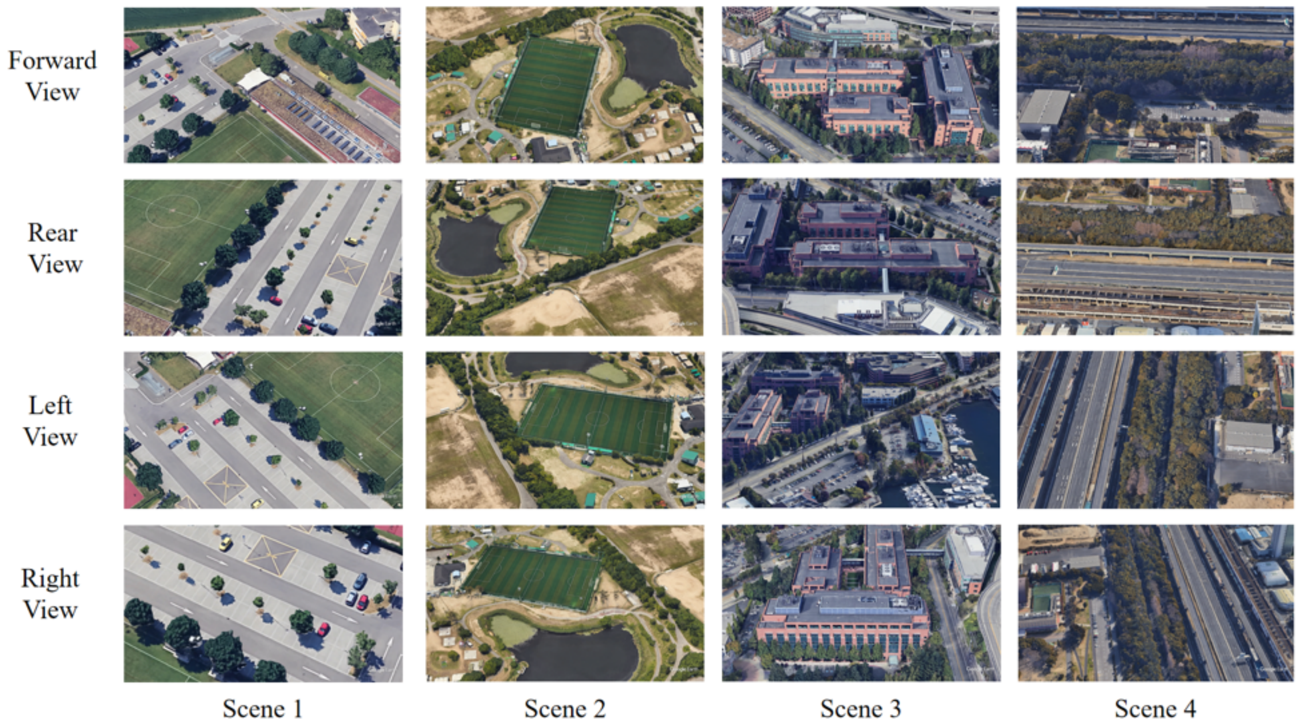

Fig. 5: Sample images of the first four scenes from the dataset

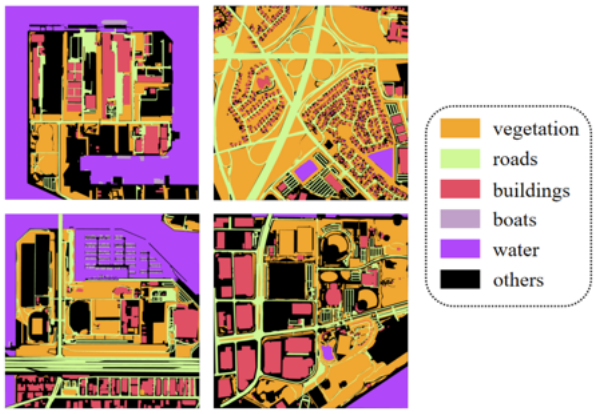

Fig. 6: Annotated BEV map of the first four scenes from the dataset

where N represents the number of cameras, $p = (u, v)$ denotes the pixel coordinates in Mask_i, $\text{grid}(I_n)$ denotes the tensor of pixel coordinates in the n-th viewpoint, and $M_{\text{trans}}(\cdot)$ denotes the projection from BEV to side view transformation using the pinhole camera model.

Thus, we obtain the data pairs of the side-view images $\mathbf{I}_i^t$ and the corresponding ground truth mask $\text{mask}_i^t = \text{Mask}_i(P^t)$ in the BEV space, as illustrated in Fig. 1(c).

C. Dataset Format

The dataset structure is inspired by the nuScenes [60] dataset. It organizes data in directories named "sample", "sweep", and "mask", with metadata stored in "data_meta.json".

1) Directory Structure:

- **sample**: Stores keyframe images from different side-view perspectives in JPEG format. Each keyframe represents significant moments in each trajectory and is stored with a corresponding timestamp.
- **sweep**: Contains time-series sequences of images captured between keyframes from different side-view perspectives, also in JPEG format, providing a continuous view of each trajectory.
- **mask**: Contains the ground truth images for each scene, annotated with six categories. The annotated ground truth images are stored in PNG format, along with the corresponding camera parameter information.
- **data_meta.json**: Contains a dictionary that records the paths of images and corresponding camera parameters for all trajectories in each scene. The dictionary structure allows easy access to the dataset's metadata.

2) Statistical Properties: Key statistical properties of the dataset include:

- **Scenes Information**:The dataset comprises a total of 10 scenes sourced from three different continents. Specific details about each scene, including coordinates, area, average height, number of frames, and main categories, are provided in Tab.II.
- **Number of Images**: Each scene contains a sequence of $4 \times 5,800$ to $4 \times 8,500$ frames captured from four distinct drone viewpoints, culminating in a total of 242,800 images.
- **Image Resolution**: Each image has a resolution of 1600×900 pixels.
- **Categories**: The ground truth annotations include six

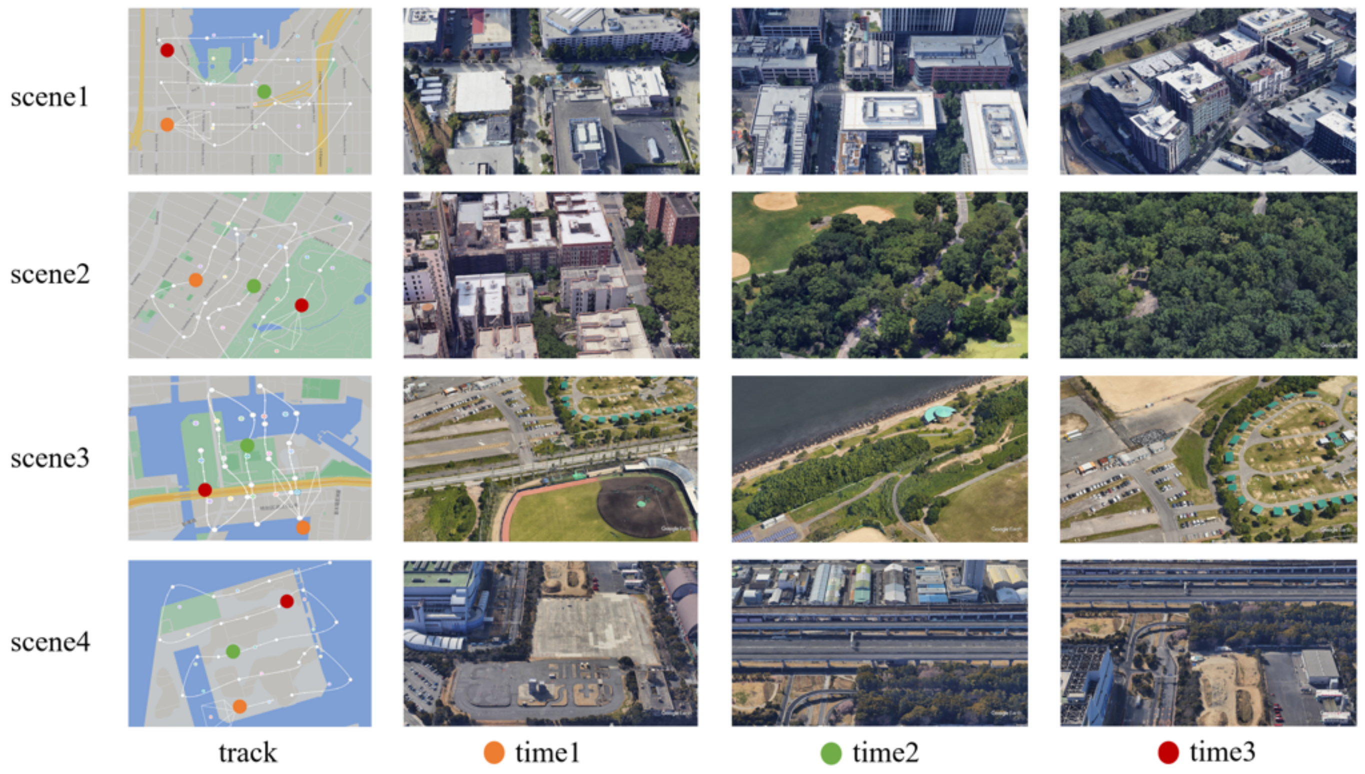

Fig. 7: Trajectories of the first four scenes and forward-looking images at different times. The leftmost images show the camera trajectories (white curves) from a forward-looking perspective for four scenes. White dots indicate the key points selected during the trajectory design, with camera poses for the remaining frames obtained using cubic spline interpolation for a smooth trajectory. The orange, green, and red dots represent the camera's position at three specific moments, corresponding to the images shown in the right three columns.

TABLE II: Detailed information of each scene

Scene	latitude&longitude	Area (km^2)	Avg Height (m)	Frames	Main Categories	Region
Scene 1	47.62, -122.34	11	205.69	5800	Building, Ship	Washington, USA
Scene 2	40.79, -73.96	12	238.31	5800	Vegetation, Building	New York, USA
Scene 3	34.67, 135.40	14	297.13	5800	Vegetation, Building	Tokyo, Japan
Scene 4	35.65, 139.83	18	396.75	5800	Vegetation, Ship	Yokohama, Japan
Scene 5	35.45, 139.96	10	170.13	5800	Building	Yokohama, Japan
Scene 6	40.79, -74.12	14	244.24	5800	Vegetation, Water	New York, USA
Scene 7	40.80, -73.49	16	235.01	5800	Building, Roads	New York, USA
Scene 8	40.79, -74.01	15	245.62	5800	Vegetation, Water	Brooklyn, USA
Scene 9	47.46, 8.58	8	112.28	8500	Vegetation, Building	Zurich, Switzerland
Scene 10	47.630, -122.346	10	265.09	5800	Road, Others	British Columbia, Canada

categories: building, road, vegetation, water, boat, and other. The distribution of these categories across the dataset is illustrated in Fig. 8.

- **Altitude Distribution**: Each scene's four trajectories are meticulously designed with a specific altitude distribution, ensuring a consistent shared viewing area among different drone perspectives throughout their motion. This design also facilitates the occurrence of image occlusions among various viewpoints. Altitude curves in the first scene are illustrated in Fig. 9.

V. Experiment

A. Experimental Setup

1) Implementation Details: We perform training on 8 scenes, validation on 1 scene, and testing on a separate scene. Our network is deployed under the PyTorch framework. We utilize a single NVIDIA GeForce RTX 4090 24-GB GPU. We employ the Adam optimizer with a learning rate of 5e-4 and a weight decay of 1e-8. The learning rate scheduler is configured to decay by 0.99 after each epoch. A batch size of 1 is utilized, and the maximum number of epochs is set to 100.

2) Evaluation Metrics: We employ IoU to assess the classification performance of each category and use the average F1 score to evaluate the overall segmentation effectiveness of the network.

The IoU for each category i is calculated using the formula:

$$\text{IoU}_i = \frac{TP_i}{TP_i + FP_i + FN_i} \tag{17}$$

Where TP_i epresents the true positives, FP_i represents the false positives, and FN_i represents the false negatives for

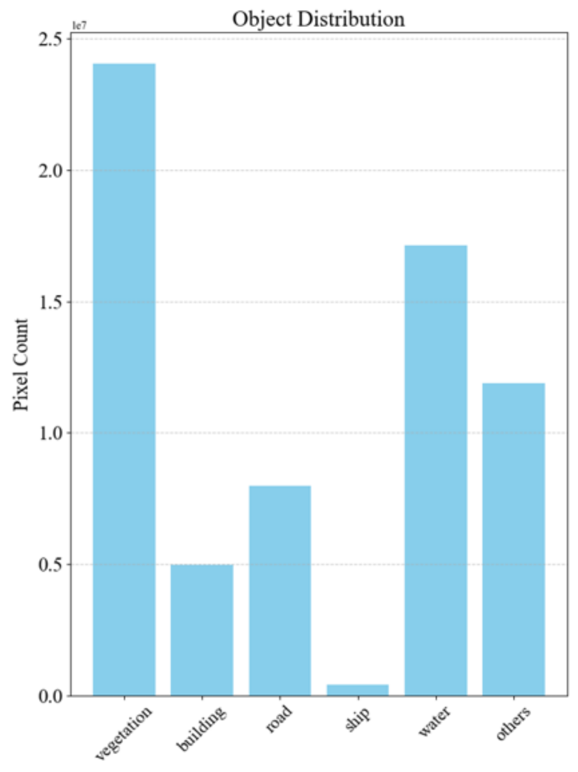

Fig. 8: Pixel count distribution of annotated categories

category i.

The average F1 score is calculated using the formula:

$$\mathrm{F1}_{avg} = \frac{1}{C}\sum_{i=1}^{C}\frac{2\cdot \mathrm{Precision}_i \cdot \mathrm{Recall}_i}{\mathrm{Precision}_i + \mathrm{Recall}_i} \tag{18}$$

Where $\mathrm{Precision}_i$ is given by $\frac{TP_i}{TP_i+FP_i}$ and Recall_i is given by $\frac{TP_i}{TP_i+FN_i}$.

B. Comparative Experiments

To further strengthen the evidence supporting the effectiveness of our proposed method and the newly introduced dataset, LEVIR-MDS, we conducted comparative experiments using two representative camera-based methods: Cam2BEV [64] and LSS [11]. We also designed a purely convolutional network for comparison with our method.

Cam2BEV [64] addresses the challenge of estimating distances in monocular camera images by transforming the camera perspective into a corrected BEV image. It uses the uNetXST model to generate a 360° BEV image from multiple vehicle-mounted cameras, making predictions for occluded areas and generalizing well to real-world data using synthetic training data, outperforming traditional IPM. In our experiments, we retained the core architecture of the uNetXST model used in Cam2BEV but adapted the input perspective to the drone's viewpoint.

LSS [11] explicitly predicts the depth distribution from monocular images and uses this information to construct precise 3D features, enabling better performance in BEV tasks. The model predicts a depth-aware feature map that aligns with the BEV space, allowing for more accurate scene understanding. In our experiments, we adapted LSS by replacing the input perspective with the drone's viewpoint and modified the output head to match the multi-class segmentation structure used in our method.

In our CNN-based model, the ViT [22] structure used in the spatial encoder-decoder of our original network was replaced with a convolutional architecture, while keeping the rest of the pipeline intact, to validate the applicability of our dataset across different model architectures.

The results of the comparison between Cam2BEV, LSS, our convolutional network, and our proposed method are shown in Table III. As demonstrated by the table, the results not only highlight the effectiveness of our method but also demonstrate the robustness of our dataset in evaluating different approaches across various methods.

C. Visualization

We visualize the classification results on the test set, as shown in Fig. 10. In Fig. 10(a), it can be observed that even with occlusion and non-uniform scales among input views, the output can still achieve accurate bird's-eye-view segmentation results at the given resolution. For example, the image depicts partially obscured houses and a swimming pool due to tree cover. In contrast, the CNN architecture model without the transformer encoder fails to extract effective spatial information, leading to significant overfitting issues. Consequently, crucial occlusion details such as the swimming pool are often lost in the side view. In Fig. 10(b), rural roads and some small buildings are well predicted by the model. These are capabilities not possessed by existing methods in the field of autonomous driving. However, in Fig. 10(c), the segmentation performance of our model for densely distributed small boats still needs improvement.

D. Ablation Study

To validate the effectiveness of the three proposed modules in our network, we conducted ablation experiments using a pure convolutional network as the baseline, in which the BEV multi-plane projection and spatial encoder-decoder components were directly removed, and the output head was simplified by eliminating the FPN method. Additionally, we developed several variants of the model by incrementally adding different components (Model-v1 to Model-v5) to assess their contributions. The results are shown in TableIV.

When the multi-plane projection and weighted pooling modules were removed, the network only demonstrated certain perception capabilities for large-area features such as vegetation and water bodies, while its perception capabilities for buildings and roads significantly decreased. This indicates that our BEV modeling strategy effectively projects spatial information, enhancing the perception capabilities for height-sensitive areas (Building) and space-sensitive areas (Road). Incorporating FPN into the decoder resulted in improved performance for categories such as buildings and boats, but a slight decrease in performance for road categories, which

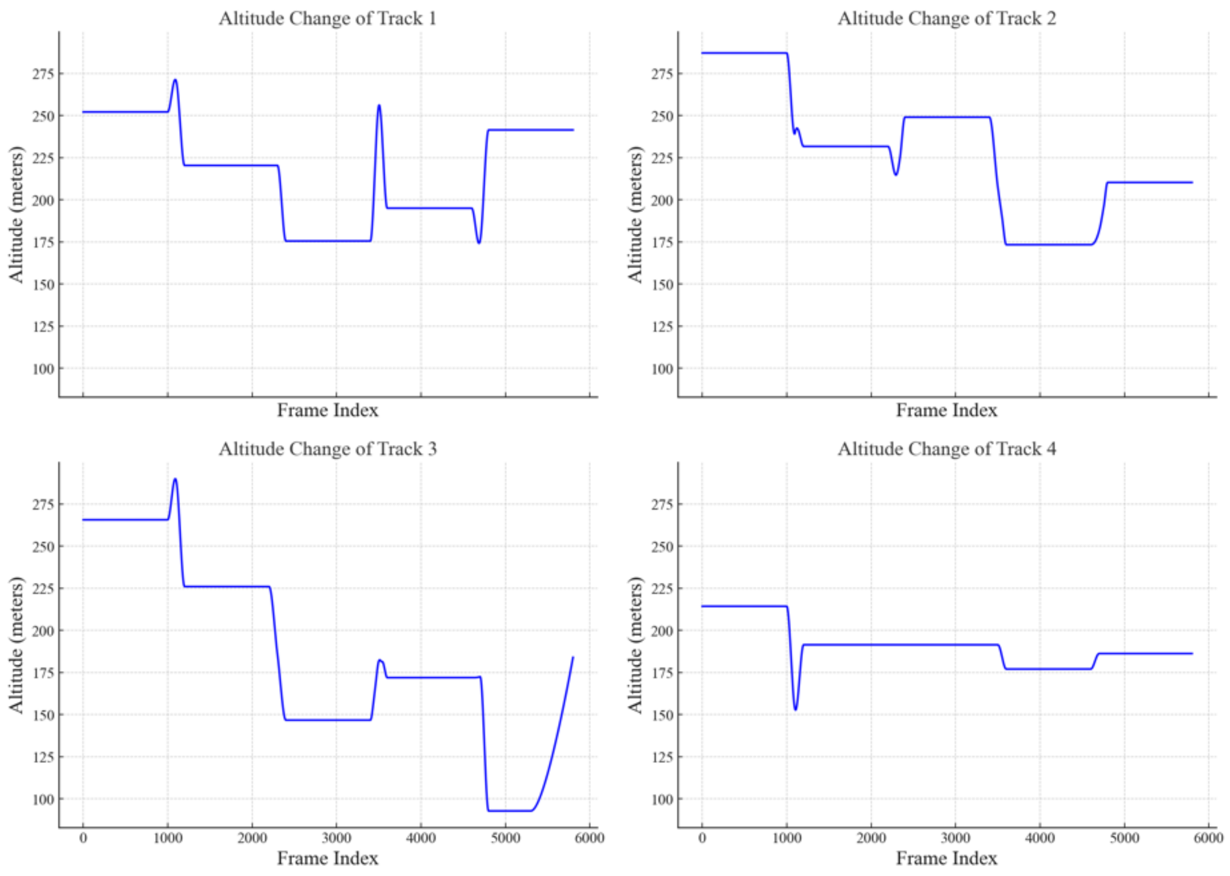

Fig. 9: Altitude curves for the four tracks in the first scene

TABLE III: Comparison of performance between Cam2BEV, LSS, our convolutional network, and our proposed method. The mIoU and F1 scores are the average values across six classes, including “other” category.

Method	**IoU_vegetation**	**IoU_building**	**IoU_road**	**IoU_ship**	**IoU_water**	**mIoU↑**	**F1↑**
Cam2BEV [64]	0.5338	0.4105	0.5023	0.7368	0.6847	0.6135	0.4886
LSS [11]	0.3514	0.4793	**0.5102**	0.7639	0.5770	0.5518	0.3932
CNN	0.4327	0.4279	0.4413	0.6114	0.7082	0.5736	0.4025
Ours	**0.5931**	**0.6536**	0.5077	**0.8324**	**0.8362**	**0.6973**	**0.5910**

we attribute to a trade-off in small-scale target detection and information complexity.

Additionally, we found that adding the class differentiation loss function L_{cls} to the loss function improved detection performance, indicating that this measure helps amplify differences in features between different categories. Experimental results are presented in TableV.

E. Model Robustness Evaluation

In this study, we conducted a series of experiments to evaluate the impact of different numbers of camera views on the performance of multi-view collaborative segmentation tasks. First, we trained the network using data from four viewpoints to ensure that the network could fully learn multi-view information under optimal conditions. Then, during the testing phase, we performed experiments using 3 or 4 viewpoints as the baseline and randomly reduced one viewpoint to simulate scenarios where some drones are occluded or some cameras are unavailable, thereby recording the changes in segmentation performance (such as mIoU and F1 scores). This process helps us understand the segmentation performance and stability of the network under different numbers of viewpoints.

To further improve the robustness of the network, we introduced a random camera dropout strategy during the training process. In each training batch, we randomly closed one viewpoint or kept the original input to enhance the network’s adaptability to different combinations of viewpoints. Subsequently, we employed the same testing method to evaluate the effect of this strategy and verified the improvement in network stability by comparing the experimental results. The experimental results, as shown in TableVI, indicate that the network performance gradually improves with an increasing number of viewpoints. Moreover, by randomly closing some viewpoints during training, the network demonstrated stronger robustness and stability when facing viewpoint loss.

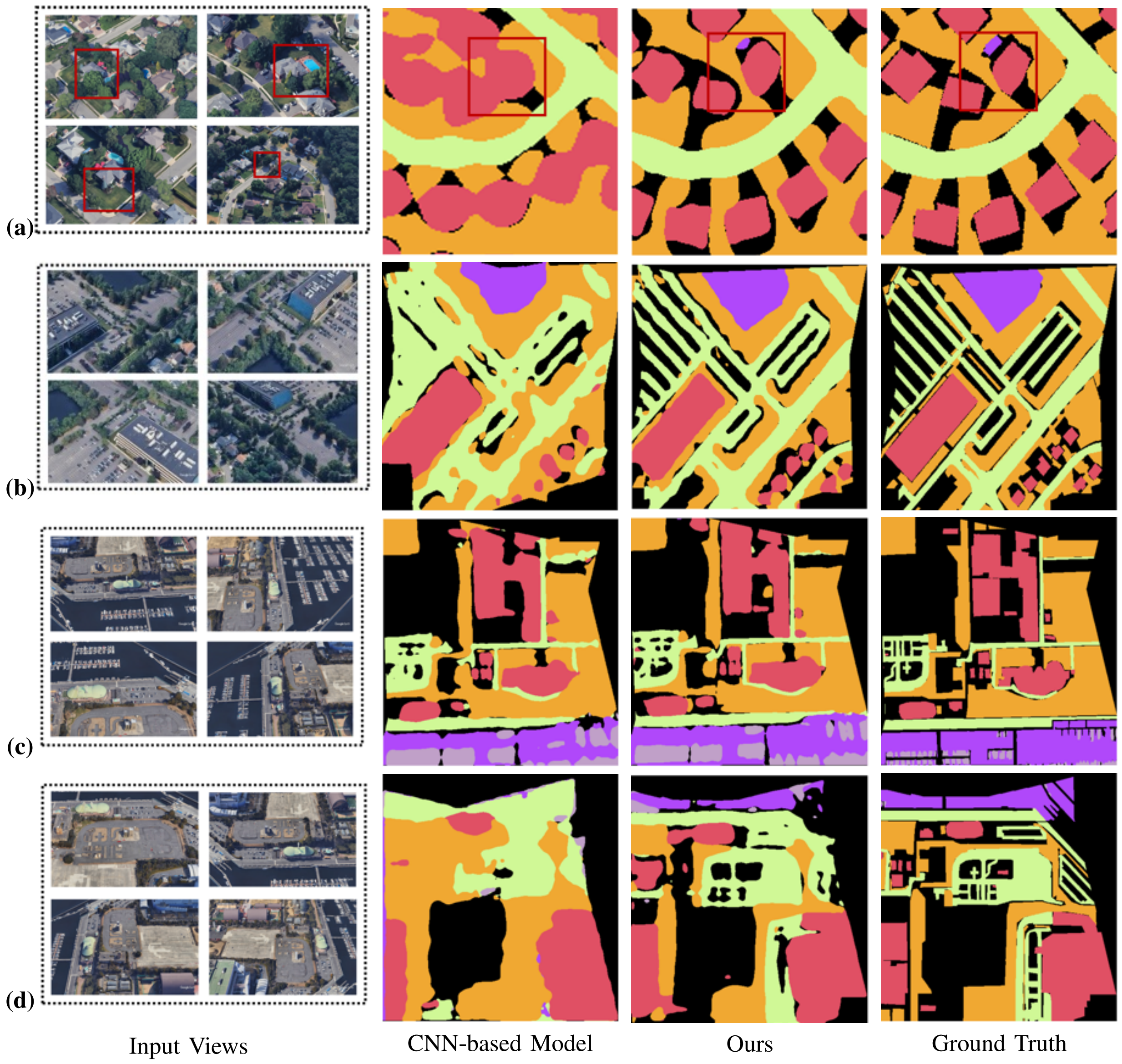

Fig. 10: Visualization of segmentation results between our model and the CNN-based model

F. Efficiency and Effectiveness

Based on the current model, we obtained four models with different parameter sizes by modifying the dimensions of feature vectors in different modules and the number of layers in the Transformer. For comparative experiments, we also used a pure convolutional architecture with 46 million parameters. Additionally, we tested different models using the same computational resources, and Table 4 presents their parameter counts, floating-point operations per second (FLOPs), mean IoU (mIoU), and F1 scores.

We found that our models outperformed the pure convolutional network with the same parameter count (42 million) and that performance improved with an increase in the number of Transformer layers.

VI. Conclusion

In summary, our work presents RSBEV, a novel approach for multi-view collaborative semantic segmentation using bird's-eye-view (BEV) representation. Leveraging multi-view side-view images under BEV space supervision, our framework improves scene understanding by utilizing 3D scene information and drone collaborative capabilities. Key contributions include a new task framework that outperforms traditional methods, a novel algorithm with BEV modeling and self-attention mechanisms, and a new dataset for validation. Experiments showed the robustness and adaptability of our model, highlighting the importance of multi-plane projection, weighted pooling, and transformer encoder. Evaluation of model robustness, including varying the number of viewpoints and implementing a random camera dropout strategy, demonstrated the network's stability under suboptimal conditions.

This work marks a significant step towards advancing drone-

TABLE IV: Ablation studies of different technical components of the proposed method.

Name	MPP&Pooling	Encoder1	Encoder2	FPN	IoU_vegetation	IoU_building	IoU_road	IoU_ship	IoU_water	mIoU↑	F1 ↑
Base	✗	✗	✗	✗	0.4175	0.5086	0.3914	0.4110	0.6283	0.4826	0.3475
Model-v1	✗	✓	✓	✓	0.5848	0.4614	0.4482	0.8279	0.8001	0.6293	0.5028
Model-v2	✓	✓	✗	✗	0.4472	0.5839	0.4974	0.3782	0.7658	0.5974	0.4213
Model-v3	✓	✗	✓	✗	0.4381	0.6042	0.4907	0.3044	0.7623	0.6010	0.4918
Model-v4	✓	✓	✓	✗	0.5914	0.6169	**0.5149**	0.5239	0.7739	0.6851	0.5887
Model-v5	✓	✗	✗	✓	0.4420	0.6002	0.3426	0.6817	0.6785	0.6003	0.4876
Ours	✓	✓	✓	✓	**0.5931**	**0.6536**	0.5077	**0.8324**	**0.8014**	**0.6973**	**0.5910**

TABLE V: Ablation studies of loss functions configurations.

L_{ce}	L_{dice}	L_{cls}	mIoU	F1
✓	✗	✗	0.6738	0.5860
✓	✗	✓	0.6804	0.6284
✓	✓	✗	0.6841	0.6403
✓	✓	✓	**0.6973**	**0.5910**

based semantic segmentation and holds promise for applications across various real-world scenarios.

REFERENCES

[1] Y. Lyu, G. Vosselman, G. Xia, A. Yilmaz, and M. Y. Yang, "Uavid: A semantic segmentation dataset for uav imagery," 2020.

[2] M. Rahnemoonfar, T. Chowdhury, A. Sarkar, D. Varshney, M. Yari, and R. Murphy, "Floodnet: A high resolution aerial imagery dataset for post flood scene understanding," 2020.

[3] M. Rahnemoonfar, T. Chowdhury, and R. Murphy, "Rescuenet: A high resolution uav semantic segmentation dataset for natural disaster damage assessment," *Scientific Data*, vol. 10, no. 1, Dec. 2023. [Online]. Available: http://dx.doi.org/10.1038/s41597-023-02799-4

[4] Q. Wang, Z. Yuan, Q. Du, and X. Li, "Getnet: A general end-to-end 2-d cnn framework for hyperspectral image change detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 1, pp. 3–13, 2019.

[5] X. Zhang, J. Jin, Z. Lan, C. Li, M. Fan, Y. Wang, X. Yu, and Y. Zhang, "Icenet: A semantic segmentation deep network for river ice by fusing positional and channel-wise attentive features," *Remote Sensing*, vol. 12, no. 2, p. 221, 2020.

[6] N. Zang, Y. Cao, Y. Wang, B. Huang, L. Zhang, and P. T. Mathiopoulos, "Land-use mapping for high-spatial resolution remote sensing image via deep learning: A review," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 5372–5391, 2021.

[7] A. Palazzi, G. Borghi, D. Abati, S. Calderara, and R. Cucchiara, "Learning to map vehicles into bird's eye view," 2017.

[8] N. Jiang, K. Wang, X. Peng, X. Yu, Q. Wang, J. Xing, G. Li, J. Zhao, G. Guo, and Z. Han, "Anti-uav: A large multi-modal benchmark for uav tracking," 2021.

[9] M. R. Heffels and J. Vanschoren, "Aerial imagery pixel-level segmentation," 2020.

[10] "Tesla AI Day," https://www.youtube.com/watch?v=j0z4FweCy4M, 2021.

[11] J. Philion and S. Fidler, "Lift, splat, shoot: Encoding images from arbitrary camera rigs by implicitly unprojecting to 3d," 2020.

[12] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," 2021.

[13] J. Huang, G. Huang, Z. Zhu, Y. Ye, and D. Du, "Bevdet: High-performance multi-camera 3d object detection in bird-eye-view," 2022.

[14] Y. Li, Z. Ge, G. Yu, J. Yang, Z. Wang, Y. Shi, J. Sun, and Z. Li, "Bevdepth: Acquisition of reliable depth for multi-view 3d object detection," 2022.

[15] Y. Wang, W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar from visual depth estimation: Bridging the gap in 3d object detection for autonomous driving," 2020.

[16] Y. You, Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger, "Pseudo-lidar++: Accurate depth for 3d object detection in autonomous driving," 2020.

[17] T. Roddick, A. Kendall, and R. Cipolla, "Orthographic feature transform for monocular 3d object detection," 2018.

[18] C. Lu, M. J. G. van de Molengraft, and G. Dubbelman, "Monocular semantic occupancy grid mapping with convolutional variational encoder–decoder networks," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, p. 445–452, Apr. 2019. [Online]. Available: http://dx.doi.org/10.1109/LRA.2019.2891028

[19] B. Pan, J. Sun, H. Y. T. Leung, A. Andonian, and B. Zhou, "Cross-view semantic segmentation for sensing surroundings," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, p. 4867–4873, Jul. 2020. [Online]. Available: http://dx.doi.org/10.1109/LRA.2020.3004325

[20] Z. Li, W. Wang, H. Li, E. Xie, C. Sima, T. Lu, Q. Yu, and J. Dai, "Bevformer: Learning bird's-eye-view representation from multi-camera images via spatiotemporal transformers," 2022.

[21] S. Gong, X. Ye, X. Tan, J. Wang, E. Ding, Y. Zhou, and X. Bai, "Gitnet: Geometric prior-based transformation for birds-eye-view segmentation," 2022.

[22] 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," 2021.

[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," 2015.

[24] Y. Ma, T. Wang, X. Bai, H. Yang, Y. Hou, Y. Wang, Y. Qiao, R. Yang, D. Manocha, and X. Zhu, "Vision-centric bev perception: A survey," 2023.

[25] H. A. Mallot, H. H. Bülthoff, J. J. Little, and S. Bohrer, "Inverse perspective mapping simplifies optical flow computation and obstacle detection," *Biological Cybernetics*, vol. 64, no. 3, pp. 177–185, 1991. [Online]. Available: https://doi.org/10.1007/BF00201978

[26] A. Abbas and A. Zisserman, "A geometric approach to obtain a bird's eye view from an image," 2020.

[27] Y. Kim and D. Kum, "Deep learning based vehicle position and orientation estimation via inverse perspective mapping image," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 317–323.

[28] Y. Hou, L. Zheng, and S. Gould, "Multiview detection with feature perspective transformation," 2021.

[29] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

TABLE VI: Performance metrics under different training and testing camera counts and random dropout strategy.

Training Cameras	Testing Cameras	Random Dropout Strategy	mIoU	F1
4	4	✗	0.6973	0.5910
4	3	✗	0.5942	0.5016
4	3	✓	0.6533	0.5627
3	3	✗	0.6801	0.5840
3	2	✗	0.5707	0.4735
3	2	✓	0.6191	0.5203

TABLE VII: Performance and Computational Analysis of Different Models.

Name	Backbone	Num_head	Transformer Layer	BEV_dim	Encoder_dim	Params (M)	GFLOPs	mIoU	F1
RSBEV-T	ResNet-18	4	3	32	32	17	108.3	0.4898	0.3103
RSBEV-S	ResNet-50	4	3	32	64	42	167.5	0.6783	0.5629
RSBEV-B	ResNet-50	4	6	64	256	60	184.2	0.6973	0.5910
RSBEV-L	ResNet-50	8	12	256	1024	124	203.4	0.7105	0.5923
CNN-based model	ResNet-50	/	/	64	/	46	116.0	0.5736	0.3425

[30] S. gil Lee, W. Ping, B. Ginsburg, B. Catanzaro, and S. Yoon, "Bigvgan: A universal neural vocoder with large-scale training," 2023.

[31] K. Mani, S. Daga, S. Garg, N. S. Shankar, K. M. Jatavallabhula, and K. M. Krishna, "Monolayout: Amodal scene layout from a single image," 2020.

[32] C. Reading, A. Harakeh, J. Chae, and S. L. Waslander, "Categorical depth distribution network for monocular 3d object detection," 2021.

[33] J. Zou, J. Xiao, Z. Zhu, J. Huang, G. Huang, D. Du, and X. Wang, "Hft: Lifting perspective representations via hybrid feature transformation," 2022.

[34] L. Chen, C. Sima, Y. Li, Z. Zheng, J. Xu, X. Geng, H. Li, C. He, J. Shi, Y. Qiao, and J. Yan, "Persformer: 3d lane detection via perspective transformer and the openlane benchmark," 2022.

[35] Y. Wang, V. Guizilini, T. Zhang, Y. Wang, H. Zhao, and J. Solomon, "Detr3d: 3d object detection from multi-view images via 3d-to-2d queries," 2021.

[36] Y. Liu, T. Wang, X. Zhang, and J. Sun, "Petr: Position embedding transformation for multi-view 3d object detection," 2022.

[37] Y. Liu, J. Yan, F. Jia, S. Li, A. Gao, T. Wang, X. Zhang, and J. Sun, "Petrv2: A unified framework for 3d perception from multi-camera images," 2022.

[38] L. Peng, Z. Chen, Z. Fu, P. Liang, and E. Cheng, "Bevsegformer: Bird's eye view semantic segmentation from arbitrary camera rigs," 2022.

[39] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[40] J. Wang, K. Sun, T. Cheng, B. Jiang, C. Deng, Y. Zhao, D. Liu, Y. Mu, M. Tan, X. Wang, W. Liu, and B. Xiao, "Deep high-resolution representation learning for visual recognition," 2020.

[41] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "Resnest: Split-attention networks," 2020.

[42] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," 2015.

[43] V. Iglovikov and A. Shvets, "Ternausnet: U-net with vgg11 encoder pre-trained on imagenet for image segmentation," 2018.

[44] F. I. Diakogiannis, F. Waldner, P. Caccetta, and C. Wu, "Resunet-a: A deep learning framework for semantic segmentation of remotely sensed data," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 162, p. 94–114, Apr. 2020. [Online]. Available: http://dx.doi.org/10.1016/j.isprsjprs.2020.01.013

[45] F. Jia, W. H. Wong, and T. Zeng, "Ddunet: Dense dense u-net with applications in image denoising," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, 2021, pp. 354–364.

[46] D. Jha, M. A. Riegler, D. Johansen, P. Halvorsen, and H. D. Johansen, "Doubleu-net: A deep convolutional neural network for medical image segmentation," 2020.

[47] R. Kestur, S. Farooq, R. Abdal, E. Mehraj, O. Narasipura, and M. Mudigere, "Ufcn: a fully convolutional neural network for road extraction in rgb imagery acquired by remote sensing from an unmanned aerial vehicle," *Journal of Applied Remote Sensing*, vol. 12, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:126186006

[48] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," 2016.

[49] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021. [Online]. Available: https://arxiv.org/abs/2111.06377

[50] D. Ibañez, R. Fernandez-Beltran, F. Pla, and N. Yokoya, "Masked auto-encoding spectral–spatial transformer for hyperspectral image classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–14, 2022.

[51] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2023.

[52] S. Zheng, J. Lu, H. Zhao, X. Zhu, Z. Luo, Y. Wang, Y. Fu, J. Feng, T. Xiang, P. H. S. Torr, and L. Zhang, "Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers," 2021.

[53] R. Strudel, R. Garcia, I. Laptev, and C. Schmid, "Segmenter: Transformer for semantic segmentation," 2021.

[54] E. Xie, W. Wang, Z. Yu, A. Anandkumar, J. M. Alvarez, and P. Luo, "Segformer: Simple and efficient design for semantic segmentation with transformers," 2021.

[55] X. He, Y. Zhou, J. Zhao, D. Zhang, R. Yao, and Y. Xue, "Swin transformer embedding unet for remote sensing image semantic segmentation," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022.

[56] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," 2021.

[57] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-net: Fully convolutional neural networks for volumetric medical image segmentation," 2016.

[58] Z. Liu, Y. Shang, T. Li, G. Chen, Y. Wang, Q. Hu, and P. Zhu, "Robust multi-drone multi-target tracking to resolve target occlusion: A benchmark," *IEEE Transactions on Multimedia*, vol. 25, pp. 1462–1476, 2023.

[59] P. Zhu, J. Zheng, D. Du, L. Wen, Y. Sun, and Q. Hu, "Multi-drone-based single object tracking with agent sharing network," *IEEE Transactions on Circuits and Systems for Video Technol-*

ogy, vol. 31, no. 10, pp. 4058–4070, 2021.
[60] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," 2020.
[61] 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, pp. 3354–3361.
[62] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine, V. Vasudevan, W. Han, J. Ngiam, H. Zhao, A. Timofeev, S. Ettinger, M. Krivokon, A. Gao, A. Joshi, S. Zhao, S. Cheng, Y. Zhang, J. Shlens, Z. Chen, and D. Anguelov, "Scalability in perception for autonomous driving: Waymo open dataset," 2020.
[63] "Google earth studio," https://www.google.com/earth/studio/.
[64] L. Reiher, B. Lampe, and L. Eckstein, "A sim2real deep learning approach for the transformation of images from multiple vehicle-mounted cameras to a semantically segmented image in bird's eye view," in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–7.

Baihong Lin received the B.S. degree from the Image Processing Center, School of Astronautics, Beihang University, Beijing, China, in 2024, where he is currently pursuing the Ph.D. degree. His research interests include computer vision, image processing, and deep learning.

Zhengxia Zou (Senior Member, IEEE) received his B.S. degree and his Ph.D. degree from the Image Processing Center, School of Astronautics, Beihang University in 2013 and 2018, respectively. He is currently a Professor at the School of Astronautics, Beihang University. During 2018-2021, he was a postdoc research fellow at the University of Michigan, Ann Arbor. His research interests include computer vision and related problems in remote sensing and autonomous driving. He has published more than 20 peer-reviewed papers in toptier journals and conferences, including TPAMI, TIP, TGRS, CVPR, ICCV, AAAI. His research has been featured in more than 30 global tech media outlets and adopted by multiple application platforms with over 50 million users worldwide. His personal website is https://zhengxiazou.github.io/.

Zhenwei Shi (Senior Member, IEEE) is currently a Professor and the Dean of the Image Processing Center, School of Astronautics, Beihang University, Beijing, China. He has authored or co-authored over 200 scientific articles in refereed journals and proceedings. His research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning. Prof. Shi serves as an Editor for IEEE Transactions on Geoscience and Remote Sensing, Pattern Recognition, ISPRS Journal of Photogrammetry and Remote Sensing, Infrared Physics and Technology, etc. His personal website is http://levir.buaa.edu.cn/.