A Self-Adaptive Feature Extraction Method for Aerial-View Geo-Localization

Jinliang Lin, Zhiming Luo, *Member, IEEE*, Dazhen Lin, Shaozi Li, and Zhun Zhong

***Abstract*— Cross-view geo-localization aims to match the same geographic location from different view images, *e.g.*, drone-view images and geo-referenced satellite-view images. Due to UAV cameras' different shooting angles and heights, the scale of the same captured target building in the drone-view images varies greatly. Meanwhile, there is a difference in size and floor area for different geographic locations in the real world, such as towers and stadiums, which also leads to scale variants of geographic targets in the images. However, existing methods mainly focus on extracting the fine-grained information of the geographic targets or the contextual information of the surrounding area, which overlook the robust feature for scale changes and the importance of feature alignment. In this study, we argue that the key underpinning of this task is to train a network to mine a discriminative representation against scale variants. To this end, we design an effective and novel end-to-end network called Self-Adaptive Feature Extraction Network (Safe-Net) to extract powerful scale-invariant features in a self-adaptive manner. Safe-Net includes a global representation-guided feature alignment module and a saliency-guided feature partition module. The former applies an affine transformation guided by the global feature for adaptive feature alignment. Without extra region annotations, the latter computes saliency distribution for different regions of the image and adopts the saliency information to guide a self-adaptive feature partition on the feature map to learn a visual representation against scale variants. Experiments on two prevailing large-scale aerial-view geo-localization benchmarks, *i.e.*, University-1652 and SUES-200, show that the proposed method achieves state-of-the-art results. In addition, our proposed Safe-Net has a significant scale adaptive capability and can extract robust feature representations for those query images with small target buildings. The source code of this study is available at: https://github.com/AggMan96/Safe-Net.**

***Index Terms*— Geo-localization, cross-view, scale-invariant, self-adaptive.**

I. Introduction

THE goal of cross-view geo-localization is to match the same geographic location from images captured by different platforms. This task can also be regarded as an image-based retrieval task, and several studies have been conducted in this area [1], [2], [3], [4], [5]. For instance, given a query image from one perspective (*e.g.*, drone view), the system retrieves the most relevant images from another view (*e.g.*, satellite view). Since satellite-view images are typically annotated with geo-referenced information, such as GPS coordinates [6], drones can determine the location of target buildings by matching corresponding satellite-view images. Aerial-view geo-localization has numerous real-world applications, including drone delivery, drone navigation, autonomous vehicles, event detection, etc [7], [8], [9], [10], [11], [12], [13], [14], [15], [16], [17]. Drone-based aerial-view geo-localization primarily involves two tasks [18]: Drone-view Target Localization and Drone Navigation. The former localizes target buildings in drone-view images by matching them with corresponding satellite-view images, while the latter retrieves relevant places in drone-view images based on a given satellite-view image and navigates UAVs according to their flight history.

Along with the success of Convolutional Neural Networks (CNNs) in various computer vision tasks, aerial-view geo-localization recently achieved great progress based on CNNs [18], [19], [20], [21], [22], [23]. Zheng et al. [18] introduced an identity loss for drone-satellite cross-view geo-localization. Ding et al. [19] proposed a location classification matching model (LCM) to address the problem of sample imbalance. Lin et al. [20] and Zhuang et al. [24] adopted attention mechanisms to mine fine-grained information. Wang et al. [21] and Tian et al. [22] extracted contextual information from images in an end-to-end manner. Additionally, Dai et al. [23] and Zhuang et al. [25] employed transformer architectures as the backbone for aerial-view geo-localization networks. However, most methods only focus on fine-grained or contextual information while significantly ignoring the scale-invariant features.

In real world scenarios, drones capture images of buildings from different heights and angles, resulting in inconsistent scales of the target buildings in the drone-view images. In other words, for the same location, some drone-view images depict the target building at a small scale, while others show it occupying a large portion of the image (refer to Fig. 1). Additionally, the geographic targets vary in size and occupy different areas in both drone-view and satellite-view images. Despite advancements made by existing methods in aerial-view geo-localization, they fail to consider a robust visual representation that can handle scale variations.

Received 25 July 2023; revised 21 February 2024 and 14 October 2024; accepted 2 December 2024. Date of publication 12 December 2024; date of current version 19 December 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62076210, Grant 62276221, and Grant 62376232; and in part by Fujian Provincial Natural Science Foundation of China under Grant 2022J01002. The associate editor coordinating the review of this article and approving it for publication was Prof. Zhu Li. *(Corresponding authors: Zhiming Luo; Zhun Zhong.)*

Jinliang Lin, Zhiming Luo, Dazhen Lin, and Shaozi Li are with the Department of Artificial Intelligence, Xiamen University, Xiamen 361005, China (e-mail: jinlianglin@stu.xmu.edu.cn; zhiming.luo@xmu.edu.cn; dzlin@xmu.edu.cn; szlig@xmu.edu.cn).

Zhun Zhong is with the School of Computer Science and Information Engineering, Hefei University of Technology, Hefei 230601, China (e-mail: zhunzhong007@gmail.com).

Digital Object Identifier 10.1109/TIP.2024.3513157

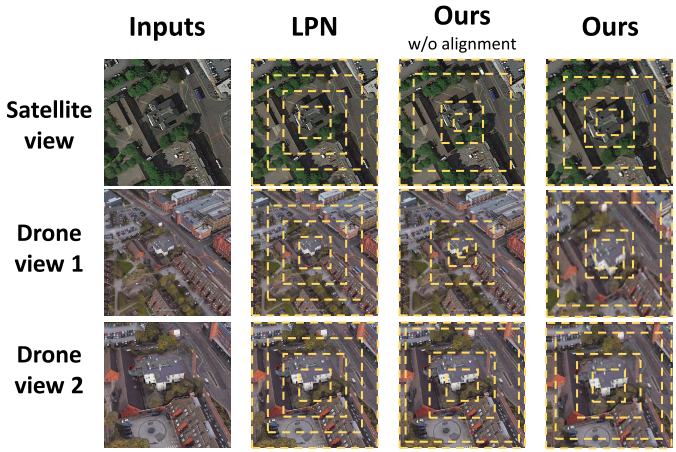


Fig. 1. The images in the first column are the input images from satellite-view and drone-view of different angles and heights. The images in the second column are the predefined rigid partitions of LPN [21]. The images in the third column are the self-adaptive partition results of our method without using the proposed feature alignment module, while the images in the fourth column are the partition results of our proposed method after feature alignment.

Inspired by LPN [21], which divides the image into uniform regions (as shown in the second column of Fig. 1), we propose a novel framework called Self-Adaptive Feature Extraction Network (Safe-Net) for aerial-view geo-localization. Safe-Net consists of two modules for extracting scale-invariant representations: global representation-guided feature alignment module and saliency-guided self-adaptive feature partition module. Specifically, the global representation-guided feature alignment module applies an affine transformation based on the global representation to align the deep local feature maps. The saliency-guided self-adaptive feature partition module utilizes fusion features to determine the saliency distribution of regions and then uses this saliency information to guide a self-adaptive square-ring feature partition. These two modules work together to promote and complement each other. The first module performs a global representation-guided affine transformation on the high-level semantic feature map, aligning the feature distribution of images to some extent. This simplifies the feature partition process and enables the model to better recognize and locate targets. In turn, an effective feature partition enhances the model's ability to perceive and distinguish geographic targets, thereby facilitating better alignment of image features. As shown in Fig. 1, our proposed feature partition module has the capability to partition regions in an adaptive manner by considering the distribution of images (see the third column in Fig. 1), and the feature alignment module can reduce scale variance of target buildings in different images to ease the difficult of feature partition(see the fourth column in Fig. 1). Without the prerequisite of extra annotations, our proposed Safe-Net can jointly extract the contextual information and robust scale-variant features to achieve feature alignment.

We conduct extensive experiments on two large-scale aerial-view geo-localization datasets, *i.e.*, University-1652 [18] and SUES-200 [26], to evaluate the effectiveness of our proposed method. Our model outperforms existing methods and achieves state-of-the-art results. Additionally, our Safe-Net demonstrates a strong ability to adapt to varying scales and effectively extract robust feature representations, even for query images containing small target buildings.

To summarize, our main contributions are as follows:

- We propose a novel network called Self-Adaptive Feature Extraction Network (Safe-Net) to address the issue of inconsistent scales in target buildings among Aerial-view Geo-localization tasks. Safe-Net can effectively learn a scale-invariant visual representation.
- Our model includes two modules: global representation-guided feature alignment module and saliency-guided self-adaptive feature partition module. These modules enhance the model's capability against changes in scale. The feature alignment and partition are adaptively conducted in an end-to-end manner without requiring any extra annotations.
- Compared to existing methods, our proposed model achieves state-of-the-art results for both drone-based retrieval tasks on two aerial-view geo-localization datasets, *i.e.*, University-1652 and SUES-200.

II. Related Works

A. Ground-Satellite Cross-View Geo-Localization

Ground-satellite cross-view geo-localization refers to estimating the location of a given ground-view query image by matching it with a large database of geo-referenced satellite-view images. Existing deep learning-based methods treat this task as an image retrieval problem, and can be categorized into the following three groups.

1) Content-Based Representation Learning: Benefiting from the success of the convolutional neural networks (CNNs), Workman and Jacobs [28] first introduced deep representations to deal with the cross-view matching task, which adopted a CNN pre-trained on ImageNet [29] and Places [30] to extract deep features for different view images. In addition, some works focused on metric learning and designed suitable loss functions to obtain a content-based robust representation. Vo and Hays [31] proposed the Soft Margin Triplet Loss to train a CNN. Hu et al. [32] designed the Weighted Soft Margin Triplet Loss and proposed CVM-Nets. Additionally, the attention mechanism was also used in geo-localization for representation learning. Cai et al. [33] proposed a context-based attention module (FCAM) including channel and spatial attention sub-modules. Rodrigues and Tani [34] designed a multi-scale attention module to address the problem of temporal variation in scenes.

2) Spatial Correspondence Establishing: In the view of explicitly bridging the domain gap, Regmin and Shah [35] adopted a conditional generative adversarial network [36] and Canny Edge Detection [37] to synthesize an aerial-view image for a given ground-view image. Shi et al. [38] used a Polar Transform algorithm to warp satellite-view images and devised a feature transport module (CVFT) [39] to transfer features from one domain to the other. Moreover, to apply orientation cues to establish a spatial correspondence, Liu and Li [4] incorporated orientation information into a Siamese network, enabling it to jointly learn orientation geometry information and feature embedding. To simultaneously estimate the orientation and position of a query image regardless of its field of

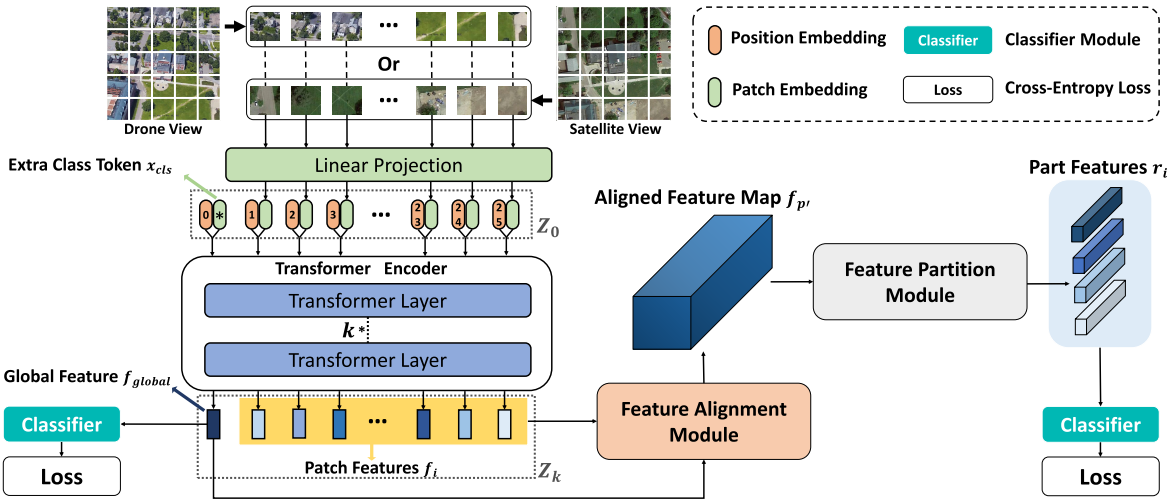


Fig. 2. Overview of the proposed Safe-Net framework. The Vision Transformer (ViT) [27] is used as the backbone to extract features from image patches. The feature alignment module utilizes a global feature to guide a transformation on the patch features, allowing for adaptive feature alignment. The feature partition module computes a saliency distribution of the input feature map and incorporates this information to divide the patch feature map into four part-level features. The classifier module contains linear layers, batch normalization and LeakyReLU, and the cross-entropy loss is adopted for optimization. It is noted that both drone-view images and satellite-view images are inputted to the same ViT backbone and two proposed modules with shared weights.

view, Shi et al. [40] designed a Dynamic Similarity Matching (DSM) module to measure feature similarity from image pairs.

3) Transformer-Based Architecture: In addition, recent works [41], [42] investigated using the transformer architectures [43] for the task of ground-satellite cross-view geo-localization. Yang et al. [41] proposed a layer-to-layer transformer (L2LTR) to model global dependencies. Furthermore, Zhu et al. [42] introduced a pure transformer-based method (TransGeo) to reduce computation costs.

B. Drone-Satellite Cross-View Geo-Localization

With the rapid development of unmanned aerial vehicles (UAVs), drones also serve as a new platform for cross-view geo-localization. To verify the effectiveness of the drone platform, Zheng et al. [18] introduced the first drone-based cross-view geo-localization dataset named University-1652, which enables two missions, *i.e.*, drone view target localization and drone navigation. Moreover, they adopted a three-branch Siamese network and instance loss [44] to perform drone-satellite cross-view geo-localization. Most recently, Zhu et al. [26] released another multi-height, multi-scene drone-satellite geo-localization dataset called SUES-200 and established an efficient pipeline to evaluate various matching models.

To solve the imbalance between drone-view images and satellite-view images, Ding et al. [19] proposed a method based on location classification (LCM) to expand the samples of satellite images. Through a feature-level partition strategy, Wang et al. [21] introduced the Local Pattern Network (LPN) to explore contextual information and perform part-wise representation learning. Inspired by the human visual system for mining local patterns, Lin et al. [20] proposed an RK-Net, which leveraged a unit subtraction attention module (USAM) to enhance the feature discrimination of images. Zhuang et al. [24] also proposed an attention module called multi-scale block attention (MSBA) to extract a more subtle relationship between different views and reduce the inference time. Based on the multi-scale block, Bui et al. [45] employed a channel-based attention mechanism and a part-based representation learning method to capture the discriminative parts of objects in images, which significantly improved matching accuracy. To reduce the domain gap between drone-view and satellite-view, Tian et al. [22] designed an architecture called PCL, which used a perspective projection transformation and a conditional generative adversarial network [36] to synthesize UAV images close to real satellite images. Instead of using a CNN-based backbone, Dai et al. [23] introduced a transformer-based structure called Feature Segmentation and Region Alignment (FSRA) to address the problem caused by the position offset and uncertainty of distance in drone-satellite cross-view geo-localization. Zhuang et al. [25] also adopted a transformer-based network to extract contextual information and designed a semantic guidance module (SGM) to align the same semantic parts between different images.

Different from existing works, the proposed method mainly considers the scale variations of geographic targets in different views. We adopt a global representation-guided feature alignment and a saliency-guided self-adaptive feature partition, which enable the network to extract a scale-invariant visual representations in an end-to-end manner.

III. Methodology

In this section, we provide a detailed description of our proposed method. We first describe the backbone model for feature extraction. Then we present the proposed global representation-guided feature alignment module and saliency-guided self-adaptive feature partition module for extracting a scale-invariant representation. Finally, we illustrate the optimization objective and the inference process. The overall framework of the proposed method is shown in Fig. 2.

A. Feature Extraction Model

Following [23], we adopt the Vision Transformer (ViT) [27] as our feature extraction backbone to extract features for

drone-view images and satellite-view images. Given an input image $I \in \mathbb{R}^{H\times W\times C}$, where H,W,C represent its height, width, and channels, the ViT backbone first divides it into N patches $p_i \in \mathbb{R}^{M\times M\times C}(i = 1, 2, \ldots, N)$. Here N equals $\frac{H}{M} \times \frac{W}{M}$. Then each patch p_i is fed into a trainable linear projection layer to generate the corresponding patch embedding $x_i \in \mathbb{R}^D$. In addition, an extra class token $x_{cls} \in \mathbb{R}^D$ is added to the sequence of patch embedding tokens. In ViT, the class token x_{cls} aggregates global information from other patch tokens, and the output of the class token after the transformer layers can be considered the global feature representation of the input image. To preserve the positional information, a position embedding $Pos \in \mathbb{R}^{(N+1)\times D}$ is added to each token, including the class token. The input sequence Z_0 to the transformer encoder can be formulated as follows:

$$\begin{aligned} Z_0 &= [x_{cls},\ x_1,\ \ldots,\ x_N] + Pos \\ &= [x_{cls},\ L(p_1),\ \ldots,\ L(p_N)] + Pos \end{aligned} \tag{1}$$

where L is the trainable linear projection layer.

Next, after the k-layer transformer encoder, we can get the final feature representation sequence as:

$$\begin{aligned} Z_k &= T([x_{cls},\ x_1,\ \ldots,\ x_N] + Pos) \\ &= [f_{global},\ f_1,\ \ldots,\ f_N] \end{aligned} \tag{2}$$

where T represents the transformer encoder, $f_i \in \mathbb{R}^D$ $(i = 1, 2, \ldots, N)$ are the output features for each patch token, and $f_{global} \in \mathbb{R}^D$ is the global feature of the class token.

B. Global Representation-Guided Feature Alignment

To align the features of images from different views, we further design a global representation-guided feature alignment module to enhance the feature representation towards geographic targets. Inspired by Spatial Transformer Networks (STN) [46], we apply an affine transformation on the features extracted by the transformer encoder to conduct feature alignment. As shown in Fig. 3, the feature alignment module contains a localization sub-network, a sampling grid generator, and a bilinear sampler.

Given the f_{global} and f_i $(i = 1, 2, \ldots, N)$ computed from the transformer encoder, we first reshape the sequence patch features f_i $(i = 1, 2, \ldots, N)$ into a feature map $f_p \in \mathbb{R}^{\frac{H}{M}\times\frac{W}{M}\times D}$, where $N = \frac{H}{M} \times \frac{W}{M}$. Then, we utilize the localization sub-network to estimate the parameter A_θ of the affine transformation by using the f_{global} as input, since it contains the global information of the geographic target. The computation of the localization sub-network is denoted as:

$$A_\theta = \begin{bmatrix} \theta_{s1} & \theta_{r1} & \theta_{t1} \\ \theta_{r2} & \theta_{s2} & \theta_{t2} \end{bmatrix} = \mathrm{Loc}(f_{global}), \tag{3}$$

where the localization sub-network Loc is implemented by two linear layers. $(\theta_{s1}, \theta_{s2})$, $(\theta_{r1}, \theta_{r2})$ and $(\theta_{t1}, \theta_{t2})$ are the scale, rotation and translation parameters, respectively.

Next, the grid generator computes a sampling grid for each feature coordinate based on the following formula:

$$\begin{pmatrix} u^{in} \\ v^{in} \end{pmatrix} = A_\theta \begin{pmatrix} u^{out} \\ v^{out} \\ 1 \end{pmatrix} = \begin{bmatrix} \theta_{s1} & \theta_{r1} & \theta_{t1} \\ \theta_{r2} & \theta_{s2} & \theta_{t2} \end{bmatrix} \begin{pmatrix} u^{out} \\ v^{out} \\ 1 \end{pmatrix} \tag{4}$$

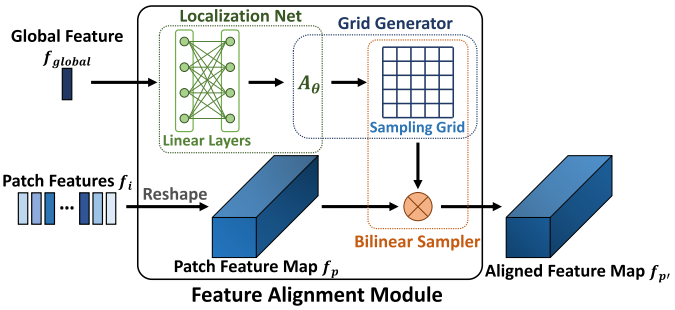


Fig. 3. The structure of the proposed global representation-guided feature alignment module.

where (u^{in}, v^{in}) are the original coordinates of the input feature map and (u^{out}, v^{out}) are the aligned coordinates of the output feature map.

Lastly, the bilinear sampler uses the sampling grid to resample the patch feature map f_p by bilinear interpolation, producing a new aligned feature map $f_{p'}$, which is used in the subsequent feature partition module. In our method, for an input image of size $256 \times 256 \times 3$, an aligned feature map $f_{p'}$ of size $16 \times 16 \times D$ can be obtained.

C. Saliency-Guided Self-Adaptive Feature Partition

Since the UAV captures images of the geographic target from different shooting angles and heights in real applications, the scale of the target buildings in the drone-view images changes significantly with the distance of the UAV to the target location. Additionally, the target buildings have different sizes and occupy different amounts of areas in both drone-view and satellite-view images.

To ease the above problem of inconsistent scale and different sizes in drone-view and satellite-view, and achieve region-level feature alignment, inspired by [21], we design a saliency-guided self-adaptive feature partition module to divide the high-level semantic feature map against scale variants. Fig. 4 illustrates the process of the saliency-guided self-adaptive feature partition module. In the following, we describe the detailed implementation of the proposed feature partition module.

1) Saliency Value: Given the aligned feature map $f_{p'} \in \mathbb{R}^{16\times16\times D}$ after the feature alignment, we adopt a max-pooling operation along the channel axis to aggregate the feature map $f_{p'}$. This simple operation can effectively highlight salient regions in the feature map [47]. After the max-pooling operation, we can obtain an aggregated feature map f_g with a size of 16×16. Then we slice the aggregated feature map f_g into 8 square-ring feature maps with a ring width of 1, denoted as f_r^i $(i = 1, 2, \ldots, 8)$. The superscript i indicates the i^{th} square-ring feature map from the center. Next, we apply a global average pooling (GAP) on each square-ring feature map f_r^i to obtain the saliency value S_i, which reflects the saliency of the corresponding square-ring feature map. In geo-localization datasets, target buildings are commonly situated at or near the center of the image. The closer the area is to the center, the more content of target buildings is covered, which means the area contains more salient information. Hence, the feature map close to the center contains a more salient feature,

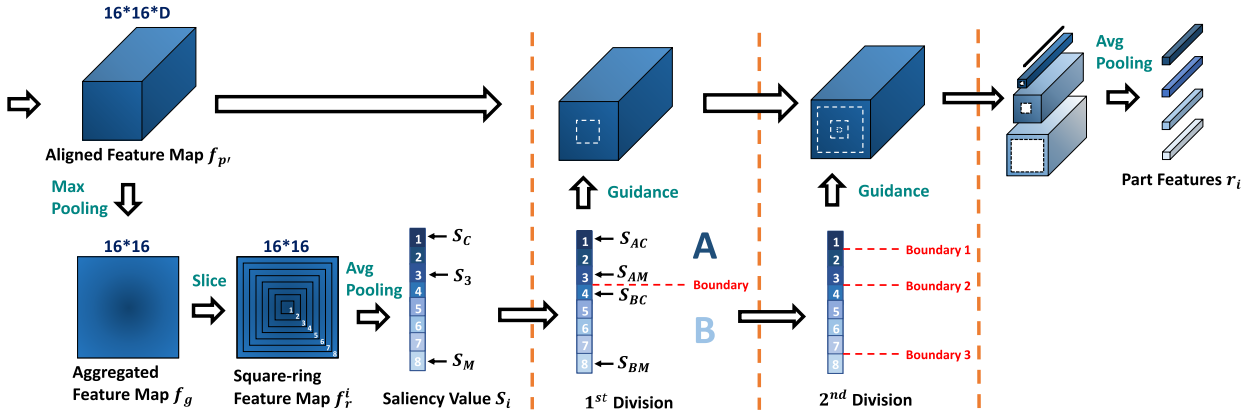


Fig. 4. The computation pipeline of the proposed saliency-guided self-adaptive feature partition module.

which can obtain a larger saliency value through the above operation. On the contrary, the feature map far from the center contains fewer salient features, which obtain a lower saliency. As a result, the saliency value S_i generally decreases with the increase of i.

2) Feature Partition: After obtaining the saliency value S, we conduct a two-stage saliency-guided feature division on the global feature. The procedure of the feature division is as follows:

3) 1^{st} Division Stage: We firstly denote the saliency value S_1 as S_C, representing the saliency value S_1 is obtained from the *Center* of the global feature map. Similarly, we denote the saliency value S_8 as S_M, which is obtained from the *Margin* of the global feature map. Next, we intend to obtain the boundary between the salient and sub-salient regions by comparing the distance from S_i to S_C and S_M. The boundary can be calculated by the following equation:

$$n = \arg\min_i \big||S_c - S_i| - |S_M - S_i|\big|, \tag{5}$$

where $|S_C - S_i|$ means the distance between S_C and S_i and $|S_M - S_i|$ means the distance between S_M and S_i. n is recorded as the boundary position of the 1^{st} division stage. According to the boundary, we divide the global feature map into region A and region B. We regard region A as a salient region (foreground) and region B as a sub-salient region (background). Region A typically covers the feature of the target building, and region B contains the feature of its surrounding context.

4) 2^{nd} Division Stage: In the second division stage, we perform the feature division within region A and region B. In region A, we denote S_1 as S_{AC} and S_{n-1} as S_{AM}, which mean the center and the margin of region A respectively. Then, similar to the operations on the 1^{st} division stage, we calculate $\arg\min_i \big||S_{AC} - S_i| - |S_{AM} - S_i|\big|$ to determine the boundary of region A. In the same way, we denote S_n as S_{BC} and S_8 as S_{BM}, which mean the center and the margin of region B respectively. By calculating $\arg\min_i \big||S_{BC} - S_i| - |S_{BM} - S_i|\big|$, we can determine the boundary of region B. Finally, we divide the global feature map into four regions based on the above three boundaries and adopt average pooling to transform each feature region into part feature $\{r_1, r_2, r_3, r_4\}$.

D. Training and Inference

1) Classifier Module: Following previous works, we treat the training process of the model as a supervised classification learning task and leverage a classifier module to map features from different views into a shared feature space. Concretely, the classifier module consists of a 512-dimensional fully connected layer, a batch normalization layer, a Leaky ReLU activation, and a fully connected layer for classification. This classifier takes the global feature and each part feature separately as input and predicts their Geo-Tags labels.

2) Loss Function: During training, each location is regarded as an individual class and images with the same Geo-Tag belong to the same class. The cross-entropy loss is used as the classification loss function to optimize the entire model. Given a global feature f_{global} and part features $\{r_1, r_2, r_3, r_4\}$, which correspond to the ground-truth Geo-Tag y, the cross-entropy loss is formulated as follows:

$$\begin{aligned} Loss = &-\mathbb{1}_y \log\Big(\text{Softmax}\big(\text{Cls}(f_{global})\big)\Big) \\ &+ \sum_{j=1}^{4} -\mathbb{1}_y \log\Big(\text{Softmax}\big(\text{Cls}(r_j)\big)\Big), \end{aligned} \tag{6}$$

where Cls is the classifier module and $\mathbb{1}_y$ is the one-hot vector related to the ground-truth label y. The loss function forces the distances of features with the same Geo-Tag to be closer and the distances of features with different Geo-Tags to be separated.

3) Inference: During the inference phase, we concatenate all features before the last classification layer in the classifier module to form the final descriptor of the input image. The cosine distance is calculated to measure the similarity between query images and gallery images.

IV. Experiments

A. Dataset and Evaluation Protocol

1) Dataset: In this study, we evaluate our method on two large-scale cross aerial-view geo-localization datasets, *i.e.*, **University-1652** [18] and **SUES-200** [26].

University-1652 is a multi-view multi-source dataset, which includes drone-view images, satellite-view images, and street-view images. Instead of selecting landmarks as the target locations, University-1652 selects 1,652 ordinary buildings of

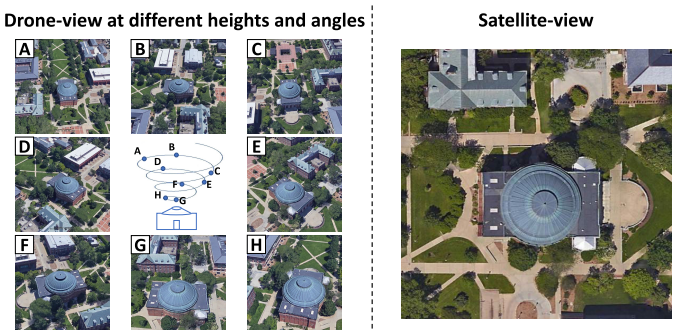


Fig. 5. Aerial examples for the University-1652 dataset.

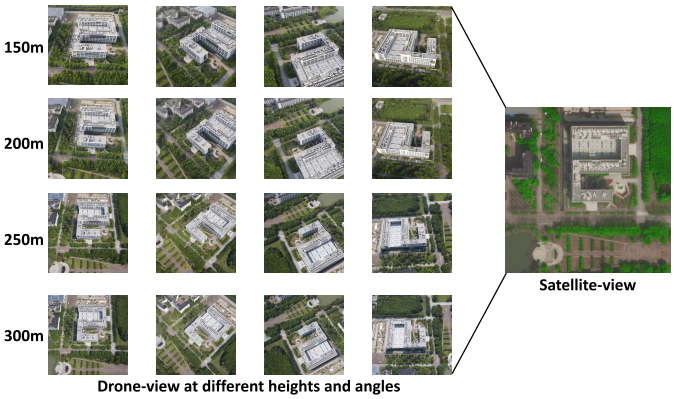


Fig. 6. Aerial examples for the SUES-200 dataset.

72 universities worldwide as targets. Concretely, there are 701 buildings of 33 universities in the training set and the rest 951 buildings of the other universities in the testing set. Each building is associated with 54 drone-view images and one satellite-view image. As shown in Fig. 5, these drone-view images are obtained from different shooting angles and heights, which results in the inconsistent scales of the target buildings in the images. The dataset enables two main tasks, *i.e.*, drone view target localization (Drone → Satellite) and drone navigation (Satellite → Drone). In the task of drone-view target localization, there is only one true-matched satellite-view image in the gallery for a drone-view query image. In the task of drone navigation, there are multiple ground truth drone-view images in the gallery for a satellite-view query image.

SUES-200 is a newly released multi-height multi-scene aerial-view geo-localization dataset, which collects drone-view images and corresponding satellite-view images at 200 locations around the Shanghai University of Engineering and Science (SUES). There are 120 locations for training and the rest 80 locations for testing. Moreover, during the testing phase, the training images are added to the gallery dataset as confusion data. Unlike University-1652, SUES-200 contains a wider range of scene types and acquires drone-view images in real environments. As shown in Fig. 6, for each location, SUES-200 collects 50 drone-view images from different angles at four different heights (150m, 200m, 250m, and 300m). Therefore, each scene includes one satellite-view image and 200 drone-view images. In addition, there is partial overlap between drone-view images of different scenes, increasing the difficulty of matching. SUES-200 also supports two cross-view matching tasks, *i.e.*, drone view target localization and drone navigation.

2) Evaluation Protocol: The recall accuracy at top 1 (Recall@1) and the average precision (AP) are adopted as the evaluation metrics to evaluate the performance of the proposed method. The value of Recall@1 is equal to 1 if the true matching image appears in the top 1 of the result ranking list. Recall@1 is sensitive to the position of the positive images that appear in the result ranking list. The average precision is the area under the Precision-Recall curve. We also calculate the mean AP (mAP) over all queries.

B. Implementation Details

In our method, a small-size Vision Transformer (Vit-S) [27] with pre-trained weights on ImageNet [29] is applied as our feature extraction model (backbone) and the number of transformer layers in the transformer encoder is set to 8. In the training and testing phases, each input image is resized to a size of 256×256. During training, data augmentation is applied to augment input images, including random flipping and random cropping. The network is trained for 120 epochs with a mini-batch of 8. We employ the stochastic gradient descent optimizer (SGD) with momentum $= 0.9$ and weight decay $= 0.0005$ to optimize our model. For the learning rate, we initially set 0.003, 0.0001, and 0.01 for the parameters of the backbone, the localization network of the feature alignment module, and the rest of the learnable parameters, respectively. The learning rate decays by a factor of 0.1 after 70 and 110 epochs for all layers. Following previous work [23], we adopt a multiple sampling strategy with a sampling number of 2 in the training phase, which applies data augmentation to generate two augmented satellite images for each original satellite image in the training set. During testing, we utilize the cosine distance to measure the similarity between the query image and gallery images. We conduct all experiments on an Nvidia RTX 2080Ti GPU and adopt the PyTorch toolbox to implement our method.

Besides, we also train a model with the ResNet-50 [48] as the backbone network for comparison. In the implementation, we retain all the convolutional layers to compute the local features and use the final global pooling layer to get the global feature. Then, we train the model in a same manner as using Vit-S as backbone.

C. Comparison With State-of-the-Art Methods

1) Results on University-1652: We compare our method with existing competitive approaches on University-1652, including Zheng et al. [18], RK-Net [20], LCM [19], LPN [21], PCL [22], SGM [25], MSBA [24], FSRA [23], and PAAN [45]. For a fair comparison, we report the results of SOTA methods with the same input image size of 256×256.

As shown in Tab. I, our proposed method with Vit-S achieves 86.98% in Recall@1 accuracy and 88.85% in mAP on the task of "Drone → Satellite", while the proposed model with ResNet-50 achieves 83.85% in Recall@1 accuracy and 86.12% in mAP. Compared to ResNet, ViT has a stronger feature extraction capability, and the transformer architecture is capable of extracting both global and local features simulta-

TABLE I

COMPARISON WITH STATE OF THE ARTS ON UNIVERSITY-1652. M DENOTES THE MARGIN OF THE TRIPLET LOSS."DRONE → SATELLITE" DENOTES THE DRONE-VIEW TARGET LOCALIZATION TASK, AND "SATELLITE → DRONE" INDICATES THE DRONE NAVIGATION TASK

Method	Backbone	Drone → Satellite		Satellite → Drone	
		R@1	mAP	R@1	mAP
Contrastive Loss [1]	ResNet-50	52.39	57.44	63.91	52.24
Soft Margin Triplet [32]	ResNet-50	53.21	58.03	65.62	54.47
Triplet (M = 0.5) [49]	ResNet-50	53.58	58.60	64.48	53.15
Triplet (M = 0.3) [49]	ResNet-50	55.18	59.97	63.62	53.85
Zheng *et al.* [18]	ResNet-50	58.49	63.13	71.18	58.74
RK-Net [20]	ResNet-50	65.63	69.68	78.32	64.87
LCM [19]	ResNet-50	66.65	70.82	79.89	65.38
LPN [21]	ResNet-50	74.18	77.39	85.16	73.68
PCL [22]	ResNet-50	79.47	83.63	87.69	78.51
SGM [25]	Swin-Tiny	82.14	84.72	88.16	81.81
MSBA [24]	ResNet-50	82.33	84.78	90.58	81.61
FSRA [23]	Vit-S	84.51	86.71	88.45	83.37
PAAN [45]	SEResNet-50	84.51	86.78	91.01	82.28
Ours	ResNet-50	83.85	86.12	90.01	82.31
Ours	Vit-S	**86.98**	**88.85**	**91.22**	**86.06**

neously, which makes it well-suited for our method. As a result of these factors, our method benefits from using ViT as the backbone network and achieves better performance compared to using ResNet as the backbone network.

According to the results, our method with Vit-S has surpassed existing competitive approaches and achieves the best performance for both tasks. The proposed method with ResNet-50 also achieves a competitive result. In particular, our proposed method with Vit-S outperforms the state-of-the-art method, *i.e.*, PAAN [45] with about 2.4% Recall@1 and 2.1% mAP on the task of "Drone → Satellite". Note that our approach uses the same backbone network as FSRA [23], *i.e.*, small-scale vision transformer network (Vit-S) [27], and our performance is better than FSRA in both Recall@1 and mAP for two tasks. Compared to FSRA and PAAN, our model introduces a feature alignment module, which enables the model to adaptively adjust the distribution of features extracted by the backbone network. As a result, our model exhibits robustness and better performance in diverse scenarios and leading to improved Recall@1 and mAP compared to FSRA and PAAN.

In addition, although our network and LPN adopt a feature-level partition strategy, our proposed approach with ResNet-50 still outperforms LPN by a large margin of about 9% Recall@1 on the task of "Drone → Satellite". In contrast to LPN, our proposed approach utilizes a saliency-guided self-adaptive feature partition mechanism, which allows the model to extract features more effectively from images with varying sizes. By adaptively partitioning the features based on their saliency values, our model achieves better performances in terms of Recall@1 and mAP, which highlights the significance of the self-adaptive feature partition strategy in handling diverse target scales.

In summary, our proposed method introduces a global representation-guided feature alignment strategy, and adopts a saliency-guided self-adaptive feature partition mechanism. These modules contribute to the better performance of our method compared to existing competitive approaches, demonstrating the effectiveness and robustness of our approach.

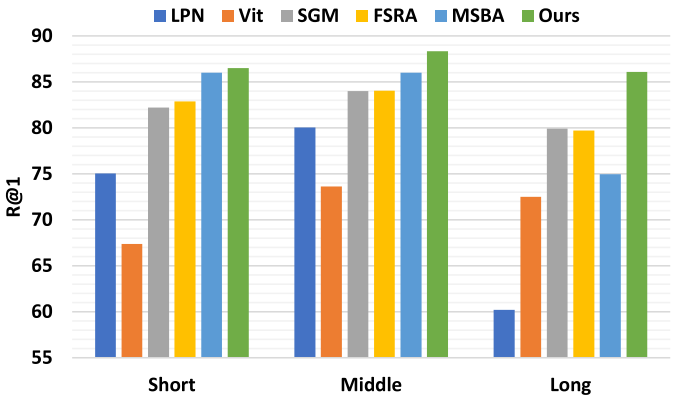


Fig. 7. The performance of different methods at different levels of distance. "Vit" represents the vision transformer [27].

2) Results on SUES-200: We also compare our approach with other competitive methods on the SUES-200 dataset, and the results are shown in Tab. II. On the task of "Drone → Satellite", our method can achieve the Recall@1 accuracy of 81.05%, 91.10%, 94.52% and 94.57% at the setting of four different heights (150m, 200m, 250m and 300m), respectively. On the task of "Satellite → Drone", the Recall@1 accuracy of ours are 97.50%, 96.25%, 97.50% and 98.75% in four heights. As the results show, our proposed approach outperforms other existing methods by a large margin and achieves the best performance under different settings. Moreover, the results of our method at different heights vary less than other methods, especially for the task of "Satellite → Drone". We argue the reason is that the target buildings in the SUES-200 dataset exhibit more significant differences in size, which poses a greater scale variation challenge. Meanwhile, compared to other methods, our method can effectively tackle this challenge by mining robust representation against scale variants, leading to improved performance of the model on the SUES-200 dataset. The above results support the stability and effectiveness of our proposed method under different settings. Compared to the University-1652 dataset, our model has a more significant advantage on the SUES-200 dataset.

D. Ablation Studies

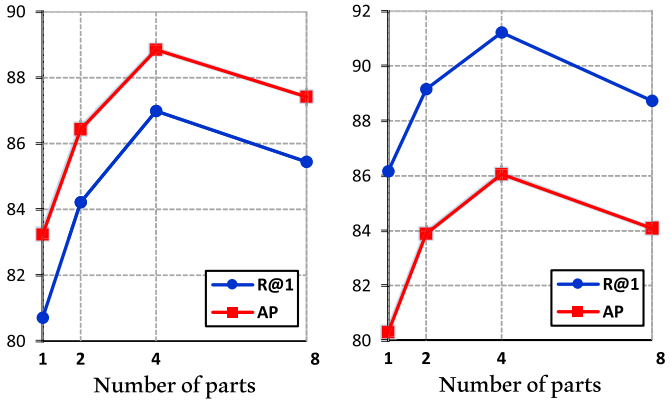
1) Robustness to Scale Changes: In aerial-view geo-localization, satellite-view images typically have a consistent scale, while drone-view images exhibit dynamically-varying scales dependent on the drone's distance to the geographic target. To verify the robustness of Safe-Net against such scale changes, we evaluate our proposed approach against other competitive methods through the task of drone-view target localization. Specifically, we organize the drone-view images in the University-1652 dataset into three distance levels based on the distance between the drone and the geographic target, namely Long, Middle, and Short scales.

In Fig. 7, we report the experimental results of different models at different levels of distance. We can observe that **1)** All approaches achieve the best performance at the "Middle" distance level, which verifies the images of "Middle"

TABLE II

COMPARISON WITH COMPETITIVE METHODS ON SUES-200. *(THE RESULTS OF FSRA [23] WERE PRODUCED BY THE PROVIDED SOURCE CODE. THE OTHERS ARE FROM [26]. ALL METHODS ARE EVALUATED WITH THE SAME SETTINGS)

Drone → Satellite								
Methods	150m		200m		250m		300m	
	Recall@1	AP	Recall@1	AP	Recall@1	AP	Recall@1	AP
LCM [19]	43.42	49.65	49.42	55.91	54.47	60.31	60.43	65.78
SUES-200 baseline [26]	59.32	64.93	62.30	67.24	71.35	75.49	77.17	80.67
LPN [21]	61.58	67.23	70.85	75.96	80.38	83.80	81.47	84.53
FSRA [23]	76.45	80.80	86.08	88.75	89.70	91.75	94.15	95.34
Ours	**81.05**	**84.76**	**91.10**	**93.04**	**94.52**	**95.74**	**94.57**	**95.60**
Satellite → Drone								
Methods	150m		200m		250m		300m	
	Recall@1	AP	Recall@1	AP	Recall@1	AP	Recall@1	AP
LCM [19]	57.50	38.11	68.75	49.19	72.50	47.94	75.00	59.36
SUES-200 baseline [26]	82.50	58.95	85.00	62.56	88.75	69.96	96.25	84.16
LPN [21]	83.75	66.78	88.75	75.01	92.50	81.34	92.50	85.72
FSRA [23]	90.00	77.61	92.50	86.10	93.75	91.08	97.75	94.61
Ours	**97.50**	**86.36**	**96.25**	**92.61**	**97.50**	**94.98**	**98.75**	**95.67**



(a) Drone → Satellite (b) Satellite → Drone

Fig. 8. The effect of different numbers of parts on the drone-view target localization task (a) and the drone navigation task (b). When the number of parts is 1, the model only utilizes a global average pooling on the patch feature map without using the process of feature partition.

distance level are the easiest to recognize and distinguish for neural networks; **2)** Our proposed method outperforms existing methods at all distance levels, especially at "Long" distance level; **3)** Compared with the "Short" distance level, most methods experience a performance drop at the "Long" distance level, while there are no significant changes for our proposed method. Our model achieves the smallest performance change magnitude, which verifies that our proposed method effectively mitigates the impact of scale changes on the model. The above observations demonstrate the robustness and stability of the proposed method against scale changes.

2) Effect of Different Numbers of Parts: To examine how different numbers of parts affect the proposed feature partition module, we conducted experiments by dividing the transformed patch feature map into different numbers of parts. The results are shown in Fig. 8, and we can observe that: **1)** By leveraging the proposed feature partition module, the performance of the model can be improved by a large margin on both tasks compared with only using the feature from global average pooling (1 part); **2)** As the number of parts increases from 1 to 4, the performance of the model increases accordingly; **3)** When the number of parts is 4, the model achieves the best performance on both Recall@1 and mAP; **4)** When further using 8 parts, the performance is slightly decreased.

TABLE III

ABLATION STUDY ON DIFFERENT COMPONENTS IN THE PROPOSED SAFE-NET. "FAM" DENOTES THE PROPOSED FEATURE ALIGNMENT MODULE, WHILE "FPM" REPRESENTS THE PROPOSED FEATURE PARTITION MODULE. **# PARAMS** DENOTES THE NUMBER OF PARAMETERS

Components		Drone→Satellite		Satellite→Drone		Training Time	Testing Time	# Params (M)
FAM	FPM	R@1	mAP	R@1	mAP			
-	-	71.15	74.64	80.88	70.90	1x	1x	45.9939
✓	-	80.70	83.24	86.16	80.31	1.015x	1.032x	46.4650
-	✓	85.99	87.99	90.44	85.12	1.742x	1.067x	47.4998
✓	✓	**86.98**	**88.85**	**91.22**	**86.06**	**1.871x**	**1.069x**	**47.5944**

3) Evaluation of Individual Components: In Tab. III, we analyze the effectiveness of the proposed global representation-guided feature alignment module ("FAM") and saliency-guided self-adaptive feature partition module ("FPM"). When both FAM and FPM are removed, the model is equal to the baseline without using the proposed modules. Based on our results, we can have the following conclusions: **1)** Both "FAM" and "FPM" can effectively improve the performance. **2)** The performance of "FPM" outperforms "FAM" by a large margin, indicating that "FPM" plays a more important role. The reason is that "FPM" performs a self-adaptive partition on the feature map, enabling the model to focus on different regions of the image and align identical semantic regions from different views, which helps the model extract finer-grained clues and alleviate the difficulty of retrieval. **3)** The proposed "FAM" and "FPM" are complementary to each other that can jointly improve the performance.

Moreover, we conduct an analysis of the computational impact of the proposed module on both the training and testing phases. We regard the training time and testing time of the baseline model(without using "FAM" and "FPM") as benchmarks for comparison. The results show that the "FAM"

TABLE IV
EFFECT OF DIFFERENT POOLING METHODS USED FOR FEATURES GENERATION IN FEATURE PARTITION MODULE (FPM)."AGGFEAT-f_g" INDICATES THE AGGREGATED FEATURE MAP, "SALIVAL-S" DENOTES THE SALIENCY VALUE AND "PARTFEAT-r" REPRESENTS THE PART FEATURE. "MAX" MEANS THE MAX-POOLING, WHILE "AVG" IS THE AVERAGE-POOLING

Pooling Methods			Drone→Satellite		Satellite→Drone	
AggFeat-f_g	SaliVal-S	PartFeat-r	R@1	mAP	R@1	mAP
MAX	**AVG**	**AVG**	**86.98**	**88.85**	**91.22**	**86.06**
MAX	AVG	MAX	84.58	86.79	89.87	84.59
MAX	MAX	AVG	84.59	86.81	89.16	83.99
MAX	MAX	MAX	82.43	84.94	86.59	81.78
AVG	AVG	AVG	85.06	87.24	90.16	85.00
AVG	AVG	MAX	83.55	85.95	88.16	83.32
AVG	MAX	AVG	85.14	87.25	89.59	84.51
AVG	MAX	MAX	82.92	85.34	87.59	82.69

introduces a slight increase in both the training time and testing time compared to the baseline model. As for "FPM", this module increases training time by about 0.74 times. This is due to the additional computations required by the feature partition module during the training process. However, we have also observed that the impact of "FPM" on the testing time is minimal. The proposed modules not only do not significantly affect the inference time during the testing phase but also greatly enhance the model's performance.

Furthermore, we also measure the parameters for each variant of our proposed methods. From the results, we can observe that the model complexity increases due to the introduction of these modules. However, it is noted that the additional parameters they introduce are relatively minimal compared to the overall parameter count of the model. Despite the slight increase in parameter count, these modules play a crucial role in improving the model's performance. We believe that the benefits gained from the improved performance outweigh the relatively small increase in model complexity.

4) Effect of Different Pooling Methods in FPM: In our proposed feature partition module (FPM), we use different pooling methods to generate different features, including aggregated feature map (AggFeat-f_g), saliency value (SaliVal-S), and part feature(PartFeat-r). To study the effect of different pooling methods, we compare different combinations of pooling methods in Tab. IV."MAX" means the max-pooling, and "AVG" is the average-pooling. From the results, we can observe that using the average-pooling to generate part feature outperforms using the max-pooling, no matter what pooling method is used to generate the other two features. A similar phenomenon also occurs in "SaliVal-S". When the average-pooling is applied to generate saliency value and part feature, using max-pooling to generate aggregated feature map achieves the best performance. According to the above observations, we adopt max-pooling to generate aggregated feature map and average-pooling to generate saliency value and part feature in our proposed model.

5) Effect of the Input Image Size: A small-sized image tends to compress the fine-grained information and compromise the discriminative features of the original image, whereas larger images typically necessitate more memory resources and take longer to process during both the training and testing phases. To achieve a balance between input image size and memory consumption, we evaluate the effect of image size by maintaining the ratio between width and height as 1:1 and varying the input size from 224 to 512. The results in Tab. V show that (1) there is a gradual performance improvement when the input image size is changed from 224 × 224 to 384 × 384 and (2) continuing to increase the image size to 512 × 512 results in a reduction in performance.

TABLE V
ABLATION STUDY ON THE IMPACT OF INPUT IMAGE SIZE ON UNIVERSITY-1652

Image Size	Drone→ Satellite		Satellite→Drone	
	R@1	mAP	R@1	mAP
224*224	85.89	87.91	91.01	85.10
256*256	86.98	88.85	91.22	86.06
320*320	88.82	90.33	92.58	87.64
384*384	**89.12**	**91.10**	**92.87**	**88.52**
512*512	88.40	90.09	90.58	87.20

TABLE VI
THE PERFORMANCE OF OUR METHOD IN THE MULTIPLE-QUERY SETTING

# Queries	Drone→Satellite	
	R@1	mAP
1	86.98	88.85
2	87.38	89.20
3	88.14	89.83
6	89.27	90.80
9	90.25	91.65
18	90.94	92.31
27	91.25	92.51
54	**92.30**	**93.36**

6) Effect of Multiple-Query Setting: In real-world scenarios, UAVs often have the advantage of obtaining a greater number of images related to geographical targets. By combining the information from multiple images, a comprehensive description of the target building can be obtained. To explore whether multiple queries can improve the retrieval performance in geo-localization, we conducted experiments to evaluate the multiple-query setting by averaging the query features, with the number of multiple-query images set to 1, 2, 3, 6, 9, 18, 27, and 54. As shown in Tab. VI, the proposed method in the multiple-query setting can further improve the model's performance compared to the single-query setting. This improvement is attributed to the fact that different query images provide diverse and complementary information about the target location. By incorporating more query images, a broader range of visual cues can be captured, resulting in a more comprehensive representation of the target scene and overall performance improvement. Notably, in our experiments, we observed a consistent trend of performance improvement as the number of query images increased, which can be credited to the enhanced feature representation obtained by incorporating more target-specific information.

7) Effect of Sharing Backbone Weights: To explore whether sharing weights affects the matching accuracy, we conduct an ablation study on the setting of sharing/not sharing backbone

TABLE VII
ABLATION STUDY ON THE SETTING OF SHARING/NOT SHARING BACKBONE WEIGHTS ON UNIVERSITY1652

Setting	Drone→Satellite		Satellite→Drone	
	R@1	mAP	R@1	mAP
Not sharing weights	85.09	87.23	88.87	83.73
Sharing weights	**86.98**	**88.85**	**91.22**	**86.06**

weights on the University1652 dataset. As shown in Tab. VII, our method achieves better performance on both tasks when adopting the sharing backbone weights setting. We assume there are two main reasons for this: 1)The drone-view and satellite-view images often exhibit similar patterns, allowing the model with shared weights to capture and leverage these similarities effectively. The shared information contributes to improved feature extraction and matching between the two viewpoints, leading to enhanced geo-localization accuracy. 2)The number of satellite-view images is significantly less than the number of drone-view images. In this context, sharing the weights across both viewpoints helps prevent model overfitting to the limited satellite-view images. By leveraging the shared weights, the model can effectively transfer knowledge learned from the abundant drone-view images to extract discriminative features for satellite-view images, ultimately enhancing the overall performance of the model.

8) Effect of Different Feature Extractors on Feature Alignment Module: The feature extractor (backbone) plays a crucial role in capturing informative and discriminative features from the input data. The effectiveness of the proposed feature alignment module can be influenced by the performance and quality of the feature extractor. In order to explore the effect of different feature extractors on the feature alignment module, we conduct experiments to compare the results with and without the use of the feature alignment module across different backbone networks, including VGG-16 [50], SeResNet-50 [51], ResNet-50 [48], EfficientNet-B0 [52], ResNeXt-50 [53], DenseNet-121 [54] and Vit-S [27]. We also calculate the number of parameters and floating point operations (FLOPs) for each backbone in the experiment. To avoid the impact of the feature partition strategy, we have excluded the feature partition module from the models in this experiment. The results are shown in Tab. VIII. The experimental results demonstrate that across different backbones, the proposed feature alignment module introduces only a minor increase in parameters and computational costs, while significantly enhancing model performance. However, the degree of improvement varies across different backbones, with the feature alignment module generally showing better performance on backbones with stronger feature extraction capabilities. Specifically, the feature alignment module combined with the Vit-S backbone outperforms other backbone networks and delivers the best results with the minimal increase in FLOPs and parameters, which means that the Vit-S is the optimal feature extractor for our approach. Therefore, to minimize the impact of potential flaws in the feature extractor, we choose the Vit-S as the backbone of our model.

9) Effect of Different Loss Functions: Different loss functions encourage models to optimize towards diverse objectives. To verify the effect of different loss functions, we conduct experiments combining our proposed method with different loss functions (including triplet loss [55], kl loss [56], contrastive loss [57] and verification loss [58]), while considering the use of different backbone networks (including Vit-S [27] and ResNet50 [48]). The results are shown in Tab. IX.

For the proposed method with Vit-S as the backbone network, we can observe that incorporating any additional loss function leads to a slight decrease in model performance, while using only the cross-entropy loss function yields the optimal result. This indicates that the proposed model with Vit-S already can capture discriminative features, and the addition of other loss functions did not provide any significant improvement. On the other hand, for the proposed method with ResNet50 as the backbone network, we find that adding triplet loss, contrastive loss or verification loss can further improve the model's performance. These loss functions enhance the alignment of similar samples in feature space, resulting in better performance. However, incorporating the KL loss function has a noticeable negative impact on the model's performance.

In summary, the proposed method uses Vit-S as the backbone network and cross-entropy Loss as the loss function can achieve the best performance.

10) Effect of Jointly Fine-Tuning the Backbone and the Proposed Modules: To assess the impact of joint training with the proposed modules on the backbone network, we conduct an ablation study on the setting of fixing/fine-tuning backbone weights on the University1652 dataset. The results are shown in Tab. X. The results show a significant drop in model performance, confirming that jointly fine-tuning the backbone alongside the proposed modules is essential for achieving enhanced feature extraction capabilities and overall improved performance.

E. Visualization

To gain a deeper insight into the functionality of our proposed Safe-Net, we perform an additional qualitative experiment on visualizing the alignment results produced by the feature alignment module, the partition results calculated by the feature partition module and the heatmaps. As described in the methodology section, our proposed feature alignment module is conducted on the patch feature map rather than the original input image. To better observe the efficacy of this module, we extract the affine parameters predicted by the feature alignment module and use them to perform an affine transformation on the input image. In addition, the heatmaps are calculated by a summation operation along the channel axis on the final aligned patch feature map from the feature alignment module. Darker or redder colors on the heatmap represent higher response values from the corresponding area on the output feature map, thereby indicating the greater influence of that area on the final retrieval results.

As shown in Fig. 9, we display the visualization results of two different geographic locations. Specifically, in the figure, the satellite-view images are presented in the first and

TABLE VIII

THE PERFORMANCE AND COMPLEXITY OF MODELS WITH/WITHOUT THE FEATURE ALIGNMENT MODULE ON DIFFERENT BACKBONES ON THE DRONE-VIEW TARGET LOCALIZATION TASK ON THE UNIVERSITY-1652 DATASET. "W/O ALIGNMENT" REFERS TO MODELS WITHOUT FEATURE ALIGNMENT, WHILE "W/ ALIGNMENT" DENOTES MODELS THAT INCLUDE THE FEATURE ALIGNMENT MODULE. **FLOPS** REFERS TO FLOATING POINT OPERATIONS AND **# PARAMS** INDICATES THE NUMBER OF PARAMETERS

Backbone	w/o alignment				w/ alignment			
	R@1	mAP	# Param (M)	FLOPs (G)	R@1	mAP	# Param (M)	FLOPs (G)
EfficientNet-B0	61.06	65.90	8.53	1.8559	70.14 (+9.08)	72.99 (+7.09)	**11.39** (+2.85)	**1.8596** (+0.0037)
DenseNet-121	63.45	68.12	13.95	7.6566	72.33 (+8.88)	75.98 (+7.86)	16.30 (+2.35)	7.6597 (+0.0031)
ResNet-50	59.84	63.32	46.18	15.1640	68.87 (+9.03)	72.59 **(+9.27)**	50.54 (+4.35)	15.1696 (+0.0056)
SeResNet-50	58.21	63.09	51.01	15.1687	66.95 (+8.74)	71.63 (+8.54)	55.36 (+4.35)	15.1743 (+0.0056)
ResNeXt-50	62.34	67.04	45.18	15.1857	71.70 (+9.36)	75.41 (+8.37)	49.53 (+4.35)	15.1913 (+0.0056)
Vit-S	71.15	74.64	45.99	22.8949	**80.70 (+9.55)**	**83.24** (+8.60)	46.47 **(+0.47)**	22.8958 **(+0.0009)**
VGG-16	59.73	64.76	28.68	37.4360	66.79 (+7.06)	71.07 (+6.31)	30.03 (+1.35)	37.4379 (+0.0019)

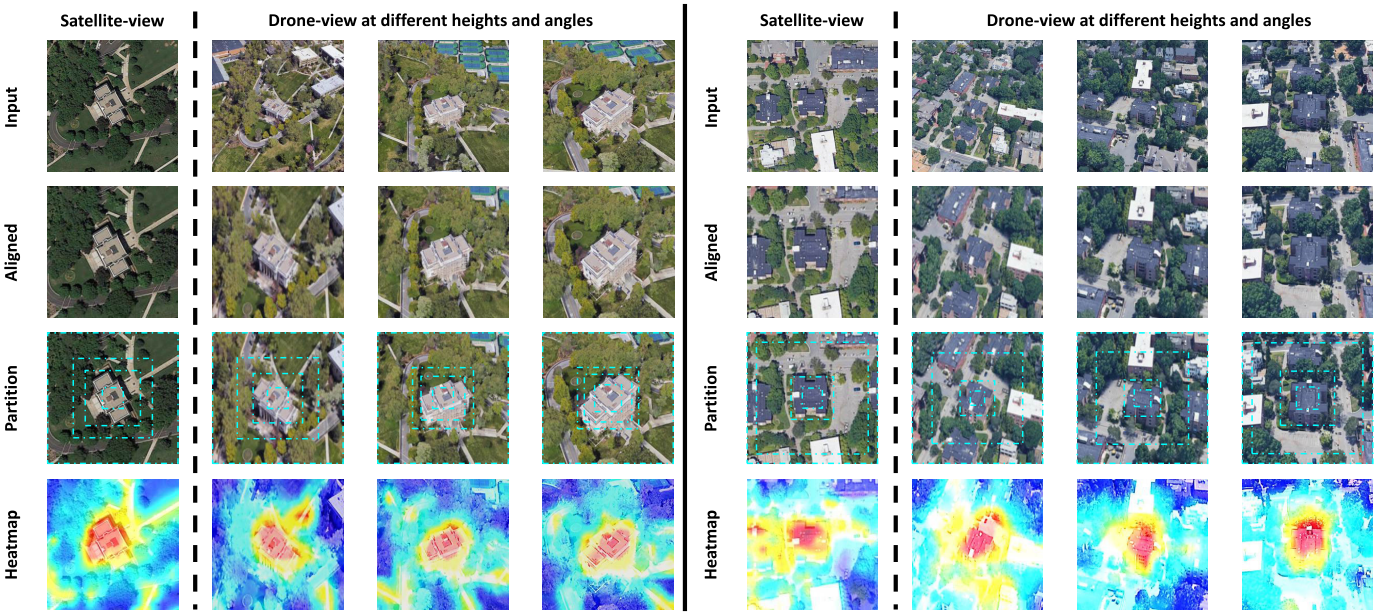


Fig. 9. Visualization results of satellite-view and drone-view images processed by our Safe-Net. The first and fifth columns display satellite-view images, while the remaining six columns show drone-view images captured from varying distances to the geographical targets. The first row exhibits the input images, and the images of the second row are aligned images generated by our proposed feature alignment module. The images of the third row are the partition results calculated by the proposed feature partition module, while the fourth row depicts the corresponding heatmaps.

TABLE IX

ABLATION STUDY ON THE EFFECT OF DIFFERENT LOSS FUNCTIONS ON THE DRONE-VIEW TARGET LOCALIZATION TASK ON UNIVERSITY1652

Method	Vit-S		Resnet50	
	R@1	mAP	R@1	mAP
Ours (Cross-Entropy Loss)	86.98	88.85	83.88	86.09
Ours + Triplet Loss(M=0.3)	86.87	88.74	84.87	86.86
Ours + KL Loss	86.04	88.01	79.47	82.24
Ours + Contrast Loss	85.67	87.73	84.29	86.52
Ours + Verification Loss	86.03	87.99	84.18	86.33

TABLE X

ABLATION STUDY ON THE IMPACT OF JOINTLY FINE-TUNING THE BACKBONE AND THE PROPOSED MODULES. THE BACKBONE NETWORK IS TRAINED USING TWO APPROACHES, NAMELY FIXED AND FINE-TUNED

Backbone	Our Modules	Drone→Satellite		Satellite→Drone	
		R@1	mAP	R@1	mAP
Fine-tuned	Fine-tuned	**86.98**	**88.85**	**91.22**	**86.06**
Fixed	Fine-tuned	72.65	76.19	87.45	72.54

fifth columns, while the drone-view images with different distances to the geographic targets are displayed in the other six columns. The first row shows the input images, and the second row showcases the aligned images generated by our proposed feature alignment module. The third row exhibits the partition results obtained by our proposed feature partition module, and the fourth row illustrates the corresponding heatmaps. According to the visualization, we can observe that: (1) For the same geographic location, the scale of the target buildings in the drone-view images changes significantly with the distance of the UAV to the target location. (2) As shown in the aligned images, the feature alignment module is capable of significantly reducing the scale variance of the target building across different drone-view images, which can alleviate the complexity of the subsequent feature partition process. (3) In the partition results, our proposed feature partition module can perform self-adaptive partition according to the visual characteristics of the target building and the layout of its surroundings. Meanwhile, the content of the same divided part from different images is similar, thus facilitating the region alignment. (4) As for heatmaps, our approach can enable the network to focus on the geographic target.

In addition, our proposed method allows the model to pay attention to the surrounding environment and obtain contextual

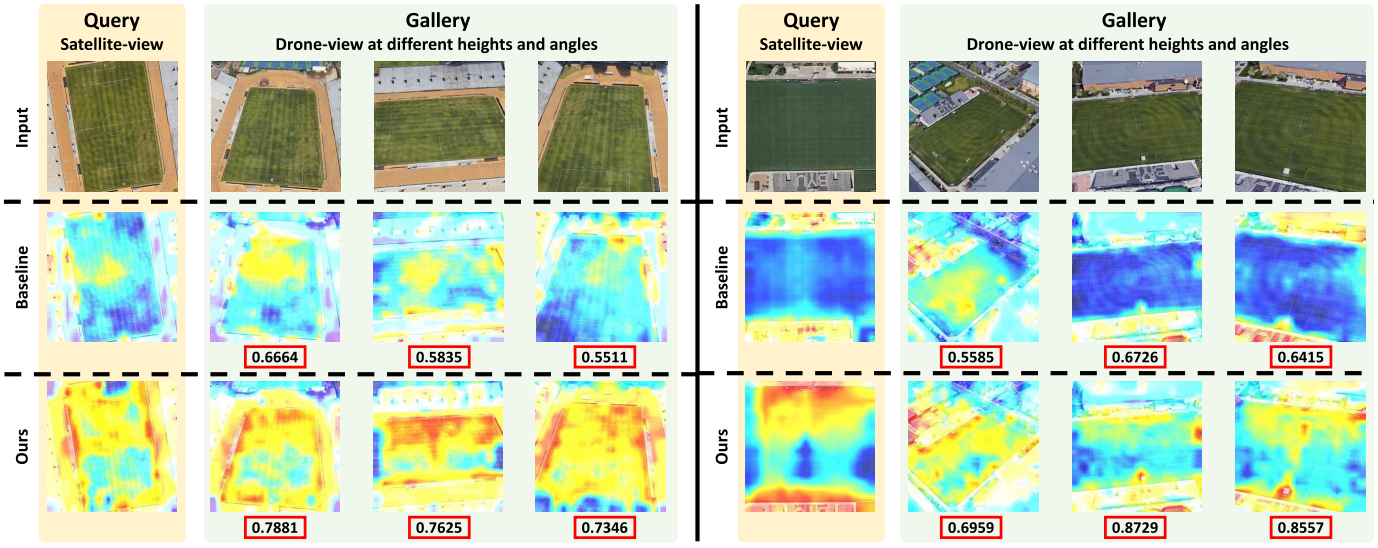


Fig. 10. Visualization of heatmaps generated by the baseline model and our method in some exceptional cases. The numbers below the heatmaps represent the similarity scores between the corresponding drone-view gallery images and satellite-view query images.

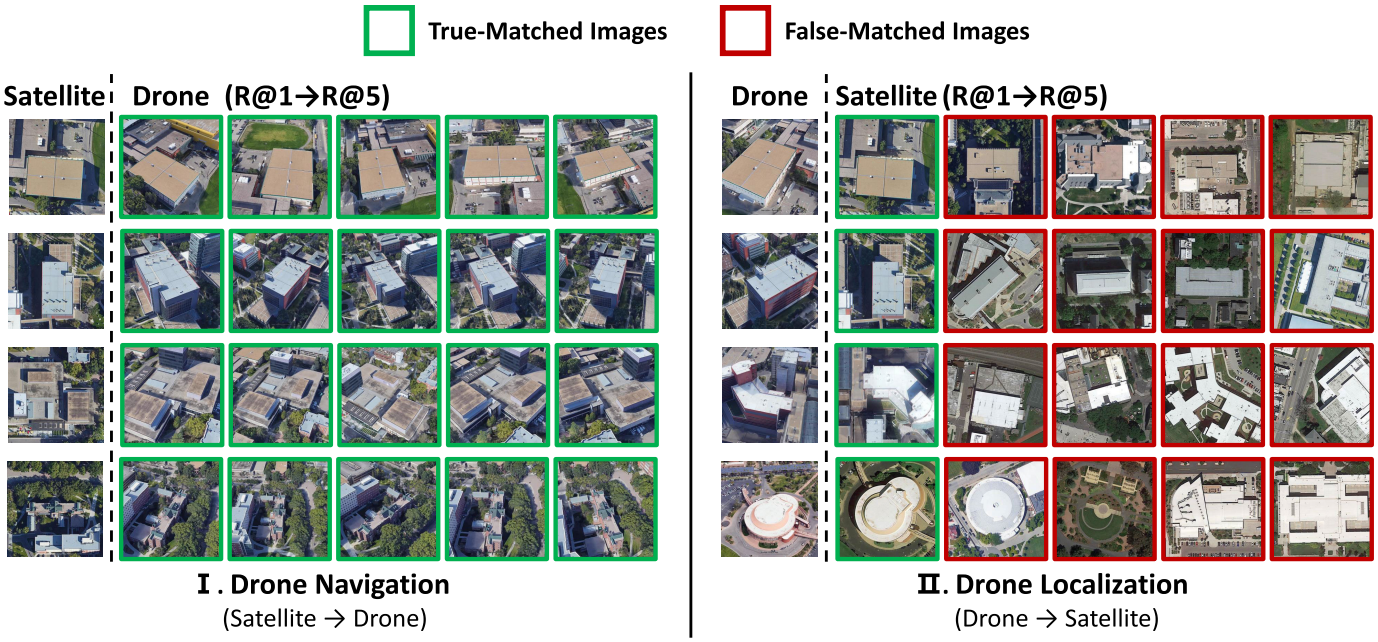


Fig. 11. Visualization of Top-5 retrieval results for drone navigation and drone localization on University-1652. The green box represents the true-matched images in retrieval results and the red box indicates the false-matched images.

information in some exceptional cases. To demonstrate the effectiveness of our approach in handling exceptional cases, we conduct visualization experiments on extreme examples, such as vast playgrounds or sports fields without obvious buildings. We also provide the similarity scores between the drone-view gallery images and the satellite-view query images, which are shown in Fig. 10. The visualization of heatmaps show that compared to the baseline model, our method is still able to extract meaningful features from the surrounding environment in some exceptional cases. Furthermore, the baseline model yields a lower similarity score between gallery images and query images, whereas our approach significantly improves the similarity scores for positive samples. The above observations indicate that our approach has the capability to capture contextual information and adapt to diverse scenarios, even in the presence of significant areas or feature distributions unrelated to the target building.

Moreover, we visualize some retrieval results for both drone navigation and drone localization tasks on the University-1652 dataset. As shown in Fig. 11, we choose four query images for each task and show the Top-5 matching images in the ranking of the retrieval results for each query image. From the retrieval results, we can observe that our model can successfully find out the true-matched images based on the content of query images. It is noted that there is only one true-matched satellite-view image in the gallery for a drone-view query image in the task of drone localization.

V. Conclusion

In this paper, we present Safe-Net, a novel and effective framework for aerial-view geo-localization that employs joint scale-invariant representation learning and feature alignment in a self-adaptive manner. In the proposed Safe-Net, we design

two powerful modules: a global representation-guided feature alignment module and a saliency-guided self-adaptive feature partition module. These two modules work together to enhance the model's robustness against scale variations. Experimental results demonstrate that Safe-Net outperforms existing approaches and achieves state-of-the-art performance on two aerial-view geo-localization datasets, *i.e.*, University-1652 and SUES-200. The proposed modules can also be easily integrated into other networks to enhance performance. In the future, we will consider exploring the combination of the proposed modules with feature matching methods (e.g., Superglue [59] and Sam [60]), or applying the proposed method to other tasks (e.g., cross-modality tasks [61], [62]).

REFERENCES

[1] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *Proc. CVPR*, Jun. 2015, pp. 5007–5015.

[2] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1437–1451, Feb. 2018.

[3] Z. Zeng, Z. Wang, F. Yang, and S. Satoh, "Geo-localization via ground-to-satellite cross-view image retrieval," *IEEE Trans. Multimedia*, vol. 25, pp. 2176–2188, 2022.

[4] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *Proc. CVPR*, Jun. 2019, pp. 5617–5626.

[5] Y. Tian, C. Chen, and M. Shah, "Cross-view image matching for geo-localization in urban environments," in *Proc. CVPR*, Jul. 2017, pp. 1998–2006.

[6] M. Modsching, R. Kramer, and K. T. Hagen, "Field trial on GPS accuracy in a medium size city: The influence of built-up," in *Proc. 3rd Workshop Positioning, Navigat. Commun.*, 2006, pp. 209–218.

[7] P. Zhu, J. Zheng, D. Du, L. Wen, Y. Sun, and Q. Hu, "Multi-drone-based single object tracking with agent sharing network," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 4058–4070, Oct. 2021.

[8] V. J. Hodge, R. Hawkins, and R. Alexander, "Deep reinforcement learning for drone navigation using sensor data," *Neural Comput. Appl.*, vol. 33, no. 6, pp. 2015–2033, Mar. 2021.

[9] H. Wei and L. Wang, "Visual navigation using projection of spatial right-angle in indoor environment," *IEEE Trans. Image Process.*, vol. 27, no. 7, pp. 3164–3177, Jul. 2018.

[10] D.-G. Sim et al., "Hybrid estimation of navigation parameters from aerial image sequence," *IEEE Trans. Image Process.*, vol. 8, no. 3, pp. 429–435, Mar. 1999.

[11] Y. Ke, R. Sukthankar, and M. Hebert, "Event detection in crowded videos," in *Proc. ICCV*, Oct. 2007, pp. 1–8.

[12] J. Choi, G. Sharma, M. Chandraker, and J.-B. Huang, "Unsupervised and semi-supervised domain adaptation for action recognition from drones," in *Proc. WACV*, Mar. 2020, pp. 1706–1715.

[13] A. M. Algamdi, V. Sanchez, and C.-T. Li, "DroneCaps: Recognition of human actions in drone videos using capsule networks with binary volume comparisons," in *Proc. ICIP*, Oct. 2020, pp. 3174–3178.

[14] Y. Yan et al., "Event oriented dictionary learning for complex event detection," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1867–1878, Jun. 2015.

[15] S. Deng et al., "A global–local self-adaptive network for drone-view object detection," *IEEE Trans. Image Process.*, vol. 30, pp. 1556–1569, 2021.

[16] H. Zhang, G. Wang, Z. Lei, and J.-N. Hwang, "Eye in the sky: Drone-based object tracking and 3D localization," in *Proc. ACM MM*, 2019, pp. 899–907.

[17] T. Peng, Q. Li, and P. Zhu, "RGB-T crowd counting from drone: A benchmark and MMCCN network," in *Proc. ACCV*, 2020, pp. 497–513.

[18] Z. Zheng, Y. Wei, and Y. Yang, "University-1652: A multi-view multi-source benchmark for drone-based geo-localization," in *Proc. ACM MM*, Oct. 2020, pp. 1395–1403.

[19] L. Ding, J. Zhou, L. Meng, and Z. Long, "A practical cross-view image matching method between UAV and satellite for UAV-based geo-localization," *Remote Sens.*, vol. 13, no. 1, p. 47, Dec. 2020.

[20] J. Lin et al., "Joint representation learning and keypoint detection for cross-view geo-localization," *IEEE Trans. Image Process.*, vol. 31, pp. 3780–3792, 2022.

[21] T. Wang et al., "Each part matters: Local patterns facilitate cross-view geo-localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 2, pp. 867–879, Feb. 2022.

[22] X. Tian, J. Shao, D. Ouyang, and H. T. Shen, "UAV-satellite view synthesis for cross-view geo-localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4804–4815, Jul. 2022.

[23] M. Dai, J. Hu, J. Zhuang, and E. Zheng, "A transformer-based feature segmentation and region alignment method for UAV-view geo-localization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4376–4389, Jul. 2022.

[24] J. Zhuang, M. Dai, X. Chen, and E. Zheng, "A faster and more effective cross-view matching method of UAV and satellite images for UAV geolocalization," *Remote Sens.*, vol. 13, no. 19, p. 3979, Oct. 2021.

[25] J. Zhuang, X. Chen, M. Dai, W. Lan, Y. Cai, and E. Zheng, "A semantic guidance and transformer-based matching method for UAVs and satellite images for UAV geo-localization," *IEEE Access*, vol. 10, pp. 34277–34287, 2022.

[26] R. Zhu, L. Yin, M. Yang, F. Wu, Y. Yang, and W. Hu, "SUES-200: A multi-height multi-scene cross-view image benchmark across drone and satellite," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 9, pp. 4825–4839, Sep. 2023.

[27] A. Dosovitskiy et al., "An image is worth 16×16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[28] S. Workman and N. Jacobs, "On the location dependence of convolutional neural network features," in *Proc. CVPRW*, Jun. 2015, pp. 70–78.

[29] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[30] B. Zhou, A. Lapedriza, J. Xiao, A. Torralab, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. NeurIPS*, 2014, pp. 487–495.

[31] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *Proc. ECCV*, 2016, pp. 494–509.

[32] S. Hu, M. Feng, R. M. H. Nguyen, and G. H. Lee, "CVM-Net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7258–7267.

[33] S. Cai, Y. Guo, S. Khan, J. Hu, and G. Wen, "Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Feb. 2019, pp. 8390–8399.

[34] R. Rodrigues and M. Tani, "Are these from the same place? Seeing the unseen in cross-view image geo-localization," in *Proc. WACV*, Jan. 2021, pp. 3752–3760.

[35] K. Regmi and M. Shah, "Bridging the domain gap for ground-to-aerial image matching," in *Proc. ICCV*, Oct. 2019, pp. 470–479.

[36] I. Goodfellow et al., "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680.

[37] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, Nov. 1986.

[38] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," in *Proc. NeurIPS*, 2019, pp. 10090–10100.

[39] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, "Optimal feature transport for cross-view image geo-localization," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11990–11997.

[40] Y. Shi, X. Yu, D. Campbell, and H. Li, "Where am i looking at? Joint location and orientation estimation by cross-view matching," in *Proc. CVPR*, Jun. 2020, pp. 4063–4071.

[41] H. Yang, X. Lu, and Y. Zhu, "Cross-view geo-localization with layer-to-layer transformer," in *Proc. NeurIPS*, 2021, pp. 1–12.

[42] S. Zhu, M. Shah, and C. Chen, "TransGeo: Transformer is all you need for cross-view image geo-localization," in *Proc. CVPR*, Jun. 2022, pp. 1162–1171.

[43] A. Vaswani et al., "Attention is all you need," in *Proc. NeurIPS*, vol. 30, 2017, pp. 5998–6008.

[44] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, M. Xu, and Y.-D. Shen, "Dual-path convolutional image-text embeddings with instance loss," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 2, pp. 1–23, May 2020.

[45] D. V. Bui, M. Kubo, and H. Sato, "A part-aware attention neural network for cross-view geo-localization between UAV and satellite," *J. Robot., Netw. Artif. Life*, vol. 9, no. 3, pp. 275–284, Dec. 2022.

[46] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, Dec. 2015, pp. 2017–2025.
[47] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. ICLR*, Jan. 2016, pp. 1–13.
[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.
[49] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," in *Proc. PRIA*, Jan. 2009, pp. 11–14.
[50] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[51] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7132–7141.
[52] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6105–6114.
[53] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. CVPR*, Jul. 2017, pp. 1492–1500.
[54] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. CVPR*, Jul. 2017, pp. 4700–4708.
[55] E. Hoffer and N. Ailon, "Deep metric learning using triplet network," in *Proc. 3rd Int. Workshop Similarity-Based Pattern Recognit. (SIMBAD)*, Copenhagen, Denmark. Heidelberg, Germany: Springer, Oct. 2015, pp. 84–92.
[56] Y. He, C. Zhu, J. Wang, M. Savvides, and X. Zhang, "Bounding box regression with uncertainty for accurate object detection," in *Proc. CVPR*, Jun. 2019, pp. 2888–2897.
[57] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. CVPR*, Jun. 2006, pp. 1735–1742.
[58] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned CNN embedding for person reidentification," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 1, pp. 1–20, Dec. 2017.
[59] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "SuperGlue: Learning feature matching with graph neural networks," in *Proc. CVPR*, Jun. 2020, pp. 4938–4947.
[60] X. Lu, Y. Yan, T. Wei, and S. Du, "Scene-aware feature matching," in *Proc. ICCV*, Oct. 2023, pp. 3681–3690.
[61] D. Zhang, G. Huang, Q. Zhang, J. Han, J. Han, and Y. Yu, "Cross-modality deep feature learning for brain tumor segmentation," *Pattern Recognit.*, vol. 110, Feb. 2021, Art. no. 107562.
[62] C. Fang, D. Zhang, L. Wang, Y. Zhang, L. Cheng, and J. Han, "Cross-modality high-frequency transformer for MR image super-resolution," in *Proc. 30th ACM Int. Conf. Multimedia*, Oct. 2022, pp. 1584–1592.

Jinliang Lin received the M.S. degree from the Department of Artificial Intelligence, Xiamen University, China, in 2022, where he is currently pursuing the Ph.D. degree. His research interests include cross-view geo-localization and fine-grained visual recognition.

Zhiming Luo (Member, IEEE) received the B.S. degree from the Cognitive Science Department, Xiamen University, Xiamen, China, in 2011, and the dual Ph.D. degree in computer science from Xiamen University and University of Sherbrooke, Sherbrooke, QC, Canada, in 2017. His research interests include traffic surveillance video analytics, computer vision, and machine learning.

Dazhen Lin received the B.S. degree in computer science, the M.S. degree in computer application technology, and the Ph.D. degree in mathematics from Xiamen University, China, in 2002, 2005, and 2012, respectively. Her research interests include natural language processing, information retrieval, computer vision, and pattern recognition.

Shaozi Li received the B.S. degree from Hunan University, the M.S. degree from Xi'an Jiaotong University, and the Ph.D. degree from the National University of Defense Technology. He is currently the Chair and a Professor with the Cognitive Science Department, Xiamen University, and the Vice Director of the Technical Committee on Collaborative Computing of CCF and Fujian Association of Artificial Intelligence. He has directed and completed more than 20 research projects, including several National 863 Programs, National Nature Science Foundation of China, and Ph.D. Programs Foundation of Ministry of Education of China. His research interests include artificial intelligence and its applications, moving objects detection and recognition, machine learning, computer vision, and multimedia information retrieval. He is also a Senior Member of ACM and China Computer Federation (CCF).

Zhun Zhong received the Ph.D. degree from the Department of Artificial Intelligence, Xiamen University, China, in 2019. He was a joint Ph.D. Student with the University of Technology Sydney, Australia. He was a Postdoctoral Researcher with the University of Trento, Italy, and an Assistant Professor with the University of Nottingham, U.K. He is currently a Professor with Hefei University of Technology, China. His research interests include person re-identification, novel class discovery, data augmentation, and domain adaptation.