AST: An Attention-guided Segment Transformer for Drone-based Cross-view Geo-localization*

Zichuan Zhao, Tianhang Tang, Jie Chen, Xuelei Shi, and Yiguang Liu

Sichuan University, No.24 South Section 1, Yihuan Road, Chengdu, China
liuyg@scu.edu.cn

Abstract. To tackle the problem of drone-based cross-view geo-localization, we address how to match drone-view images and satellite-view images, which is extremely challenging due to the variability of view angles and view distances. Inspired by how humans recognize aerial images, we propose an effective Attention-guided Segment Transformer (AST) structure: a novel segmentation strategy is introduced to cope with the huge variations between aerial views, and this segmentation is adaptive and non-uniform, allowing it to segment regions with corresponding relationships even after significant changes in viewpoint; furthermore, a new segment token module is designed to generate segment tokens that are concatenated with the original class token to supplement the local information. Compared to CNN-based methods, AST fully utilizes the self-attention mechanism to establish global context correlations; and the newly introduced segment token module allows AST to effectively extract local features as well — a capability not present in the vanilla vision transformer. Remarkably, AST demonstrates good robustness to viewpoint changes, even when there are overlapping regions, and this good treat is confirmed by the experimental results on the University-1652 dataset, which also show competitive performance for both tasks of drone-view target localization and drone navigation.

Keywords: Geo-localization · Image retrieval · Drone-based cross-view.

1 Introduction

Image-based cross-view geo-localization involves matching images that depict the same geospatial location but are captured from different views or platforms. This can be considered an image retrieval task. As the automation industry develops and satellite imaging technology matures, image-based cross-view geo-localization has become increasingly important. For example, in situations where GPS signals are weakened or lost due to interference, unmanned devices require an alternative independent positioning method, such as matching images of the surrounding environment with geo-tagged images to determine their locations.

* This work is supported by NSFC under grants U19A2071 and 61860206007, Sichuan Science and Technology Program under grant 2023YFG0334, as well as the funding from Sichuan University under grant 2020SCUNG205.

Image-based geo-localization has been applied to various real-world fields such as autonomous driving, robot positioning, drone navigation, and precision delivery. Compared to sensor-based positioning methods, image-based methods have several advantages, including lower cost, stronger resistance to electromagnetic interference, and better environmental adaptability.

In recent years, research on image-based cross-view geo-localization has mainly focused on matching images between ground-level and satellite-level. And ground-to-aerial datasets such as CVUSA[37] and CVACT[21] have emerged. However, matching images between drone-view images and satellite-view images has received less attention. University-1652[38] is the first drone-based geo-localization dataset that expands cross-view geo-localization from ground-satellite imagery to drone-satellite imagery and brings two new tasks: drone-view target localization and drone navigation. This expansion facilitates deep learning research on image-based cross-view geo-localization.

Researchers typically use a Siamese-like network architecture to tackle cross-view geo-localization tasks[8, 21, 28, 29, 33, 36, 40]. Identifying similarities between images from different views or platforms is the key to solving this problem. Humans tend to prioritize landmark buildings or patterns in aerial images and then analyze surrounding areas before making judgments based on global information. Inspired by this, we believe that image-based cross-view geo-localization should fully extract global image information and establish global context correlations. While CNN-based approaches often focus on small discriminative regions since the effective receptive field size of deep convolutional neural networks is Gaussian distributed[24], they ignore global context correlations, which can be detrimental for cross-view geo-localization. Therefore, we have chosen to use the Vision Transformer (ViT)[12] as our backbone.

In this paper, we introduce an Attention-guided Segment Transformer (AST) structure to address two tasks in the University-1652 dataset, i.e., drone-view target localization and drone navigation. To enhance the network's robustness to viewpoint changes, we propose an adaptive segmentation strategy. Utilizing the self-attention mechanism of the transformer, patches of similar importance are grouped together to form multiple non-uniform regions of decreasing significance. These regions can effectively adapt to variations in targets' position and size across different views — a crucial feature for aerial images. This process mirrors how humans match cross-view images, where attention is first directed towards key regions before expanding to surrounding regions. To enhance the extraction of local features, we design a new segment token module that generates segment tokens carrying additional local information for each region. These segment tokens are concatenated with the class token to form the final embedding features for matching. The segment tokens also enable spatial alignment between corresponding regions. Compared to CNN-based methods, AST establishes global context correlations while also focusing on local information. In contrast, the vanilla vision transformer mainly focuses on global features.

In summary, the main contributions of this paper are as follows:

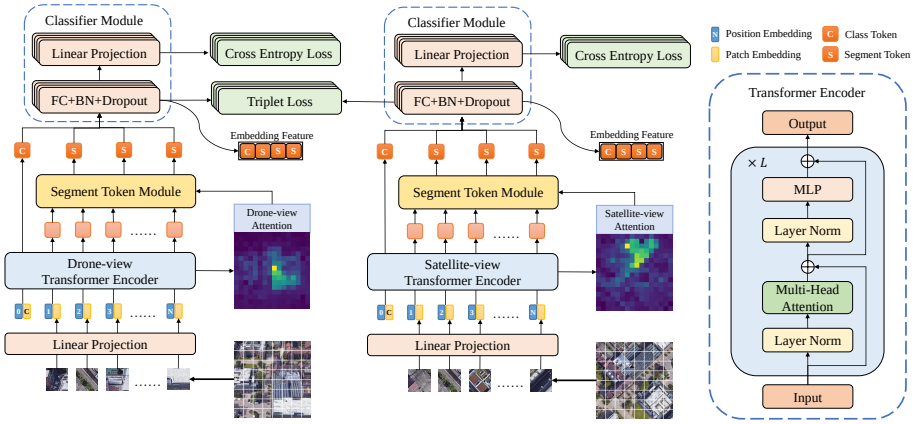

Fig. 1. An overview of our proposed AST framework. While training, the output of the class token and those segment tokens are fed into classifier modules that do not share parameters, and all the tokens are trained, respectively. While testing, the class token and segment tokens before linear projection layers are concatenated as the embedding feature for cross-view geo-localization tasks.

1) A novel attention-guided segmentation strategy. The segmentation is adaptive, the regions are non-uniform, and no human intervention is required, so it can flexibly respond to changes in viewpoint.
2) A new segment token module. It enhances local information and achieves spatial alignment between cross-view images. Besides, it is easy to implement and has the potential to be fused with other backbones as long as the attention mechanism is available.
3) An effective Attention-guided Segment Transformer (AST) structure. AST outperforms the baseline model of University-1652 by a large margin on both benchmarks and achieves competitive results compared to existing methods. Astonishingly, experiments show that AST has good robustness to changes in viewpoint, even when there are overlapping regions.

The remainder of the paper is structured as follows: We briefly introduce several pertinent work in Section II. Section III details our designed AST. Section IV presents the experimental results, while Section V offers the conclusion.

2 Related work

2.1 Image-based Cross-view Geo-localization.

In recent years, image-based cross-view geo-localization has attracted a lot of attention due to its huge application potential. The large changes in viewpoint and the differences between imaging platforms make cross-view image matching more difficult. Inspired by Siamese Network[6], Lin *et al.*[20] apply it to

image-based cross-view geo-localization, and many of the subsequent methods also adopt Siamese-like architecture. Our proposed AST has a Siamese-like architecture, too.

In order to deal with the large changes in viewpoint in cross-view geo-localization, a lot of methods have been proposed. Zhai *et al.*[37] use a VGG network to generate semantic segmentation and then apply an adaptive transformation to map aerial semantic segmentation into the ground-level perspective. Furthermore, Toker *et al.*[30] synthesize street views from satellite images, and Tian *et al.*[29] use a CGAN to conduct drone-satellite view synthesis. Besides, Hu *et al.*[17] combine the feature extractor with the NetVLAD, creating a model called CVM-Net, and introduce an effective weighted soft-margin ranking loss function, which speeds up its training convergence and improves its performance. Recently, Wang *et al.*[33] propose a square-ring partition strategy to take contextual patterns into consideration, which can be fused with existing methods to further boost performance. Lin *et al.*[19] combine representation learning and keypoint detection, which enhances the model's capability against large changes in viewpoint. With the rise of ViT, Yang *et al.*[36] propose a Layer-to-Layer Transformer (L2LTR) to model global dependencies, which decreases visual ambiguities. Zhu *et al.*[40] propose an "attend and zoom-in" strategy by taking advantage of ViT. And these two methods mainly focus on matching images between ground-level and satellite-level. Previous research has shown that researchers are attempting to determine the transformational relationship between different views. However, fixed transformations lack flexibility and hinder the creation of a robust feature space. Our adaptive segmentation strategy significantly alleviates this issue.

2.2 Vision Transformer

Transformer[31] was first proposed for large-scale pre-training in natural language processing (NLP) tasks and demonstrated its excellent performance and great potential[2, 9]. ViT[12] is the first pure transformer-based architecture applied to classify the full images and achieves excellent performance with substantially fewer computational resources to train compared to other CNN-based methods.

With the proposal of ViT, a series of variants have come up to improve the performance of transformer in vision tasks. As ViT simply projects an image patch into a vector (patch token) through linear mapping, the extraction of local features is ignored. Han *et al.*[13] design a new architecture termed Transformer-iN-Transformer (TNT). It further divides patches into smaller patches (sub-patches) and applies an inner transformer block to excavate finer features and details. Similarly, Swin[22], Cswin[11], and Twins[7] are also working in this direction. In addition, improving the calculation of self-attention is another noteworthy direction. DeepViT[39] introduces a re-attention mechanism that enhances information exchange among attention heads. Similarly, there are KVT[32] and XCiT[1]. Moreover, many researchers try to improve vision transform architecture. Learning from CNN, many new architectures are proposed. For example,

the pyramid-like architecture is adapted in PVT[34], HVT[25], PiT[15], and so on. Some other architectures are also applied, e.g., two-stream architecture[5] and U-net architecture[4, 35]. Researchers have made important contributions to the improvement of ViT by enhancing locality, improving self-attention, and designing new architectures. However, we discover that the majority of these improvements concentrate on getting a better global class token while disregarding patch tokens, whereas our suggested segment token module can utilize patch tokens more effectively.

3 Proposed Method

3.1 Problem Formulation

In the University-1652[38] dataset, each satellite-view image has 54 corresponding drone-view images, and there are two tasks that we need to do:
Drone-view target localization (Drone → Satellite). Given one query drone-view image, the task aims to find the most similar geo-tagged satellite-view image so that the target building can be located. This is a many-to-one match task in the University-1652 dataset.
Drone navigation (Satellite → Drone). Given one query satellite-view image, the drone aims to find the most relevant place (drone-view images) according to its flight history so that it can be navigated back to the target place. Also, a set of satellite images can be given to guide the drone step by step. This is a one-to-many match task in the University-1652 dataset.

In brief, we aim to output the ranking of the gallery images that are most similar to the query image. Therefore, we regard the two tasks as cross-view image retrieval problems. In the training set, we have a set of drone-view images $\{I_d\}$, a set of satellite-view images $\{I_s\}$, and class labels $\{y\}$ corresponding to all the images. Images are classified by geo-tags or target buildings. We aim to train a neural network to identify a mapping function $F(.)$ that could project drone-view images and satellite-view images to a shared feature space. In this space, the feature vectors with the same label are close together, while those with different labels are separated. When given a query image, we can extract its feature vector, which can subsequently be used to search for the closest gallery images' feature vectors in the shared feature space. We use $D(F(x_d), F(x_s)), x_d \in \{I_d\}, x_s \in \{I_s\}$ to measure the distance between feature vectors and apply the superscript y to represent images' corresponding labels. The optimal situation in the shared feature space can be expressed as follows:

$$\begin{aligned} &\forall x_d^y \in \{I_d\}, \forall x_s^{y'} \in \{I_s\}, y \neq y', \\ &\quad D(F(x_d^y), F(x_s^y)) < D(F(x_d^y), F(x_s^{y'})) \end{aligned} \tag{1}$$

where the positions of subscripts d and s can be switched. Usually, it doesn't matter whether the feature vectors with the same label are close if they are from the same view, as it doesn't affect the matching result directly.

3.2 Vision Transformer for Cross-view Geo-localization

We briefly introduce the transformer structure adapted in AST, including patch embedding, class token, position embedding, and transform encoder.

Patch Embedding: Different from CNN, we need to convert an image into some tokens as the input of transformer encoder. Given the input images $x \in \mathbb{R}^{H\times W\times C}$, where H, W, C are respectively the height, width, and channel numbers of x. Firstly, the input images are divided into N patches with the same size $P \times P$ (usually $P = 16$), therefore $N = HW/P^2$. Then all the patches are reshaped into a 2D matrix $x_p \in \mathbb{R}^{N\times P^2C}$. By adopting a trainable linear projection layer, we will get N tokens $x_t \in \mathbb{R}^{N\times D}$, where D is a hyperparameter representing the feature dimension of transformer encoder.

Class Token: Referring to ViT[12] and BERT[9], our vision transformer also has an extra learnable class token with the same dimension D in front of the N tokens, which is used to integrate the features of each patch as the global feature, and we get $x_t \in \mathbb{R}^{(N+1)\times D}$. After passing through the last transformer encoder, the output class token will become an important reference for subsequent cross-view geo-localization tasks, as will the segment token, which we propose and will introduce later.

Position Embedding: As images are divided into patches, the relative position relationships between patches are ignored. Referring to ViT[12], to make up for the information loss, learnable position embedding is adopted in our vision transformer. It is a learnable matrix $x_{pos} \in \mathbb{R}^{(N+1)\times D}$. Now we get the input of transformer encoder $x_{in} = x_t + x_{pos}$. The explicit position embedding of each patch enables the model to better learn the geometric correspondence between different views. Besides, it makes us able to improve the model's performance by flexibly aligning patches.

Transformer Encoder: As shown in Figure 1, each transformer encoder has multiple cascaded transformer block layers. Each block consists of layer norm (LN), multi-head self-attention (MSA), and multi-layer perceptron (MLP), where MSA plays a key role. It converts the input matrix into three matrices Q, K, V through a learnable linear projection layer. They represent the query, key, and value of all the tokens, respectively. Then we can obtain the attention map of the input image by the following computation:

$$z = LN(x_{in}) \tag{2}$$

$$Q = zW^q, K = zW^k, V = zW^v \tag{3}$$

$$A = softmax\left(\frac{QK^T}{\sqrt{D}}\right)V \tag{4}$$

where W^q, W^k, W^v are linear projection matrices, K^T means the transpose of K and D is feature dimension. An h-head attention module performs linear projection with h different heads and conducts subsequent self-attention computations in parallel. For each token, the h outputs are concatenated and projected back to a vector with dimension D. The generated global attention map can distinguish the importance of different patches. However, it is important to note that in

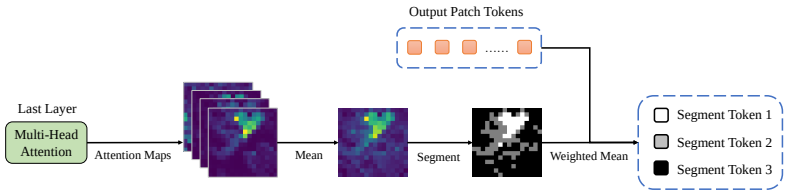

Fig. 2. Pipeline for generating segment tokens. The h-head attention module generates h different attention maps that represent different points of the input images. The lighter the pixel is, the more important the patch is. Then, with the guide of segmentation information, we can figure out the corresponding segment tokens for each group.

the vanilla vision transformer, only the output class token is considered at the end, while the rest of the tokens are ignored. We believe that these discarded patch tokens contain valuable information that can be leveraged to enhance the model's understanding of images. Our ablation studies (1) and (2) confirm this idea.

3.3 Attention-guided Segment Tokens

In this subsection, we will introduce the pipeline for generating segment tokens, which includes our segmentation strategy and the segment token module.

The global attention map highlights the importance of different patches in a manner similar to how humans recognize aerial images. On this basis, we propose an attention-guided segment token that simulates the human process of matching cross-view images. Segment tokens are calculated in the last transformer block layer, as shown in Figure 2. In the last MSA, we can get h attention maps of different heads $\{\boldsymbol{A_i}|i \in \{1, 2, ..., h\}, \boldsymbol{A_i} \in \mathbb{R}^{(N+1)\times(N+1)}\}$. Then we obtain the integrated attention map by doing an average operation. The computation is formulated as follows:

$$\boldsymbol{A_i} = \frac{Q_i K_i^T}{\sqrt{D}} \qquad i \in \{1, 2, ..., h\} \tag{5}$$

$$\boldsymbol{A} = \sum_{i=1}^{h} \boldsymbol{A_i} \tag{6}$$

Based on the attention map, we can segment output tokens into groups, excluding the class token. These tokens are segmented into three groups according to their corresponding values in the attention map: most important, less important, and least important. It is important to note that we only consider the row corresponding to the class token in matrix $\boldsymbol{A}$ and reshape the row to its original grid size as the attention map. Then we use the following formula to calculate the proportion of tokens in each group:

$$softmax([1, 2, ..., N_{group}]) \tag{7}$$

where N_{group} is the number of groups and $[1, 2, ..., N_{group}]$ is a vector that increases from 1 to N_{group}. The smaller the number in the vector is, the smaller the percentage is, and the more important the group is. We then perform $softmax$ operations on the attention values of each group separately to obtain token weights within each group. During grouping, we do not interfere with the original computation of the last MSA. Instead, we obtain additional grouping and weight information based on the global attention map. After the last transformer block layer, we compute a weighted mean of tokens within each group as segment tokens using the following formula:

$$x_{seg_i} = \sum_{j=1}^{N_i} w_{ij} x_{ij} \qquad i \in \{1, 2, ..., N_{group}\} \tag{8}$$

where x_{seg_i} is the segment token of group i, N_i is the number of tokens in group i, w_{ij} is the weight of the j-th patch token x_{ij} in group i. Compared to the vanilla vision transformer, we take into consideration $[x_{cls}, x_{seg_1}, x_{seg_2}, ...]$.

3.4 Loss Function and Training Strategy

In the vanilla vision transformer, the output class token is then fed into an MLP or a classifier module to generate the final embedding feature. We feed $[x_{cls}, x_{seg_1}, x_{seg_2}, ...]$ into classifier modules that do not share parameters to generate multiple embedding features for different parts. The classifier module consists of a fully connected layer (FC), a batch normalization layer (BN), a dropout layer (Dropout), and a linear projection layer. During training, the classifier module is used to predict the class of each part. We can simply minimize the sum of the cross-entropy losses over all parts to optimize the network. To further optimize the distribution of embedding features of different parts in the shared feature space, we train the embedding features with weighted soft-margin triplet loss[17, 40], respectively. The weighted soft-margin triplet loss function can be formulated as follows:

$$T(d_{pos}, d_{neg}) = log\left(1 + e^{\alpha(d_{pos} - d_{neg})}\right) \tag{9}$$

where d_{pos} and d_{neg} denote the squared l_2 distance of embedding features of the positive and negative pairs in a mini-batch, and α is a coefficient.

For each class, we sample one satellite-view image and N_{sample} drone-view images to form N_{sample} image pairs (images are from different views) in an epoch. While calculating triplet loss, we have B drone-view images and B satellite-view images after augmentation. B is the mini-batch size. We suppose one of these images is the query image, and each query image has B image pairs. Image pairs with the same label are positive pairs; otherwise, they are negative pairs. We can calculate the triplet loss of one part for the query image using the following formula:

$$\mathcal{L}_{tri} = \frac{1}{n_{pos} n_{neg}} \sum_{i=1}^{n_{pos}} \sum_{j=1}^{n_{neg}} T(d_{pos}^i, d_{neg}^j) \tag{10}$$

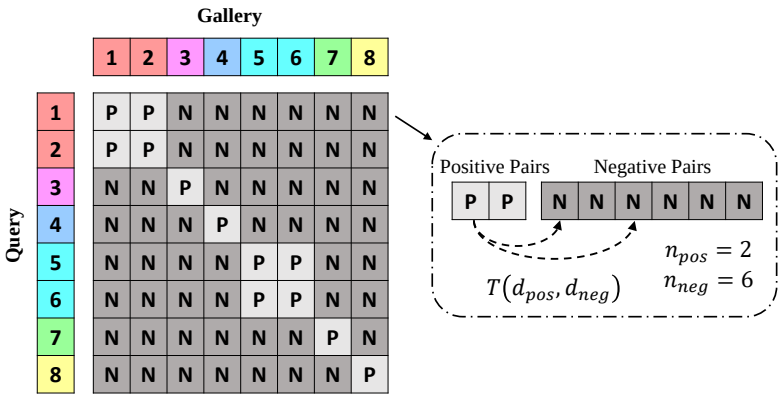

Fig. 3. Different colors stand for different labels. Image pairs with the same label are positive pairs (P); otherwise, they are negative pairs (N). Each query image has n_{pos} positive pairs and n_{neg} negative pairs in a mini-batch. d_{pos} and d_{neg} denote the squared l_2 distance of embedding features of the positive and negative pairs. We calculate $T(d_{pos}, d_{neg})$ for each possible combination and do an average to get the final triplet loss of this query image.

where n_{pos} and n_{neg} are numbers of positive and negative pairs for this query image, as shown in Figure 3. Finally, we minimize the sum of the cross-entropy losses and the triplet losses over all parts to optimize the network. While testing, we obtain the embedding features of different parts before the linear projection layer, and they are concatenated as the final embedding feature of an input image for matching.

4 Experiment

4.1 Datasets and Evaluation Metrics

We train our model and conduct experiments on the University-1652[38] dataset. It is a large-scale multi-view, multi-source dataset used for two drone-based geo-localization tasks. It selects 1,652 architectures from 72 universities around the world as target locations. For each target location, there are multiple synthetic drone-view images, which are generated from different angles, different heights, and different distances by Google Earth. According to University-1652[38], the model trained on this dataset also has good generalization ability and still works on the real-world drone-view images. Following the University-1652, we use Recall@K (**R@K**) and Average Precision (**AP**) to evaluate the performance of our proposed method. More details are shown in Table 1.

Split	Views	Images	Classes	Universities
Train	Drone Satellite	37,854 701	701	33
Query	Drone Satellite	37,855 701	701	39
Gallery	Drone Satellite	51,355 951	951	

Table 1. The detailed statistics of University-1652 training and test sets.

4.2 Implementation Details

We implement our method using Pytorch[26]. Both the drone-view images and the satellite-view images are resized to 256×256. For each class, we sample $N_{sample} = 8$ image pairs in one epoch. We adopt a small-size Vision Transformer (ViT-S)[12] as our backbone. We set $N_{group} = 3$, which means there is 1 class token and 3 segment tokens, and the final concatenated feature dimension is 2048. The transformer backbones of two branches share the same parameters. We adopt a stochastic gradient descent (SGD) optimizer. Our model is trained for 20 epochs in total with a mini-batch size 8. While training and testing, we use cosine similarity to measure two feature vectors, which is equivalent to measuring distance. All the experiments were performed on the Nvidia RTX 3090 GPU.

4.3 Comparison with Existing Methods

Methods	Drone→Satellite		Satellite→Drone	
	R@1	AP	R@1	AP
Contrastive Loss[38]	52.39	57.44	63.91	52.24
Weighted Soft Margin Triplet Loss[38]	53.21	58.03	65.62	54.47
Instance Loss + GeM Pooling[27]	65.32	69.61	79.03	65.35
LCM (ResNet-50)[10]	66.65	70.82	79.89	65.38
LPN[33]	75.93	79.14	86.45	74.79
LPN + CA-HRS[23]	76.67	79.77	86.88	74.84
Instance Loss + USAM + LPN[19]	77.60	80.55	86.59	75.96
LDRVSD[19]	78.66	81.55	89.30	79.17
PCL[29]	79.47	83.63	87.69	78.51
FSRA[8]	84.51	86.71	88.45	83.37
PAAN[3]	84.51	86.78	**91.01**	82.28
AST + Contrast Loss	85.45	87.52	90.44	84.81
AST + Weighted Soft-margin Triplet Loss	**86.29**	**88.20**	89.59	**85.06**

Table 2. Comparison with existing methods in terms of R@1 and AP on University-1652. The first two methods serve as baseline models and are distinguished by their use of different loss functions.

We evaluate our proposed AST on the University-1652 dataset and compare its performance with existing methods, as shown in Table 2. The first two methods serve as baseline models and are distinguished by their use of different loss functions. LPN (CNN-based) adopts a similar training strategy, and it segments the image into fixed square-ring blocks. FSRA optimizes LPN by using ViT to replace the backbone and segments the image into several regions of the same area. PAAN also optimizes LPN by combining the SE-block[16] and ResNet-50[14] to replace the backbone, but still segments images into square-ring blocks. And we make full use of the self-attention mechanism, making the segmentation more flexible and reasonable, and assigning different weights to each patch, thus having better robustness to the change of viewpoint. PCL uses CGAN to perform perspective transformation to reduce the differences between cross-view images, but the information loss caused by occlusion is still difficult to make up for. Therefore, we not only pay attention to the overall information of the image, but also pay more attention to the key regions, and combine the surrounding regions to generate a discriminative feature vector. These improvements all lead to better performance of AST. When it comes to PAAN, we believe it might be more applicable at certain shooting angles and distances and thus achieve a higher R@1 value but a lower AP value. AST has better robustness to changes in viewpoint, so it achieves a higher AP value. Moreover, our segment token module is easy to implement and has the potential to be fused with other backbones as long as the attention mechanism is available.

4.4 Ablation Study

1) Effect of the Segment Token Module: We conduct experiments on the effect of our segment token module. We remove the segment token module and train the class token like the vanilla vision transformer does. As shown in Table 3, we list the values of R@5, and R@1% for further reference. In the drone→satellite task, the R@1 value improves by 15.15% and the AP value improves by 13.17%. In the satellite→drone task, the R@1 value improves by 5.00% and the AP value improves by 13.96%. Compared to the vanilla vision transformer, our model considers both global and local information. Local information helps extract global information, which in turn guides the extraction of local information. This positive feedback process effectively promotes the model's understanding of aerial images. In addition, the R@1 value is relatively little boosted in the satellite→drone task. When adopting the vanilla vision transformer, we find that the R@1 value in the satellite→drone task is significantly higher than the other three metric values (R@1 and AP). This is because the satellite→drone task is a one-to-many match, where each satellite-view query image has 54 corresponding drone-view images. This means it has a higher hit probability for R@1.

2) Effect of the Number of Segment Tokens: We conduct experiments to figure out how many groups we should segment so that our model has the best performance. As shown in Figure 4, the horizontal colored dotted lines stand for the vanilla method's R@1 and AP values in both tasks. When $N_{group} = 3$, all the metrics have the highest values. We believe that at this point there is

Ablation	R@1	R@5	R@1%	AP
Drone→Satellite				
Vanilla	71.14	88.55	99.95	75.03
Ours	86.29	94.72	99.99	88.20
Satellite→Drone				
Vanilla	84.59	90.44	91.16	71.10
Ours	89.59	92.58	93.30	85.06

Table 3. Ablation study on the segment token module.

better discrimination between groups and better similarity within groups, with less variance in the attention values of the patches within groups. In addition, we visualize the attention maps generated in AST, as shown in Figure 6. When $N_{group} = 4$ or $N_{group} = 6$, most of the attention is wrongly attracted by the surrounding buildings, resulting in a significant drop in performance. On the other hand, we concatenate the output segment tokens in order of importance (i.e., attention values), which achieves spatial alignment between cross-view images to some extent.

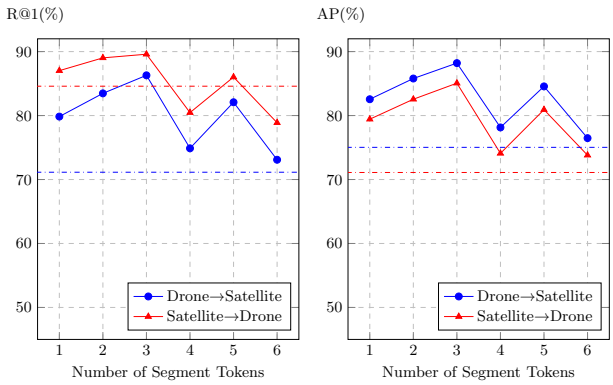

Fig. 4. Ablation study on the effect of number of segment tokens. The horizontal colored dotted lines stand for the vanilla method's R@1 and AP values in both tasks.

3) Effect of Grouping Strategy: Specifically, we conduct experiments on the effect of the number of tokens in each group. We consider three grouping strategies: the decreasing strategy, the averaging strategy, and the increasing strategy. They indicate the trend in the number of patches in the group as the attention value decreases. As shown in Table 4, we evaluate the different grouping strategies in detail. In the drone→satellite task, the increasing strategy achieves the best performance. In the satellite→drone task, the averaging strategy having the highest R@1 value, 0.16% higher than the increasing strategy, while the

increasing strategy has the highest AP value, 0.74% higher than the averaging strategy. Regions with high recognition in aerial images generally have a small area percentage, and attention values are mainly concentrated in a few patches. Therefore, when patches are sorted by attention value, their values drop rapidly and then level off. The increasing strategy can make the attention values of patches in the same group relatively close to each other, resulting in better similarity within groups. It is worth noting that the increasing strategy with fixed proportions is applicable in most cases, but not all.

Ablation	R@1	R@5	R@1%	AP
Drone→Satellite				
Decreasing	83.51	93.70	99.95	85.80
Averaging	85.19	94.32	99.98	87.27
Increasing	86.29	94.72	99.99	88.20
Satellite→Drone				
Decreasing	88.30	92.01	92.30	83.20
Averaging	89.73	93.44	93.58	84.32
Increasing	89.59	92.58	93.30	85.06

Table 4. Ablation study on grouping strategy.

4) Effect of Fusion Strategy: We consider two strategies for fusing patch tokens to generate segment tokens. In addition to fusing patch tokens according to the weights generated by the attention map, we further try to average the patch tokens directly. As shown in Table 5, the weighted mean strategy is our default fusion strategy. In the drone→satellite task, the weighted mean strategy performs better than the mean strategy. In the satellite→drone task, the mean strategy has a 0.28% higher R@1 value, while the weighted mean strategy has a 0.86% higher AP value. We believe that the weighted mean strategy has better robustness to changes in viewpoint and thus achieves higher AP values in both tasks. The mean strategy, on the other hand, might be more applicable at certain shooting angles and distances and thus achieve a higher R@1 value in the satellite→drone task. And the data show that this advantage of the mean strategy is also weak.

5) Effect of Sample Size: In the University-1652 training dataset, each target has 54 drone-view images but only 1 satellite-view image because drones have a variety of viewpoints while satellites usually have a vertical view. This is realistic but not good for the training of neural networks. Therefore, we sample N_{sample} image pairs for each class, each image pair containing one drone-view image and one satellite-view image. There are two advantages: 1) We have the same number of drone-view images and satellite-view images after image augmentation, which can alleviate the problem of data imbalance to some extent. 2) The paired format is more beneficial to the metric learning of cross-view images. As shown in Figure 5, when $N_{sample} = 8$, the model achieves the best perfor-

Ablation	R@1	R@5	R@1%	AP
Drone→Satellite				
Mean	84.91	94.13	99.98	86.99
Weighted Mean	86.29	94.72	99.99	88.20
Satellite→Drone				
Mean	89.87	92.58	93.01	84.20
Weighted Mean	89.59	92.58	93.30	85.06

Table 5. Ablation study on fusion strategy.

mance. We believe that data overfitting may occur when N_{sample} is too large. On the other hand, image augmentation cannot fundamentally solve the problem that the number of satellite-view images is much less than that of drone-view images. As for metric learning, an appropriate value of N_{sample} can increase the probability of images of the same class appearing in a mini-batch, which is beneficial for the model to mine the commonality among different perspectives of images. But it is bad for the model to learn inter-class differences when the probability is too high.

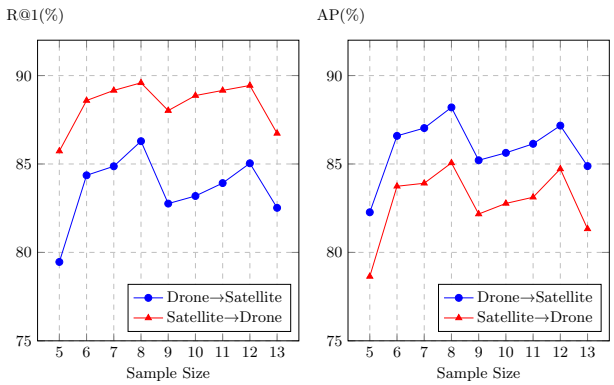

Fig. 5. Ablation study on the effect of sample size.

6) Effect of Triplet Loss: We obtain the embedding features before the linear projection layer, and these features can be used directly for image matching without optimization using triplet loss. We conduct several experiments to verify the effect of the additional triplet loss. As shown in Table 6, we try to remove the triplet loss and also try to adopt other loss functions. In both tasks, the values of each metric improve to different degrees after adopting the additional loss function. These loss functions are widely used in metric learning tasks. And they can be used to adjust the distance between the feature vectors of samples

in the shared feature space, narrowing the distance between samples of the same class and expanding the distance between samples of different classes. This is beneficial for image matching.

Ablation	R@1	R@5	R@1%	AP
Drone→Satellite				
CE	84.90	94.28	99.98	87.04
CE + Contrast	85.45	94.54	99.98	87.52
CE + MT	85.64	94.90	99.97	87.73
CE + WST	86.29	94.72	99.99	88.20
Satellite→Drone				
CE	89.02	92.44	92.87	84.40
CE + Contrast	90.44	92.72	92.87	84.81
CE + MT	89.16	93.30	94.01	84.96
CE + WST	89.59	92.58	93.30	85.06

Table 6. Ablation study on loss function. CE means Cross-Entropy loss, Contrast means Contrast loss, MT means Max-margin Triplet loss, WST means Weighted Soft-margin Triplet loss.

7) Effect of Changes in Viewpoint: In the drone-view images, the regions with high recognition vary with the viewpoints. We divide the query drone-view images into 3 groups based on the distance to explore the effect of the drone distance to the target. In addition, we divide the images into 18 groups according to the shooting angle to verify the effect of view angle. As shown in Table 7, we obtain the best performance when the distance is middle, followed by the short distance, and finally the long distance, and they have close performances. As shown in Table 8, all the view angles also have close performance. Our grouping strategy and fusion strategy can guarantee discriminative patches' role in segment tokens when the area of highly discriminative regions changes. Experimental results indicate that our model has good robustness to changes in viewpoint.

Drone→Satellite				
Distance	R@1	R@5	R@1%	AP
ALL	86.29	94.72	99.99	88.20
Short	86.16	94.27	99.48	88.01
Middle	87.30	95.14	99.66	89.07
Long	85.40	94.76	99.59	87.51

Table 7. Ablation study on the effect of the drone distance to the target.

Drone→Satellite					
Angle	R@1	AP	Angle	R@1	AP
0°	87.69	89.36	180°	85.21	87.27
20°	87.26	89.04	200°	84.40	86.61
40°	87.35	89.11	220°	84.83	86.93
60°	87.26	89.19	240°	85.02	86.99
80°	87.97	89.64	260°	85.92	87.74
100°	87.02	88.88	280°	86.45	88.17
120°	86.21	88.24	300°	86.16	88.13
140°	86.07	88.05	320°	86.12	88.05
160°	85.54	87.60	340°	86.69	88.56

Table 8. Ablation study on the effect of view angle.

8) Effect of Sharing Weights: As we introduced before, our transformers of two branches share the same weights because both satellite-view and drone-view images are captured from an aerial view and have some similar patterns. Also, we test the model, which does not share weights during the training, as shown in Table 9. The R@1 value and the AP value drop rapidly when the weights are not shared. We believe that there are two main reasons: 1) The lack of satellite-view images. The single branch is liable to overfit since there is only one satellite-view image per location. 2) A decrease in the number of input images. When weights are not shared, the number of input images per branch is reduced by half. However, training a transformer requires sufficient data to achieve satisfactory performance. By sharing weights, more drone-view images can be input into the transformer to adjust the model. This helps address the above two issues and achieve better performance.

Ablation	R@1	R@5	R@1%	AP
Drone→Satellite				
W/o Sharing	23.27	47.70	98.03	29.20
Sharing	86.29	94.72	99.99	88.20
Satellite→Drone				
W/o Sharing	26.53	35.38	36.80	23.08
Sharing	89.59	92.58	93.30	85.06

Table 9. Ablation study on sharing weights.

4.5 Visualization

In Figure 6, we present visualization of attention maps. Comparing the attention maps generated by the two models introduced in our first ablation experiment,

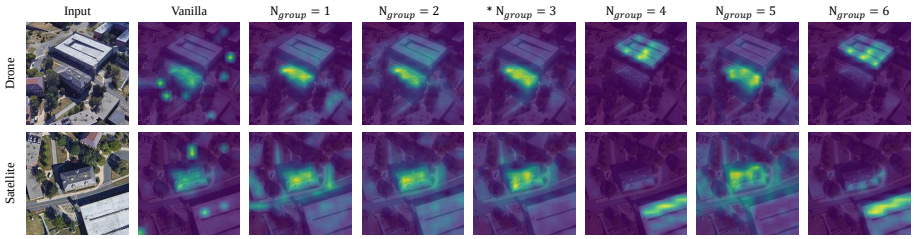

Fig. 6. Visualization of attention maps. We show attention maps generated by vanilla vision transformer and our AST using different numbers of segment tokens. Highlights represent the distribution of attention in the model.

we observe that while the vanilla model primarily focuses on the target building, it also exhibits unexpected interference spots around it. This can negatively impact cross-view image matching. In contrast, our model ($N_{group} = 3$) correctly focuses on the top of the target building before expanding its attention to surrounding buildings, roads, and finally indistinguishable trees. This allows for accurate matching between drone and satellite-view images. These results demonstrate that our proposed segmentation token module effectively enhances the vision transformer's ability to correctly interpret aerial images and improves drone-based cross-view image matching performance.

In Figure 7 and Figure 8, we present cross-view matching results in both drone-view target localization and drone navigation tasks. In the drone-view target localization task, we randomly select 3 drone-view images from and retrieve the top 5 satellite-view matches. For each query image, only the first-ranked satellite-view image corresponds to it. Notably, our model successfully distinguishes between satellite-view images with different centers despite overlapping areas, which situation appears in the first and third rows. In the drone navigation task, we follow a similar process and retrieve the top 5 drone-view matches. All retrieved drone-view images indicate the same location as their corresponding input satellite-view images. In summary, our method achieves completely accurate results in both tasks shown in the figures.

We further conduct some experiments to test the generalization ability of our method to real-world case. As shown in Figure 9, we present two retrieval results: Real Drone → Synthetic Drone and Real Drone → Satellite. The former evaluates how well the synthetic data mimics the real drone camera images. We display the top-5 most similar images in the test set retrieved by our model. The results suggest that the synthetic drone-view images have similar visual features to the real drone-view query. The latter tests the generalization performance of our model on the real drone-view data. The result demonstrates that our model can also handle the real drone-view images for drone-view target localization. These results demonstrate that our model has good generalization ability to real-world case.

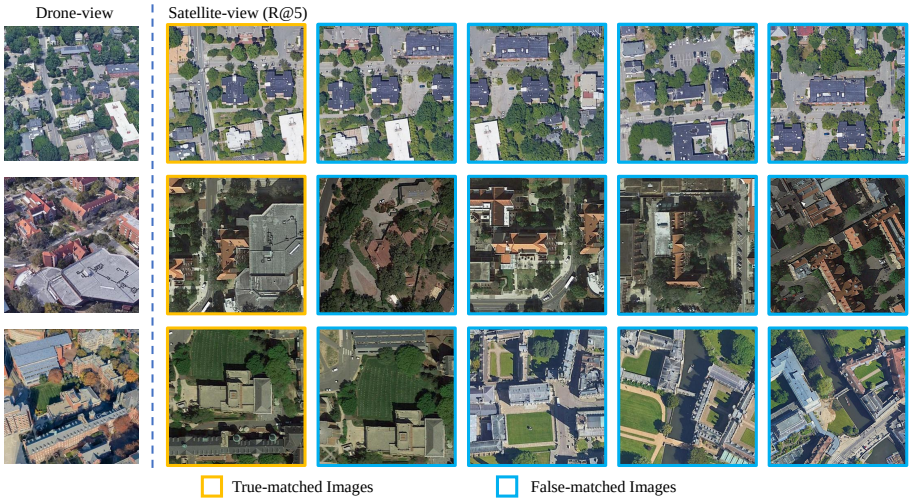

Fig. 7. Visualization of drone-view target localization. Inputting the drone-view images as the query images, we show the top 5 satellite-view images in the ranking of the matching results in the drone→satellite task.

Fig. 8. Visualization of drone navigation. Inputting the satellite-view images as the query images, we show the top 5 drone-view images in the ranking of the matching results in the satellite→drone task.

5 Conclusion

In this paper, we have addressed how to match drone-view images with satellite-view images to tackle the problem of drone-based cross-view geo-localization.

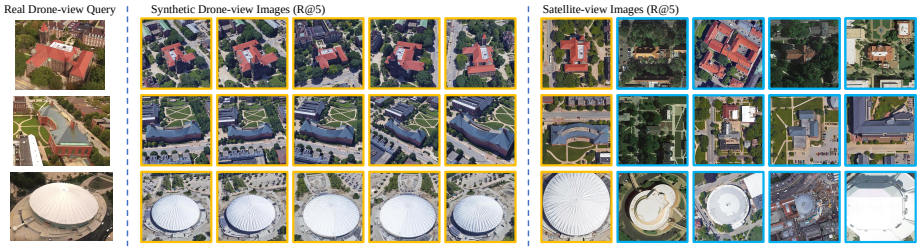

Fig. 9. Qualitative image search results using real drone-view query. In the left column, we show the real drone-view images used for the query. In the middle column, we show the top-5 most similar images in the test set retrieved by our model. In the right column, we show the retrieval results for drone-view target localization.

To overcome the challenges posed by the variability of aerial views, an effective Attention-guided Segment Transformer (AST) structure has been proposed: we have introduced a novel segmentation strategy that is adaptive and non-uniform, allowing it to effectively handle the huge variations between aerial views by segmenting regions with corresponding relationships even after significant changes in viewpoint; furthermore, we have designed a new segment token module to generates segment tokens that are concatenated with the original class token to supplement local information. In contrast to CNN-based methods that are inclined to extract more fine-grained features but underestimate neighboring patches, AST takes full advantage of the self-attention mechanism to establish global context correlations; and the newly introduced segment token module enables AST to effectively extract local features as well, a capability not present in the vanilla vision transformer. Notably, our method has demonstrated good robustness to variations in viewpoint, even when there are overlapping regions. Our proposed AST has achieved competitive performance in both drone-view target localization and drone navigation tasks in the University-1652 benchmark.

Nonetheless, there remains potential for further improvement. One limitation of our approach is that we segment patch tokens in a fixed proportion, which may not be suitable in all situations. In future work, we plan to explore adaptive proportions or even an adaptive number of groups to improve model performance. Besides, it might be a good idea to improve our segmentation strategy by using advanced segmentation models, such as SAM[18].

References

1. Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al.: Xcit: Cross-covariance image transformers. In: Adv. Neural Inf. Process. Syst. vol. 34, pp. 20014–20027 (2021)
2. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. In: Adv. Neural Inf. Process. Syst. vol. 33, pp. 1877–1901 (2020)

3. Bui, D.V., Kubo, M., Sato, H.: A part-aware attention neural network for cross-view geo-localization between uav and satellite. J. Rob., Networking Artif. Life **9**(3), 275–284 (2022)
4. Cao, H., Wang, Y., Chen, J., Jiang, D., Zhang, X., Tian, Q., Wang, M.: Swin-unet: Unet-like pure transformer for medical image segmentation. arXiv:2105.05537 (2021), https://arxiv.org/abs/2105.05537
5. Chen, C.F.R., Fan, Q., Panda, R.: Crossvit: Cross-attention multi-scale vision transformer for image classification. In: Proc. IEEE/CVF Int. Conf. Comput. Vis. pp. 357–366 (2021)
6. Chopra, S., Hadsell, R., LeCun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. vol. 1, pp. 539–546. IEEE (2005)
7. Chu, X., Tian, Z., Wang, Y., Zhang, B., Ren, H., Wei, X., Xia, H., Shen, C.: Twins: Revisiting the design of spatial attention in vision transformers. In: Adv. Neural Inf. Process. Syst. vol. 34, pp. 9355–9366 (2021)
8. Dai, M., Hu, J., Zhuang, J., Zheng, E.: A transformer-based feature segmentation and region alignment method for uav-view geo-localization. IEEE Trans. Circuits Syst. Video Technol. **32**(7), 4376–4389 (2021)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv:1810.04805 (2018), https://arxiv.org/abs/1810.04805
10. Ding, L., Zhou, J., Meng, L., Long, Z.: A practical cross-view image matching method between uav and satellite for uav-based geo-localization. Remote Sens. **13**(1), 47 (2020)
11. Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., Chen, D., Guo, B.: Cswin transformer: A general vision transformer backbone with cross-shaped windows. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 12124–12134 (2022)
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv:2010.11929 (2020), https://arxiv.org/abs/2010.11929
13. Han, K., Xiao, A., Wu, E., Guo, J., Xu, C., Wang, Y.: Transformer in transformer. In: Adv. Neural Inf. Process. Syst. vol. 34, pp. 15908–15919 (2021)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
15. Heo, B., Yun, S., Han, D., Chun, S., Choe, J., Oh, S.J.: Rethinking spatial dimensions of vision transformers. In: Proc. IEEE/CVF Int. Conf. Comput. Vis. pp. 11936–11945 (2021)
16. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
17. Hu, S., Feng, M., Nguyen, R.M., Lee, G.H.: Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 7258–7267 (2018)
18. Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.Y., et al.: Segment anything. arXiv:2304.02643 (2023), https://arxiv.org/abs/2304.02643
19. Lin, J., Zheng, Z., Zhong, Z., Luo, Z., Li, S., Yang, Y., Sebe, N.: Joint representation learning and keypoint detection for cross-view geo-localization. IEEE Trans. Image Process. **31**, 3780–3792 (2022)

20. Lin, T.Y., Cui, Y., Belongie, S., Hays, J.: Learning deep representations for ground-to-aerial geolocalization. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 5007–5015 (2015)
21. Liu, L., Li, H.: Lending orientation to neural networks for cross-view geo-localization. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 5624–5633 (2019)
22. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proc. IEEE/CVF Int. Conf. Comput. Vis. pp. 10012–10022 (2021)
23. Lu, Z., Pu, T., Chen, T., Lin, L.: Content-aware hierarchical representation selection for cross-view geo-localization. In: Proc. Asian Conf. Comput. Vis. pp. 4211–4224 (2022)
24. Luo, W., Li, Y., Urtasun, R., Zemel, R.: Understanding the effective receptive field in deep convolutional neural networks. Adv. Neural Inf. Process. Syst. **29** (2016)
25. Pan, Z., Zhuang, B., Liu, J., He, H., Cai, J.: Scalable vision transformers with hierarchical pooling. In: Proc. IEEE/CVF Int. Conf. Comput. Vis. pp. 377–386 (2021)
26. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Adv. Neural Inf. Process. Syst. vol. 32 (2019)
27. Radenović, F., Tolias, G., Chum, O.: Fine-tuning cnn image retrieval with no human annotation. IEEE Trans. Pattern Anal. Mach. Intell. **41**(7), 1655–1668 (2018)
28. Shi, Y., Liu, L., Yu, X., Li, H.: Spatial-aware feature aggregation for image based cross-view geo-localization. In: Adv. Neural Inf. Process. Syst. vol. 32 (2019)
29. Tian, X., Shao, J., Ouyang, D., Shen, H.T.: Uav-satellite view synthesis for cross-view geo-localization. IEEE Trans. Circuits Syst. Video Technol. **32**(7), 4804–4815 (2021)
30. Toker, A., Zhou, Q., Maximov, M., Leal-Taixé, L.: Coming down to earth: Satellite-to-street view synthesis for geo-localization. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 6488–6497 (2021)
31. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Adv. Neural Inf. Process. Syst. vol. 30 (2017)
32. Wang, P., Wang, X., Wang, F., Lin, M., Chang, S., Li, H., Jin, R.: Kvt: k-nn attention for boosting vision transformers. In: Proc. Eur. Conf. Comput. Vis. pp. 285–302. Springer (2022)
33. Wang, T., Zheng, Z., Yan, C., Zhang, J., Sun, Y., Zheng, B., Yang, Y.: Each part matters: Local patterns facilitate cross-view geo-localization. IEEE Trans. Circuits Syst. Video Technol. **32**(2), 867–879 (2021)
34. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: Proc. IEEE/CVF Int. Conf. Comput. Vis. pp. 568–578 (2021)
35. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 17683–17693 (2022)
36. Yang, H., Lu, X., Zhu, Y.: Cross-view geo-localization with layer-to-layer transformer. In: Adv. Neural Inf. Process. Syst. vol. 34, pp. 29009–29020 (2021)
37. Zhai, M., Bessinger, Z., Workman, S., Jacobs, N.: Predicting ground-level scene layout from aerial imagery. In: Proc. IEEE Conf. Comput. Vis. Pattern Recognit. pp. 867–875 (2017)

38. Zheng, Z., Wei, Y., Yang, Y.: University-1652: A multi-view multi-source benchmark for drone-based geo-localization. In: Proc. 28th ACM Int. Conf. Multimedia. pp. 1395–1403 (2020)
39. Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Jiang, Z., Hou, Q., Feng, J.: Deepvit: Towards deeper vision transformer. arXiv:2103.11886 (2021), https://arxiv.org/abs/2103.11886
40. Zhu, S., Shah, M., Chen, C.: Transgeo: Transformer is all you need for cross-view image geo-localization. In: Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. pp. 1162–1171 (2022)