

Finding Point with Image: An End-to-End Benchmark for Vision-based UAV Localization

Ming Dai, Jiahao Chen, Yusheng Lu, Wenlong Hao, Enhui Zheng

arXiv:2208.06561v1 [cs.CV] 13 Aug 2022

Abstract—In the past, image retrieval was the mainstream solution for cross-view geolocation and UAV visual localization tasks. In a nutshell, the way of image retrieval is to obtain the final required information, such as GPS, through a transitional perspective. However, the way of image retrieval is not completely end-to-end. And there are some redundant operations such as the need to prepare the feature library in advance, and the sampling interval problem of the gallery construction, which make it difficult to implement large-scale applications. In this article we propose an end-to-end positioning scheme, Finding Point with Image (FPI), which aims to directly find the corresponding location in the image of source B (satellite-view) through the image of source A (drone-view). To verify the feasibility of our framework, we construct a new dataset (UL14), which is designed to solve the UAV visual self-localization task. At the same time, we also build a transformer-based baseline to achieve end-to-end training. In addition, the previous evaluation methods are no longer applicable under the framework of FPI. Thus, Metre-level Accuracy (MA) and Relative Distance Score (RDS) are proposed to evaluate the accuracy of UAV localization. At the same time, we preliminarily compare FPI and image retrieval method, and the structure of FPI achieves better performance in both speed and efficiency. In particular, the task of FPI remains great challenges due to the large differences between different views and the drastic spatial scale transformation.

Index Terms—unmanned aerial vehicle, geo-localization, benchmark, transformer.

I. INTRODUCTION

UAV generally uses GPS signals for self-positioning, which naturally relies on satellites. However, the GPS signal is not present in all scenarios. Drones tend to fly in environments with weak or no GPS signals. We collectively call this situation a denial environment [1][2][3]. Vision-based UAV positioning is mainly to solve the problem of UAV self-localization in the denial environment. The concept of the denial environment is relatively common in the military. Signal jamming is a commonly used method in military operations. Position information is intuitively important information for an aircraft. However, once the position information is lost, the aircraft will also be disabled or even out of control. Therefore, the additional assisted positioning capability can improve the reliability and stability of the aircraft [4]. The pure visual localization task in this paper is a self-localization task for UAVs that only relies on cameras, which can be used as an auxiliary positioning method when the GPS signal cannot be obtained.

Feature point matching [5][6] is a common solution for image matching. Because its hand-designed feature description method has certain characteristics of anti-scale, anti-rotation, and anti-noise, it is widely used in slam and image stitching.

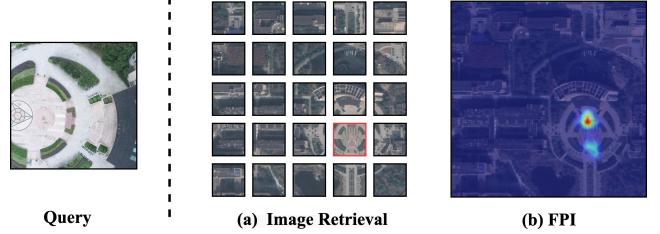


Fig. 1. Query refers to the UAV view image. (a) The basic process of image retrieval to solve the UAV localization. (b) Heatmap for solving the localization problem using the FPI method proposed in this paper.

However, this method of pairing of corner points can only utilize a small number of salient features, and basically does not work when the image quality varies greatly between viewing angles. Therefore, in some challenging tasks, deep learning-based schemes can mine more associations between features and work better.

DenseUAV [7] draws on the framework of Cross-View Geo-Localization to realize the visual positioning of UAVs. Thus adopted an image retrieval approach, using drone images as a query to retrieve the most similar satellite images in the entire satellite image database. Then, GPS information is obtained to realize the positioning of the UAV. Although this method of image retrieval is very direct and has a significant effect on positioning accuracy, it is very costly on the application side. First, we need to crop the satellite image, as shown in Fig. 1 (a). We need to cut the satellite image into $N \times N$ blocks, that is, N^2 copies, and save them to disk. The pixels of each image is $S \times S$. Then we need to extract features from these cropped images through the forward propagation of the model (the single calculation amount is M , output dimension is D) for storage. That is to say before we start matching, we need to perform $N^2 \times M$ calculations (without considering the calculation of cropping satellite maps), and $(N^2 \times S^2 \times 3 + N^2 \times D)$ storage space. This expression may not be intuitive enough. If N is 100 (assuming that 5 meters are sampled once, $N = 100$ means an area of $500m \times 500m$). It is assumed that the forward propagation time of the model for a single image is 0.01 seconds. Then preprocessing a $500m \times 500m$ area takes 100s and requires nearly 240MB of disk space ($S = 256$, $D = 512$). In addition, although we have completed the data preprocessing of a large area after we iterate the model, the previous features need to be overturned, which undoubtedly increases the difficulty of the application.

Based on the above thinking, we re-examined the core problem of UAV positioning. We believe that the most direct

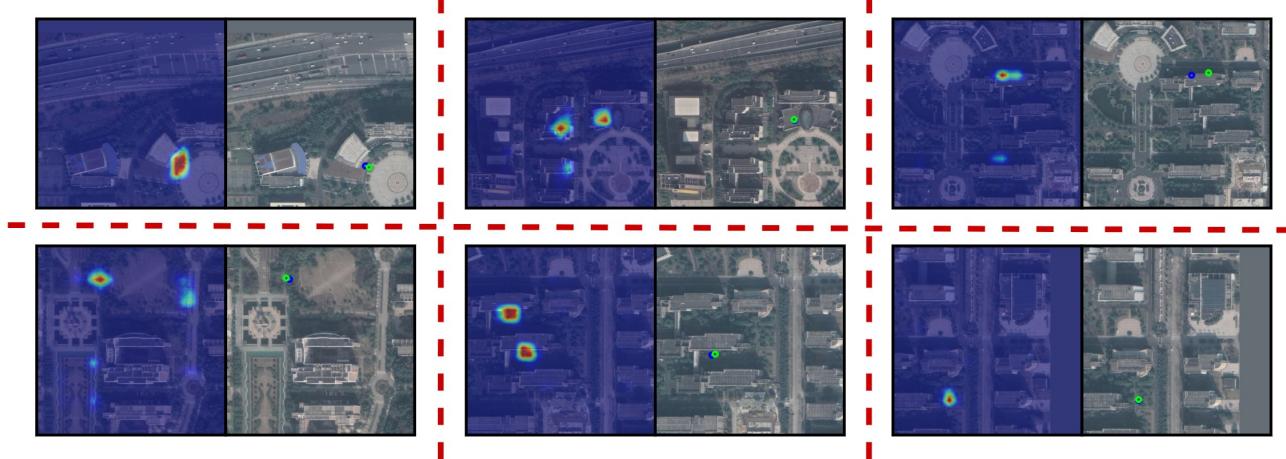


Fig. 2. Six pairs of visualization results based on the FPI method. In each pair, the heatmap of the model output is shown on the left, and the results of groundtruth and prediction are shown on the right (blue is the position predicted by the model, green is the groundtruth).

way to locate is to find the point. Therefore, we propose a positioning-based benchmark called FPI. For the convenience of description, we uniformly call the input image to be positioned (corresponding to the drone image in this paper) as a query, and the image to be retrieved (corresponding to the satellite image) as a search map. Localization tasks including UAV localization and cross-view geo-localization are essential to finding the corresponding location in the search map. From the perspective of input and output, FPI directly inputs the query and search map into the model, and finally outputs a heatmap representing the predicted location distribution of the query in the search map, as shown in Fig. 1 (b). To further analyze the advantages and disadvantages of the image retrieval method and the FPI method, we analyze them from the following aspects. First of all, the FPI method does not require a lot of preparatory data and feature extraction operation in advance, and the only storage is a search map. Assuming that the resolution of the search map is 1000×1000 , the storage space is 0.4MB, which is only 1/600 of the image retrieval scheme. Assuming that the FPI is the same as the model delay of image retrieval, the calculation amount of a single prediction is M (the time consumption is 0.01s), which is only 1/10000 of the image retrieval scheme. Data preparation is very cumbersome, and FPI avoids the shortcomings of previous image retrieval in data preprocessing through an end-to-end approach, which we believe provides the possibility for large-scale landing applications of UAV visual localization.

How is FPI implemented? The main idea is inspired by SiamFC [8]. In the Single Object Tracking (SOT) domain, the input to the model consists of two parts. Part of the template, which is like a query in FPI, is a relatively small image. The other part is the image that contains the template, which is like the search map in FPI. The difference is that the input of SOT is generally of the same sources. The template is generally cropped from the camera image. However, the input of FPI comes from different sources, the query comes from drones, and the search map comes from satellites. After receiving the input, the respective features are extracted through the back-

bone. Then the features of the query and search map are fused through a grouped convolution, and finally, a heat distribution map is an output. At last, the heatmap is supervised by the relative position of the query in the search map. In Fig. 2, we show six pairs of visualization structures based on the FPI architecture.

In conclusion, FPI is a new paradigm based on the positioning of one image in another. Compared with the image retrieval scheme, in summary, it has the following advantages. 1) There is no time-consuming preprocessing, and there is no need to prepare an image feature library in advance, just get the real-time captured UAV view picture and a satellite image. 2) There is no natural positioning error caused by the sampling interval, and the positioning accuracy depends entirely on the performance of the model. 3) There is no need to calculate the cosine similarity with the feature vectors in the gallery, and the inference speed only depends on the inference time of the model. 4) It has good iterability. After the model is iteratively tuned, there is no need to refresh the library, and the old model can be replaced directly.

We summarize our main contributions as follows:

- 1) This paper does not inherit the scheme of image retrieval, but solves multi-source geolocation problem in a more direct way of Finding Points with Image. FPI does not require additional preparations, but just need to store a large satellite image. The architecture of FPI greatly reduces the amount of computation and storage space, which is conducive to the realization of landing applications.
- 2) Based on the idea of FPI, we constructed a brand-new benchmark, which includes the UL14 dataset, a strong baseline based on Transformer, and intuitive evaluation indicators Metre-level Accuracy(MA) and Relative Distance Score(RDS).
- 3) To the best of our knowledge, this is the first time to solve the problem of UAV localization by using the method of direct point finding on the map. Based on the benchmark proposed in this paper on the metre-

level accuracy, the average accuracy within 20 meters can reach 57.7%. And FPI outperforms image retrieval schemes both in speed and performance.

II. RELATED WORK

A. Geo-Localization Dataset Review

The geo-localization is a task of image retrieval between different perspectives. Initially, the geo-localization task was proposed to solve the ground-to-aerial matching problem. Lin [9] proposed to leverage the public sources to build image pairs, which consists of 78k image pairs from ground and aerial views. Subsequently, Tian [10] collected image pairs from the town view for the task of geo-localization, which argued that architecture can provide more notable features for geo-localization. Vo and Hays [11] aimed to determine the location and orientation of a ground-level query image by matching a reference database of satellite images, and they collected a dataset with one million pairs of street view and overhead images sampled from eleven U.S. cities. Subsequently, CVUSA [12] constructed image pairs from ground-based panoramic images and satellite images. Based on CVUSA, CVACT [13] added spatial factors, i.e., orientation maps, to reconstruct and expand the CVUSA dataset. The first two mentioned datasets [9], [10] targeted the buildings in images, while the latter three mentioned [11], [12], [13] targeted users with photographic devices. VIGOR [14] changes the previous center-to-center image matching and redefines the problem with a more realistic assumption that the query image can be arbitrary in the area of interest and the reference images are captured before the queries emerge. Recently, University-1652 [15] introduced drone view into Cross-View Geo-localization and proposed two drone-based subtasks: drone-view target localization and drone navigation, and regarded them as image retrieval tasks. Inspired by University-1652, DenseUAV [7] achieved high precision localization of UAVs by dense sampling, which is the first time to solve the UAV localization problem by image retrieval. Table I lists the amount of training data, sampling platforms, data distribution, localization targets, and the evaluation metrics for all datasets mentioned above.

B. Deeply-Learned Geo-Localization

Due to a large number of potential applications, cross-view geo-localization has gotten increased attention in recent years. UAV Localization is a task that expands from cross-view geo-localization. There are also many similarities in terms of methods. Some pioneering methods [16], [17], [18], [19], concentrate on extracting hand-crafted features. Inspired by the extraordinary performance of deep convolutional neural networks in tasks of vision, researchers have turned to the deeply-learned feature. Workman et al. [20] were among the first to try to extract features for the cross-view geo-localization challenge using a pre-trained CNN. They showed that characteristics from CNN's high-level layer contain semantic information about location. Workman et al. [21] took it a step further by fine-tuning the pre-trained network by lowering the feature distance between pairs of ground-level and aerial images, resulting in improved performance. Lin et

al. [9] used a modified Siamese Network [22], which used the contrastive loss [23], [24], to optimise network parameters. The NetVLAD [25] was plugged into a Siamese-like architecture by Zhai et al. [26], making image descriptors robust against massive perspective changes. Shi et al. [27] applied spatial layout information to compensate for the shortcomings of global aggregation step in feature extraction. Furthermore, Shi et al. [28] employed domain alignment and a spatial attention mechanism to increase cross-view geo-localization performance. DSM [29] also took into account a limited field of view and employed a dynamic similarity matching module to align the orientation of cross-view images. PLCD [30] took advantage of drone-view information as a bridge between ground-view and satellite-view domains. LPN [31] proposed the square-ring partition strategy to allow the network to pay attention to more fine-grained information at the edge and achieved a huge improvement. DBLP [32] proposed an end-to-end cross-view matching method that integrates cross-view synthesis module and geo-localization module, which fully considered the spatial correspondence of UAV-satellite views and the surrounding area information. GeoNet [33] learned powerful intermediate feature maps and allowed the stable propagation of gradients in deep CNNs and utilizes the capsule network to encapsulate the intermediate feature maps into several capsules. FSRA [34] introduced a simple and efficient transformer-based structure to enhance the ability of the model to understand contextual information as well as to understand the distribution of instances.

C. Transformer in Vision

With Transformer leading the way in Natural Language Processing (NLP) [35], [36], a lot of work has been done to apply it to the field of computer vision. Before the Vision Transformer took off, the CNN approach was the most popular method in the field of computer vision. Most of the structure of CNN adopts a hierarchical structure. Taking Resnet as an example, the model is generally divided into 4 stages, and the scale of the feature map is reduced one by one. Hierarchical structure can facilitate the operation of feature fusion such as FPN [37]. Transformer-based models are typical of ViT [38], and self-attention is not friendly to large scales, so models such as ViT reduce the scale of the input image at the beginning. Subsequently, multiple self-attention blocks are used to construct patch-level feature relationships. DeiT [39] adds a Distillation token based on ViT, which further improves the performance of Vision Transformer through knowledge distillation. However, the way that ViT compresses the scale at the beginning is not friendly to low-level tasks such as object detection, semantic segmentation, and GAN. So there is the Transformer structure of the PVT [40] hierarchical structure. Then Swin-Transformer [41] made a big hit in various fields of vision with its window-size self-attention mechanism and hierarchical structure. The benchmark proposed in this paper is a multi-source and multi-input task, and the distribution of data from different sources varies greatly. Therefore, the requirements for the ability of model feature mining are relatively high. In section IV-E, we formulate experiments to

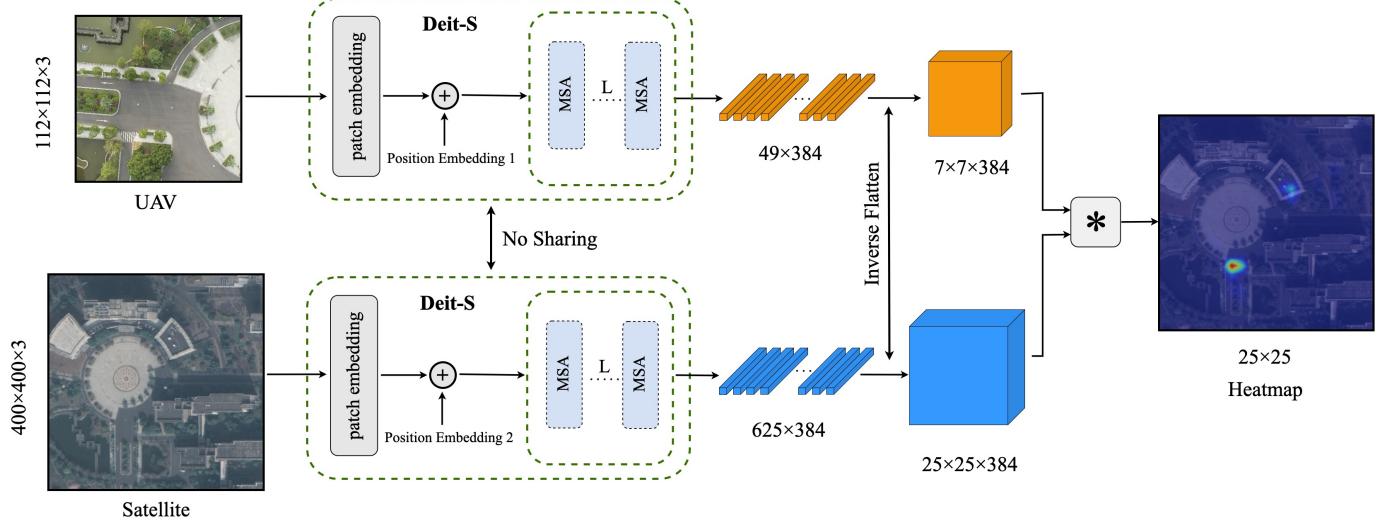


Fig. 3. Schematic diagram of the model structure of FPI. *MSA* refers to multi-head self-attention

explore the influence of backbone on the benchmark proposed in this paper.

III. METHODOLOGY

A. The Framework of FPI

We performed some attempts based on different backbones in order to develop a solid and robust benchmark model. After conducting tests, we discovered that the Transformer-based model outperforms the CNN-based model by a large margin, which we think it is because CNN lacks rotation invariance and scale invariance. FPI is a model built on the Transformer architecture, and the model structure is shown in Fig. 3. Patch embedding contains a convolution with the kernel size of 16, the stride of 16, and the output channel of 384 to downscale the length and width of the input image. The input image is then flattened into a number of patches, and a learnable position embedding is applied following patch embedding. After the L-layers multi-head self-attention, where L is 12 in Deit-S, The query is converted to 49×384 dimension features, and the search map is converted to 625×384 features. Inverse Flatten is the inverse operation of Flatten, which is the process of converting vectors into a map. The $*$ component of Fig. 3 is similar to the correlation operation in SiamFC [8], which applied group convolution to compute the correlation between the feature map of query and search map. The difference is that we include the padding operation in this group convolution in order to maintain the scale of the satellite image. We have also attempted Cross-Attention in TransT [42] and Concat-Attention in SwinTrack [43] to replace the group convolution operation, but does not get a substantial boost. In the test phase, we will restore the heatmap to the original satellite image scale through bilinear interpolation, and obtain the maximum thermal position according to the 3×3 hanning window.

B. Dataset Description

UL14 is reconstructed from the DenseUAV dataset. The difference is that DenseUAV is based on the task of image

retrieval, and UL14 is formulated for the FPI benchmark. The same is that they are composed of images from drone perspective and satellite perspective. The number of training sets and test sets is detailed in Table II.

1) *Train Set*: The training set data consists of images collected by ten universities. The drone image is composed of three different flight heights, which are 80, 90, and 100 meters. The satellite view uses level 20 imagery. To improve the operability of the training set. The resolution of drone images in the training set are uniformly saved as (512, 512), while the satellite images are saved as (1280, 1280). The reason we keep larger satellite images is to provide more flexibility for later data augmentation.

2) *Test Set*: The test set is constructed from the data sampled from four universities, and there is no intersection with the training set sampling area. The corresponding positions in the satellite images are found based on the GPS position information of the UAV images. The spatial scale of the satellite images is determined by setting the pixels in the interval of 700-1800 (where 700 pixels correspond to a spatial dimension of 180 meters in length and width, and 1800 pixels correspond to 463 meters) at an interval of 100 pixels, which means that one UAV image in the test set will generate 12 satellite images of different scales. The position of the UAV (query) in the satellite image (search map) is similar to that in the training data, except that the coverage rate is set to 0.95 in order to increase the challenge of the evaluation set, which means that some of the query positions will be distributed on the edge of the search map. In UL14, Table II shows the number of images in the training and testing sets. It is worth noting that we will execute more random cropping of the satellite images throughout the training phase (Random Crop is explained in detail in Section IV-D), enhancing the expansion of data.

C. Data Enhancement

We expect the training images to be diverse, so we propose a data augmentation method that can make the search map have

TABLE I
GEO-LOCALIZATION DATASET INFORMATION SUMMARY TABLE. AMONG THEM, THE D, S, G OF THE #PLATFORM LINE REPRESENT DRONE, SATELLITE, GROUND RESPECTIVELY. THE LINE #TARGET SPECIFIES THE POSITION OBJECT

Datasets	UL14	DenseUAV[7]	University-1652[15]	VIGOR[14]	CVUSA[12]	CVACT[13]
#Training	6.8k×2	2.3k × 6	701×71.64	91k+53k	35.5k×2	35.5k×2
#Platform	D,S	D,S	D,G,S	D,S	G,S	G,S
#Imgs/Location	1 + 1	3 + 3	54 + 16.64 + 1	/	1 + 1	1 + 1
#Target	UAV	UAV	Building	User	User	User
#Method	FPI	Image Retrieval	Image Retrieval	Image Retrieval + Regression	Image Retrieval	Image Retrieval
#Evaluation	RDS & MA	SDM	Recall@K & AP	MA	Recall@K	Recall@K

TABLE II
STATISTICAL TABLE FOR THE UL14 DATASET. WHERE #university REFERS TO THE NUMBER OF UNIVERSITIES INCLUDED

#split	#image uav	#image satellite	#university
train	6768	6768	10
test	2331	27972	4

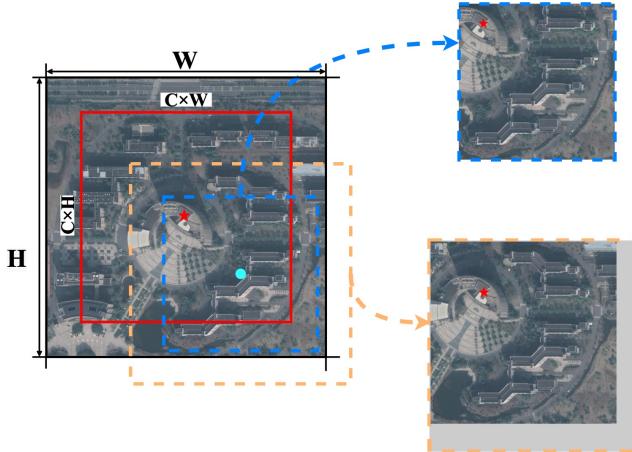


Fig. 4. The pentagram represents the center of the satellite image, as well as the location of the drone. The light blue circle is the randomly created center point. The image scales are randomly selected between 512 and 1000, and the illustration generates an enhanced image of two random scales, both containing the drone position (the excess is filled with the mean of the image)

the properties of random scale and random target distribution at the same time. In implementation, we intentionally make some redundancy in the scale of satellite images while creating the dataset, so that we can dynamically crop satellite images with varied scales and offsets during the training phase, diversifying the matching pairs of UAV and satellite images. Fig. 4 depicts the implementation of this data enhancement approach, in which we create two hyperparameters to dynamically crop satellite images: one is the centroid coverage range C (0.75 in default), and the other is the scale range (upper limit 1000 and lower limit 512 in default). The hyperparameter C determines

the distribution interval of the query in the search map. The smaller C is, the closer the query is distributed in the middle of the search map; the larger C is, the larger the interval of query distribution. The hyperparameter $scale$ determines the spatial scale of the search map. The larger the $scale$, the greater the spatial distance occupied by the unit pixel after resize. This data augmentation method is to improve the model's ability to resist scale and offset. Simply, as shown in Fig. 4. R determines the distribution interval of the light blue point (query) (in the red area on the left of Fig. 4, the side length is determined by C); $scale$ determines the scale of the cropped image.

D. Evaluation Indicators

Image-based retrieval schemes mostly use evaluation indicators such as Recall@K and AP, and the evaluation of these indicators requires the concept of positive and negative examples. In contrast, the framework of FPI finds points directly in the graph, and the previous evaluation method no longer applies. Metre-level Accuracy is an intuitive indicator for evaluating position error in space, and Relative Distance Score is an indicator for evaluating positioning accuracy from the model level.

1) *Metre-level Accuracy (MA)*: In order to visually evaluate the effectiveness of the positioning, we have adopted a metre-level accuracy evaluation. the expression of the MA evaluation metric is shown in Equation 1.

$$MA_{<Km} = \frac{\sum_{i=1}^N 1_{SD < Km}}{N} \quad (1)$$

$$1_{SD < Km} = \begin{cases} 1 & SD < Km \\ 0 & SD \geq Km \end{cases} \quad (2)$$

where SD refers to the real spatial distance in meters. K is an adjustable parameter. MA refers to the number of samples with positioning errors within K m as a proportion of the total samples. To understand it in another way, we take the sample with error less than K m as a positive example, and then MA can be represented as the accuracy of positioning error less than K m. The Equation 3 shows the calculation process of SD .

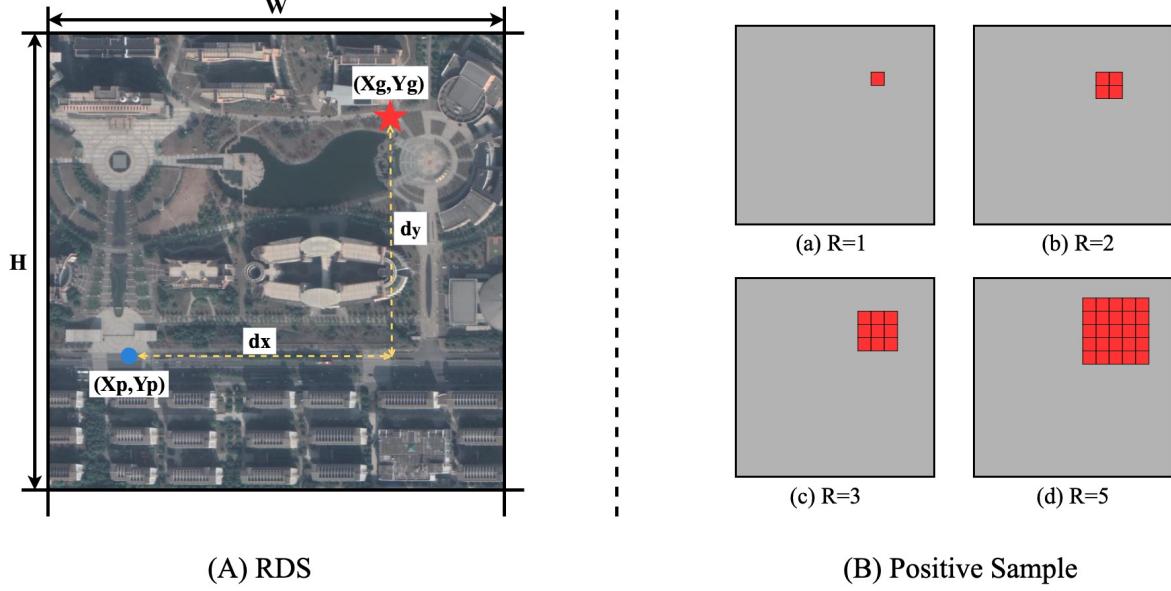


Fig. 5. (A) is the schematic diagram of the relative distance score evaluation metric, with the pentagram indicating groundtruth and the blue circle indicating the position predicted by the model. (B) is the distribution of positive samples under different R settings

$$SD = \sqrt{(\Delta x)^2 + (\Delta y)^2} \quad (3)$$

Δx denotes the meter-level error between the model prediction and the ground truth in the longitude direction, and Δy denotes the meter-level error in the latitude direction.

2) *Relative Distance Score (RDS)*: Although Metre-level Accuracy can express the accuracy of positioning intuitively, it also has some shortcomings. First, the setting of K has different effects on search maps of different scales. For large-scale search maps, after resizing, the spatial distance represented by a unit pixel will also become larger. Therefore, there will be a small deviation in the search map, but a large deviation in the spatial distance. Second, the evaluation method of MA often needs to be expressed by a series of K, which is not convenient for directly express the performance of the model. Finally, when K is greater than a certain threshold, MA has no reference from the model level, which can be considered as wrong positioning. Based on the above discussion, we propose Relative Distance Score. First, the whole calculation process of RDS is introduced. The expression for relative distance (RD) is presented in Equation 4,5,6. Fig. 5(A) shows the schematic diagram, (X_p, Y_p) denoting the predicted coordinates and (X_g, Y_g) denoting the ground-truth coordinates.

$$RD = \sqrt{\frac{(\frac{dx}{w})^2 + (\frac{dy}{h})^2}{2}} \quad (4)$$

$$dx = |X_p - X_g| \quad (5)$$

$$dy = |Y_p - Y_g| \quad (6)$$

We convert the distance into a score, and the expression of the converted RDS is presented in Equation 7, in order to

make the evaluation criteria perform in the same direction as the accuracy and distributed between 0 and 1.

$$RDS = e^{-k \times RD} \quad (7)$$

Where k is the scaling factor, which in this article is set to 10. The following complete RDS expression is generated after substituting RD . RDS measures the performance of the model by the relative distance at the pixel level in the image, which is also evaluated from the perspective of the model.

$$RDS = e^{-k \times \sqrt{\frac{(\frac{dx}{w})^2 + (\frac{dy}{h})^2}{2}}} \quad (8)$$

Why propose an RDS ? What are his strengths? First of all, we believe that RDS , a relative evaluation method at the pixel level, can resist the scale transformation of the search map. The distance in RDS is no longer the real space distance, but the pixel distance in the picture, which is beneficial to evaluate the performance from the model level. Second, the RDS is an overall score value, which is distributed between 0 and 1, and is positively correlated with the positioning accuracy, that is, the higher the RDS , the higher the positioning accuracy. Finally, RDS uses an exponential method to expand the relative distance. When the distance increases, the RDS will return to 0 in an exponential form, which is in line with the expectation that the large distance deviation will be regarded as a wrong positioning.

E. Balance Loss

We attempted to use cross-entropy loss [44] directly in the loss function, as well as FocalLoss[45] to balance positive and negative samples and dig for hard examples, and we even attempted to adopt the idea of Gaussian Focal Loss from CenterNet [46] to assign weights based on distance. After extensive experimentation, finally, we adopted Balance Loss,

Algorithm 1 algorithm of Balance Loss

Input: $map \in \mathbb{R}^{H \times W}$ $label \in \mathbb{R}^{H \times W}$ $w_{neg} \in \mathbb{R}$ $R \in \mathbb{N}^+$
Output: loss

- 1: $(map, label, R) \Rightarrow t$
generate the 0,1 matrix as shown in Fig. 5(B)
- 2: $w \leftarrow t$
- 3: $N_{pos} = R^2$
- 4: $N_{neg} = H \times W - R^2$
num of the positive and negative samples
- 5: $W_{pos} = 1/N_{pos}$
- 6: $W_{neg} = (1/N_{neg}) \times w_{neg}$
weight of the positive and negative samples
- 7: $w \leftarrow w / \sum w$
weight normalization
- 8: $p \leftarrow sigmoid(map)$
map normalization
- 9: $loss = -\sum(log(p_{ij})t_{ij}w_{ij} + log(1 - p_{ij})(1 - t_{ij})w_{ij})$
balance loss
- 10: **return** loss

and Algorithm 1 shows the pseudo-code of the Balance Loss. The core of Balance Loss is to make the weight of positive and negative samples to be 1 : 1. However, in the selection of positive and negative samples we designed some specific strategies. Firstly, hyperparameter R is chosen as the sampling cut-offs to divide the positive and negative samples. As shown in Fig. 5 (B), $R = 1$ means that only the position closest to GroundTruth is taken as a positive sample, and the rest are negative samples, as shown in Fig. 5(a). $R = 2$ means that the four points around GroundTruth are taken as positive samples, as shown in Fig. 5(b). $R = 3$ means that the area with the closest point to GroundTruth as the center and the length and width of 3 are all taken as positive samples. After a slew of tests, we have determined that when $R = 1$, the model performs the best which is actually a bit unexpected. Besides, the negative weight is also an important component affecting the performance of the model , which we discuss in detail in the Section V-E.

IV. EXPERIMENT

A. Implementation Details

We adopt the Deit-S pre-trained on ImageNet as our backbone model and remove the original classifier for ImageNet. The model is trained by AdamW with a weight decay of 5e-4. The initial learning rate is 3e-4, which is reduced by a factor of 10 for the 10th and 14th epoch, respectively. We all train a total of 16 epochs and set the batch size to 16. While training, the size of UAV-view images is 112×112 pixels and the size of Satellite-view images is 400×400 pixels. Besides, we applied data augmentation such as random crop mentioned in Section III-C. We use the backbone to extract features and fuse the features of the UAV and satellite view images by a group convolution to obtain a new feature map incorporating information from both views. Balance Loss is applied to supervise the feature map during the training process. In the post-processing stage of inference, we use hanning windows

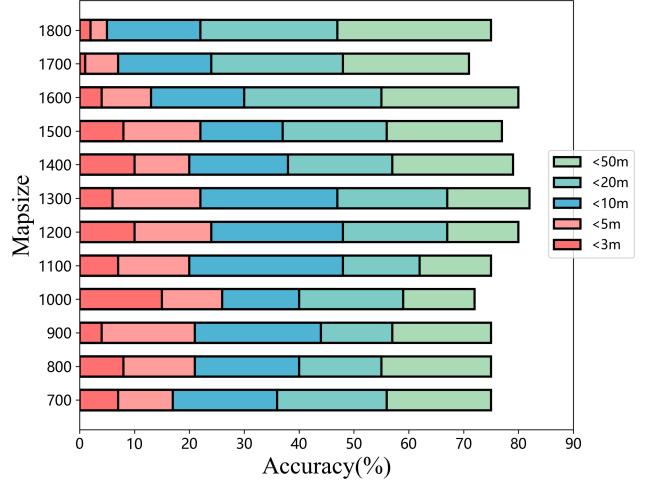


Fig. 6. The relationship between meter-level positioning accuracy and search map scale, where $< 3m$ represents the percentage of positioning error less than 3. Others are the same.

TABLE III
COMPARING FPI WITH STATE-OF-THE-ART MODELS FOR IMAGE RETRIEVAL

Method	Trained Dataset	RDS (%)	Inference Time
LPN[47]	University-1652[15]	17.2	1.48x
FSRA-2B[34]	University-1652[15]	24.42	1x
FSRA-2B[34]	UL14-R	25.57	1x
FPI	UL14	57.22	0.25x

as convolution kernels to filter noise in the heatmap. It is worth mentioning that we do not share model weights for different source inputs. All experiments were conducted on a single RTX3080Ti and were performed using mixed precision training.

B. FPI v.s. Image Retrieval

FPI and image retrieval schemes are significantly different in evaluation indicators, but in order to compare and prove the effectiveness of FPI, we reconstructed the test set to meet the requirements of image retrieval scheme evaluation. We divide each satellite image in the test set into 5×5 small satellite images without overlapping (as shown in Fig. 1(a)) to serve as a gallery of UAV images. Then, the cosine similarity between the UAV-view perspective image and the gallery is calculated to find the most similar satellite image, and convert it to the coordinates in the original satellite image to calculate the RDS indicator. To ensure fairness, the image scales used for retrieval are consistent with FPI. Besides, In order to reduce the gap between domains, we reconstruct the training set for image retrieval in the same way as DenseUAV [7] named UL14-R. We counted RDS and inference time as shown in III. While the speed of FPI is 4 times that of the image retrieval model (FSRA-2B), it also has nearly 33 points of improvement on RDS. We believe that the method of image retrieval can have

better results, but in order to maintain alignment, we use the same input scale as FPI, which may lead to poor image retrieval scheme due to the problem of satellite image and UAV image scale, however, we did not do further research in this paper.

C. The Effect of The Spatial Scale

The spatial scale of the search map will greatly examine the ability of the model to locate accurately. The ability of the model to find the corresponding location in a very large search space will greatly simplify the application of UAV localization tasks. However, since the scale of the model input is often fixed, the ability of the model to be adaptive (resistant to scale shifts) to such scale shifts is crucial in this task. To scrutinize the impact of the model for different scales, we conducted group comparison experiments for satellite images of 700–1800 pixels (we quantified these two values by the geospatial area of the search map, with 700 corresponding to a spatial area of $32,400 \text{ m}^2$ and 1800 corresponding to $214,300 \text{ m}^2$), and the experimental results are shown in Fig. 6. The results show that spatial scale transformation is a challenging factor in UAV localization. However, there is a surprising part of the experimental results, where we expected a monotonically decreasing relationship between scale and accuracy. However, the model still falls short of the expected localization accuracy at small scales. This is mainly due to the fact that in the small-scale case, the query positions are distributed at the edges of the searched map, which leads to a reduction in the amount of available information. It is easy to understand that the real space occupied by a unit pixel decreases after resizing the small-scale satellite image to a fixed scale. However, the spatial scale of the drone is basically kept constant, and if the drone is located at the edge of the satellite image at this time, there will be information loss. We believe that this is the main reason affecting the low accuracy of small-scale satellite images. On the contrary, in the scenario of large scale space to be retrieved, we believe that the excessive amount of spatial information is the main reason that affects the positioning accuracy. This is because the possibility of similar interference regions will be greatly increased in a larger spatial extent.

D. The Effect of Query Distribution

This section explores the influence of the location distribution of the query in the search map on the location accuracy. During the application, the relative position of query in the search map will vary due to factors such as time delays, drone flight speed, etc. In the worst case, the query may be beyond the edge of the search map. This situation, which is like the tracked target leaving the view of the camera, requires some policy adjustments on the application side. In this paper, we focus on the impact of query located in different position intervals of the search map on the localization accuracy. As shown in Fig. 7(a), we divide different levels according to the relative spatial distance between query and the center point of search map (0-0.2 means the query is distributed in a circle with diameter 0.2, 0.2-0.4 means the query is distributed in a ring with diameter 0.2-0.4, >1.0 means other parts). We

used the meter-level evaluation to do the analysis, and the experimental results are shown in Fig. 7(b). We can clearly see that there is a significant decrease in positioning accuracy in the case of >1.0 . This result is not difficult to explain. To take an extreme example, when the query is located on 4 corners of the searched map, only 1/4 of the information will be available, and the remaining 3/4 of the image information will not find the corresponding information on the searched map, which undoubtedly increases the difficulty of localization. The reason for the low accuracy of the 0.8-1.0 interval is similar to the above. This interval will lose half of the image information in the worst case (i.e. the query is located at the intersection of the circle and the 4 sides of the image).

E. The Effect of Drone Flight Altitude

In the actual application scenario, the altitude of the UAV flight is not fixed, which directly affects the spatial scale of the UAV acquisition to the query. Taking this into account, UL14 collected UAV images from 80, 90 and 100m altitudes during data acquisition. In the testing phase, we evaluate according to height-level. The experimental results are shown in Fig. 8, and it can be seen that the model has good robustness to the spatial scale of query. Due to the limitation of UAV flight height, there is no way to develop experiments at larger heights.

V. ABLATION STUDY

A. Impact of Model Architecture

Several architecturally representative models are selected for experimentation. The most representative ones in CNN are Resnet architecture, ViT, the pioneer of Transformer in the field of vision, and Swin-Transformer, which has the same hierarchical structure as Resnet and has achieved SOTA in various fields. However, since Swin-Transformer has high requirements for the input, PvT is used instead in this paper in order to control the input scale to maintain consistency. Deit-S is the backbone used in our benchmark. The experimental results are shown in Fig. 9. It is very surprising that the CNN-based model is not optimized on the benchmark of this paper, and we guess that the rotation and scale uncertainty of the UAV localization task make the CNN model unoptimized. Although the model based on Deit-S is slightly inferior to ViT-S in terms of inference speed, it improves the RDS by more than one point. The parameter details and inference time of the 4 backbone models are added to Table IV.

B. The Effect of Input Scale

In the model training phase, a larger input scale generally represents richer information. In UAV localization tasks, high-resolution input images can provide more spatial information. To test this hypothesis, we conduct two sets of comparison experiments on input images of different scales. One group is to control the UAV image scale as 112 pixels and change the satellite image scale ($272 \rightarrow 432$ pixels; stride=32), and the experimental results are shown in Fig. 10(b). The other group is to control the satellite image scale as 400 pixels and change the UAV image scale ($80 \rightarrow 160$ pixels; stride=16), and the

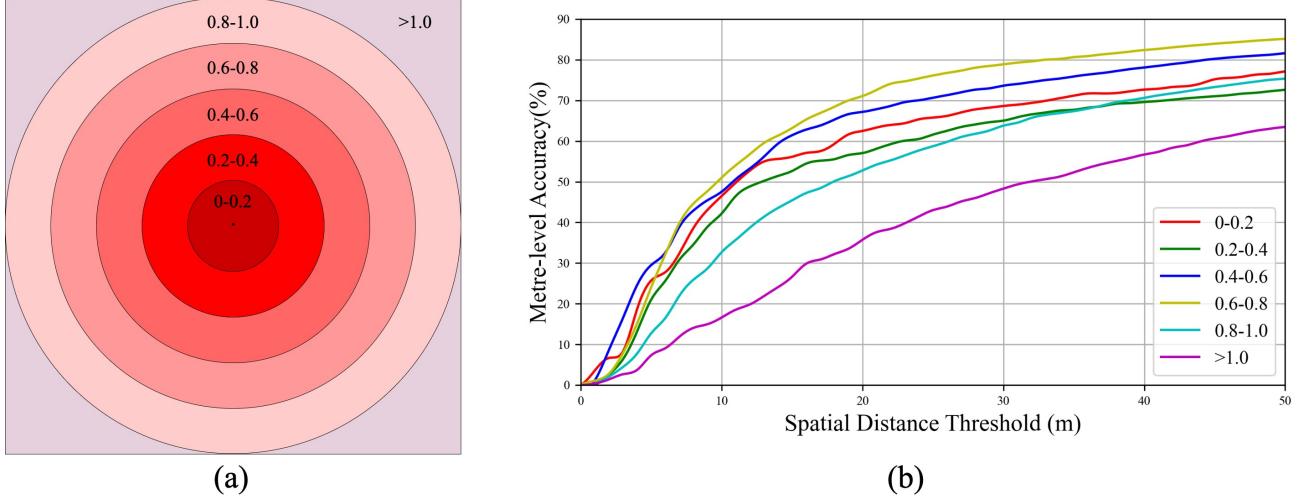


Fig. 7. (a) is the spatial distribution of query in the search map. (b) MA curves under different spatial distributions. It can be observed from (b) that the positioning accuracy between 0 and 0.8 is relatively close. The localization accuracy is poor in regions other than 0.8, which is mainly related to the loss of information.

TABLE IV

THE TABLE COMPARING THE IMPACT OF DIFFERENT BACKBONE ON PERFORMANCE. STRIDE DENOTES THE COMPRESSION MULTIPLE OF THE SCALE, DIM DENOTES THE CHANNEL DIMENSION OF THE FINAL DATA, $< 5m$ DENOTE $MA_{<5m}$, AND INFERENCE TIME DENOTES THE RELATIVE INFERENCE TIME

Backbone	Stride	Dim	$< 5m(\%)$	$< 10m(\%)$	$< 20m(\%)$	RDS(%)	Inference Time
Resnet50[48]	16	2048	0.14	0.53	2.29	9.30	1.28×
Pvt-S[49]	16	256	14.23	31.69	50.88	52.37	2.02×
ViT-S[38]	16	768	19.03	38.03	56.25	56.20	1×
Deit-S[39]	16	384	18.63	38.36	57.67	57.22	1.08×

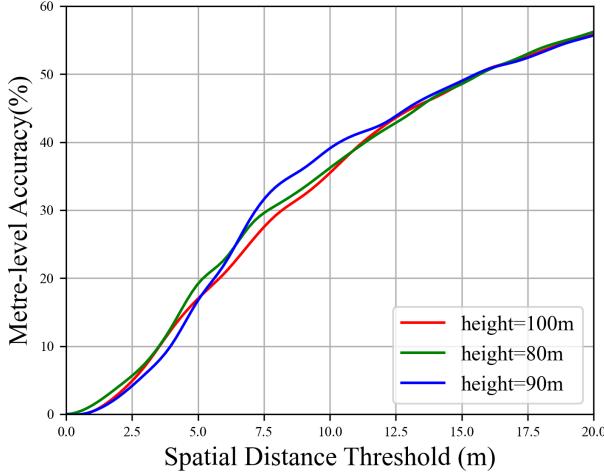


Fig. 8. MA curves of UAV at different flight altitudes, height refers to the flying height of the drone, in meters.

experimental results are shown in Fig. 10(a). It is well known that the inference time increases as the input scale increases. Additional, we counted the curves of inference time as shown in Fig. 10(c)(d). It is worth noting that the effect of image scale variation of UAV on the inference time is smaller, mainly because the image scale of UAV is smaller compared with

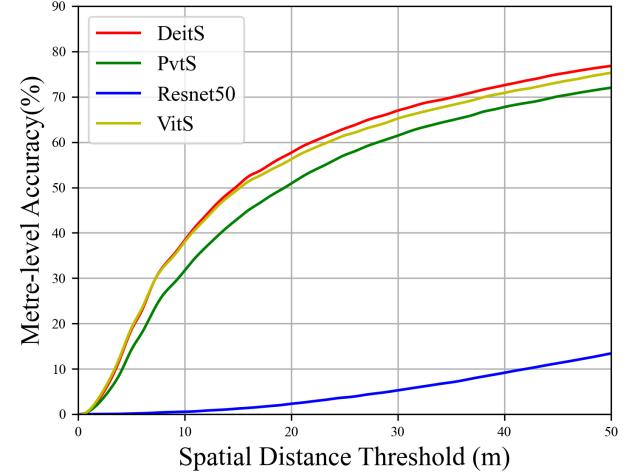


Fig. 9. MA curves of different backbone models.

that of satellite images. The results are largely consistent with expectations, and we can conclude that the resolution of model inputs can directly affect the accuracy of localization in UAV localization tasks. In the process of practical application, one can choose the appropriate input scale ratio according to one's hardware equipment, and the indicator in Fig. 10 can serve as a reference. The satellite image and UAV image scales used

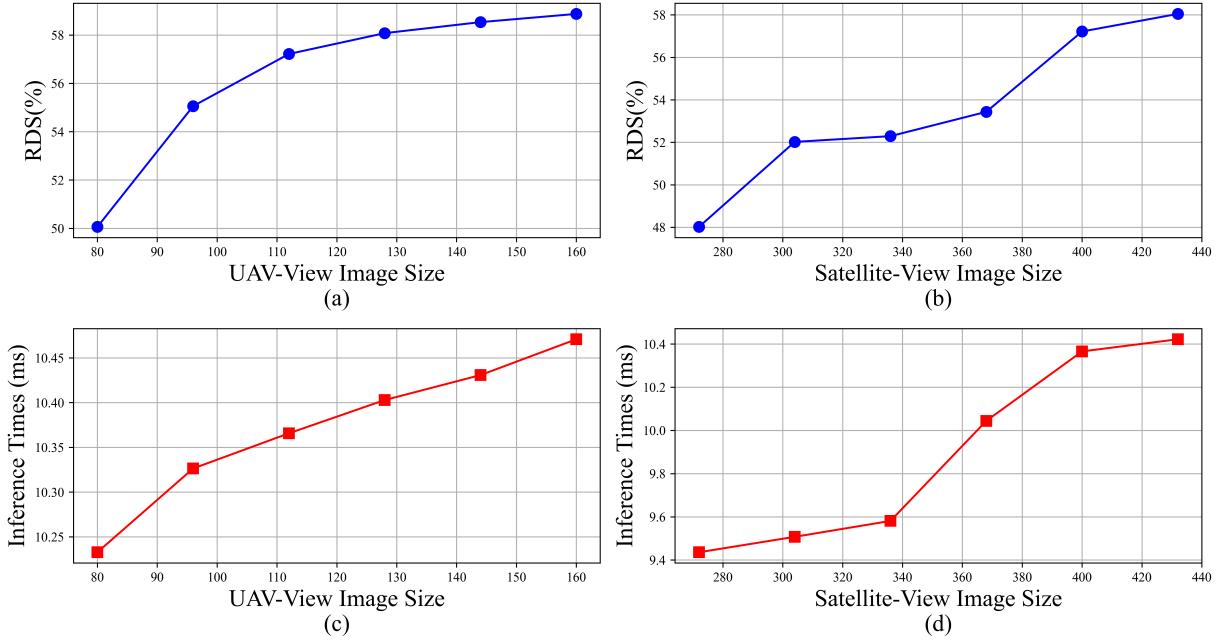


Fig. 10. The impact of varied input scales on the performance of FPI. For one experiment, (a)(c) illustrate the variation curves of RDS and Inference Time, respectively, with the satellite image scale set to 400 and the UAV scale set to 80-160 with a 16-pixels interval. (b)(d) illustrate the variation curves of RDS and Inference Time, respectively, with the UAV image scale set to 112 and the Satellite scale set to 272-432 with a 32-pixels interval.

in this paper are 400 and 112 pixels, respectively.

C. The Effect of Sharing Weight

Twin networks have achieved remarkable results in the fields of cross-view geo-localization, object tracking, etc. The main advantage is that it allows input data from different sources to share one set of weight parameters, thus reducing the number of parameters. FPI-based UAV localization is also a multi-source input task, so twin networks are also applicable. However, since the input scales of UAV and satellite images are not the consistent, they cannot share a common position encoding vector while sharing weights. The solution is to assign two different lengths of position encoding to the query and the search map. As shown in Table V, we find that the twin-based architecture is slightly inferior to the non-twin-based architecture in the FPI benchmark. the RDS metric is 1.2 points lower than that of the non-twin-based network. This means that the twin network, although it can serve to reduce the number of parameters, is not a good enhancement for the UAV localization benchmark proposed in this paper.

D. The Effect of Padding

The final convolution in single object tracking algorithms like SiamFC employs group convolution without padding, which is related to their training data. Even though the frameworks are similar, the challenges that are tackled are vastly different. In the correlation operation, if convolution without padding is used, not only the edge information is lost, but also the spatial correlation is destroyed. The goal of group convolution with added padding is not to change the scale of the feature map, but rather to compare the similarity between

TABLE V
THE EFFECT OF SHARED WEIGHTS ON THE FPI MODEL

Share Weight	RDS
✓	56.01
✗	57.21

TABLE VI
THE EFFECT OF PADDING IN CORRELATION OPERATION

Padding	RDS
✗	47.69
✓	57.21

the convolution kernel (feature map corresponding to query) and the search map, so that the scale of the input and output remains the same. Table VI shows the experimental findings, which indicate that by simply adding the padding operation, the RDS Indicator improves by roughly 9 points.

E. The Effect of Negative Weight in Balance Loss

Balance Loss [8] is the practice of setting the weight of positive samples to negative samples to 1:1, but the entire balance is not conducive to model training. Because we only take the nearest point of ground truth as a positive sample and all other samples as negative samples, each negative sample has a tiny weight and a single positive sample has a huge weight. We experiment with increasing the weight of negative samples and find that it has a significant impact. The model works best when the negative weight approaches 15, as shown

in Fig. 11(a), which is approximately a 13-point improvement over negative weight=1.

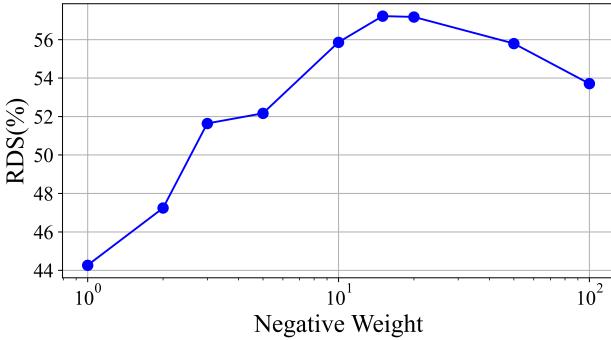


Fig. 11. The curve of RDS with the Negative Weight parameter, the horizontal coordinate is the exponent of 10.

VI. VISUALIZATION

For prediction, we randomly selected some pairs of UAV and Satellite images from the testing set, and the results are displayed in Fig. 12. We anticipate that the greater the resemblance, the higher the heat value. The distribution of heat tends to be near to the genuine site, as indicated in the heatmap column in Fig. 12. The part with the greatest thermal force has a high centre and a low surrounding. Furthermore, we believe that this task is extremely challenging since even the human eye has difficulty recognising the correct place rapidly. In addition, in order to reflect the difficulty of the UAV visual localization task, we recruited 3 volunteers to manually find the position of the query image in the search map which will be discussed in Appendix C.

VII. CONCLUSION

This paper proposes a method of Finding Points with Image to solve the Geo-localization problem, and constructs a dataset UL14 for UAV visual localization task. UL14 includes both drone and satellite view images. FPI, in short, is the task of finding the query position directly in the search map. Compared with the previous image-based retrieval methods, the computational and storage overhead is greatly reduced. At the same time, it also provides the possibility to achieve higher-precision positioning. In addition, this paper proposes MA and RDS evaluation indicators to more intuitively evaluate the accuracy of UAV positioning. At the model level, we build a strong baseline based on Transformer, and implement feature interaction between query and search map through a group convolution. The final FPI-based benchmark has an accuracy of 57.7% within a 20-meter error. Besides, when the image retrieval method uses non-overlapping cutting to construct the gallery, the FPI is increased by nearly 4 times, and the RDS is increased by nearly 33 points. It is also worth mentioning that the inference speed of FPI on a single image pair on 3080Ti can reach about 10 ms. However, FPI still has many limitations, such as the ability to resist scale transformation. In Fig. 6, we can see that the effect of scale on the localization

accuracy is huge. How to resist the change of scale will be the problem that we need to focus on in the future. It is worth mentioning that FPI can be used as a paradigm to solve the localization problem of all multi-source inputs. In the future, we will also apply the point-finding method to other visual localization tasks.

REFERENCES

- [1] H. Ma, E. Smart, A. Ahmed, and D. Brown, "Radar image-based positioning for usv under gps denial environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 72–80, 2018, doi:<https://doi.org/10.1109/TITS.2017.2690577>.
- [2] J. Han, Y. Cho, and J. Kim, "Coastal slam with marine radar for usv operation in gps-restricted situations," *IEEE Journal of Oceanic Engineering*, vol. 44, no. 2, pp. 300–309, 2019, doi:<https://doi.org/10.1109/JOE.2018.2883887>.
- [3] H. Ma, E. Smart, A. Ahmed, and D. Brown, "Radar image-based positioning for usv under gps denial environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 72–80, 2018, doi:<https://doi.org/10.1109/TITS.2017.2690577>.
- [4] D. Turner, A. Lucieer, and L. Wallace, "Direct georeferencing of ultrahigh-resolution uav imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 5, pp. 2738–2745, 2014.
- [5] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, 2004.
- [6] H. Bay, A. Ess, T.uytelaars, and L. V. Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding*, 2008.
- [7] M. Dai, J. Huang, J. Zhuang, W. Lan, Y. Cai, and E. Zheng, "Vision-based uav localization system in denial environments," *arXiv preprint arXiv:2201.09201*, 2022.
- [8] M. Cen and C. Jung, "Fully convolutional siamese fusion networks for object tracking," in *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, oct 2018, doi:<https://doi.org/10.1109/icip.2018.8451102>.
- [9] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocation," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015, doi:<https://doi.org/10.1109/cvpr.2015.7299135>.
- [10] Y. Tian, C. Chen, and M. Shah, "Cross-view image matching for geo-localization in urban environments," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017, doi:<https://doi.org/10.1109/cvpr.2017.216>.
- [11] N. N. Vo and J. Hays, "Localizing and orienting street views using overhead imagery," in *Computer Vision – ECCV 2016*. Springer International Publishing, 2016, pp. 494–509, doi:https://doi.org/10.1007/978-3-319-46448-0_30.
- [12] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs, "Predicting ground-level scene layout from aerial imagery," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jul 2017, doi:<https://doi.org/10.1109/cvpr.2017.440>.
- [13] L. Liu and H. Li, "Lending orientation to neural networks for cross-view geo-localization," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2019, doi:<https://doi.org/10.1109/cvpr.2019.00577>.
- [14] S. Zhu, T. Yang, and C. Chen, "VIGOR: Cross-view image geo-localization beyond one-to-one retrieval," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021, doi:<https://doi.org/10.1109/cvpr46437.2021.00364>.
- [15] Z. Zheng, Y. Wei, and Y. Yang, "University-1652: A multi-view multi-source benchmark for drone-based geo-localization," in *Proceedings of the 28th ACM international conference on Multimedia*, 2020, pp. 1395–1403, doi:<https://doi.org/10.1145/3394171.3413896>.
- [16] F. Castaldo, A. Zamir, R. Angst, F. Palmieri, and S. Savarese, "Semantic cross-view matching," in *2015 IEEE International Conference on Computer Vision Workshop (ICCVW)*. IEEE, dec 2015, doi:<https://doi.org/10.1109/iccvw.2015.137>.
- [17] T.-Y. Lin, S. Belongie, and J. Hays, "Cross-view image geolocalization," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2013, doi:<https://doi.org/10.1109/cvpr.2013.120>.
- [18] T. Senlet and A. Elgammal, "A framework for global vehicle localization using stereo images and satellite and road maps," in *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, nov 2011, doi:<https://doi.org/10.1109/iccvw.2011.6130498>.

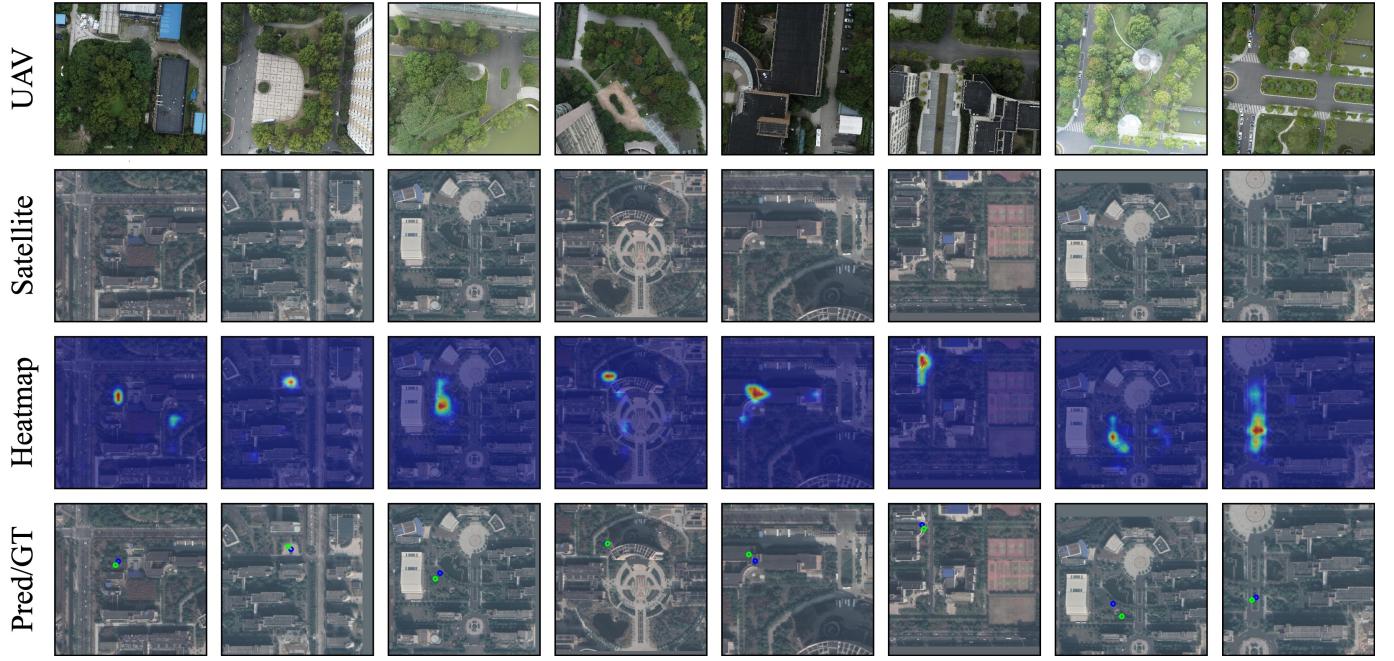


Fig. 12. *UAV* denotes the input UAV-view images, *Satellite* denotes the input satellite-view images, *Heatmap* denotes the outputs of the model, and *Pred/GT* denotes the final predicted and actual positions, respectively (green is groundtruth, blue is predicted position)

- [19] M. Bansal, H. S. Sawhney, H. Cheng, and K. Daniilidis, “Geo-localization of street views with aerial image databases,” in *Proceedings of the 19th ACM international conference on Multimedia - MM '11*. ACM Press, 2011, doi:<https://doi.org/10.1145/2072298.2071954>.
- [20] S. Workman and N. Jacobs, “On the location dependence of convolutional neural network features,” in *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*. IEEE, jun 2015, doi:<https://doi.org/10.1109/cvprw.2015.7301385>.
- [21] S. Workman, R. Souvenir, and N. Jacobs, “Wide-area image geolocation with aerial reference imagery,” in *2015 IEEE International Conference on Computer Vision (ICCV)*. IEEE, dec 2015, doi:<https://doi.org/10.1109/cvpr.2005.202>.
- [22] S. Chopra, R. Hadsell, and Y. LeCun, “Learning a similarity metric discriminatively, with application to face verification,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. IEEE, doi:<https://doi.org/10.1109/cvpr.2005.202>.
- [23] R. Hadsell, S. Chopra, and Y. LeCun, “Dimensionality reduction by learning an invariant mapping,” in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2 (CVPR'06)*. IEEE, doi:<https://doi.org/10.1109/cvpr.2006.100>.
- [24] W. Deng, L. Zheng, Q. Ye, G. Kang, Y. Yang, and J. Jiao, “Image-image domain adaptation with preserved self-similarity and domain-dissimilarity for person re-identification,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018, doi:<https://doi.org/10.1109/cvpr.2018.00110>.
- [25] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, doi:<https://doi.org/10.1109/cvpr.2016.572>.
- [26] 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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. IEEE, jun 2018, doi:<https://doi.org/10.1109/cvpr.2018.00758>.
- [27] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, “Optimal feature transport for cross-view image geo-localization,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, pp. 11990–11997, apr 2020, doi:<https://doi.org/10.1609/aaai.v34i07.6875>.
- [28] Y. Shi, L. Liu, X. Yu, and H. Li, “Spatial-aware feature aggregation for image based cross-view geo-localization,” *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [29] Y. Shi, X. Yu, D. Campbell, and H. Li, “Where am i looking at? joint location and orientation estimation by cross-view matching,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2020, doi:<https://doi.org/10.1109/cvpr42600.2020.00412>.
- [30] Z. Zeng, Z. Wang, F. Yang, and S. Satoh, “Geo-localization via ground-to-satellite cross-view image retrieval,” *IEEE Transactions on Multimedia*, pp. 1–1, 2022, doi:<https://doi.org/10.1109/TMM.2022.3144066>.
- [31] T. Wang, Z. Zheng, C. Yan, J. Zhang, Y. Sun, B. Zhenga, and Y. Yang, “Each part matters: Local patterns facilitate cross-view geo-localization,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, doi:<https://doi.org/10.1109/TCSVT.2021.3061265>.
- [32] X. Tian, J. Shao, D. Ouyang, and H. T. Shen, “Uav-satellite view synthesis for cross-view geo-localization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4804–4815, 2022.
- [33] Y. Zhu, B. Sun, X. Lu, and S. Jia, “Geographic semantic network for cross-view image geo-localization,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022, doi:<https://doi.org/10.1109/TGRS.2021.3121337>.
- [34] M. Dai, J. Hu, J. Zhuang, and E. Zheng, “A transformer-based feature segmentation and region alignment method for UAV-view geo-localization,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2021, doi:<https://doi.org/10.1109/tcsvt.2021.3135013>.
- [35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [36] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv preprint arXiv:1810.04805*, 2018.
- [37] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature pyramid networks for object detection,” in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 936–944, doi:<https://doi.org/10.1109/CVPR.2017.106>.
- [38] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.
- [39] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers & distillation through attention,” in *International Conference on Machine Learning*, vol. 139, July 2021, pp. 10347–10357.
- [40] W. Wang, E. Xie, X. Li, D.-P. Fan, K. Song, D. Liang, T. Lu, P. Luo, and L. Shao, “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, oct 2021, doi:<https://doi.org/10.1109%2Ficcv48922.2021.00061>.

- [41] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10 012–10 022.
- [42] X. Chen, B. Yan, J. Zhu, D. Wang, X. Yang, and H. Lu, "Transformer tracking," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2021, doi:<https://doi.org/10.1109/cvpr46437.2021.00803>.
- [43] L. Lin, H. Fan, Y. Xu, and H. Ling, "Swintrack: A simple and strong baseline for transformer tracking," *arXiv preprint arXiv:2112.00995*, 2021.
- [44] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," *Advances in neural information processing systems*, vol. 31, 2018.
- [45] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*. IEEE, oct 2017, doi:<https://doi.org/10.1109/iccv.2017.324>.
- [46] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.
- [47] T. Wang, Z. Zheng, C. Yan, J. Zhang, Y. Sun, B. Zhenga, and Y. Yang, "Each part matters: Local patterns facilitate cross-view geo-localization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, doi:<https://doi.org/10.1109/TCSVT.2021.3061265>.
- [48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2016, doi:<https://doi.org/10.1109/cvpr.2016.90>.
- [49] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha *et al.*, "Resnest: Split-attention networks," *arXiv preprint arXiv:2004.08955*, 2020.