

Vision-Based UAV Self-Positioning in Low-Altitude Urban Environments

Ming Dai®, Enhui Zheng®, Zhenhua Feng®, Senior Member, IEEE, Lei Qi®, Jiedong Zhuang®, and Wankou Yang®, Member, IEEE

Abstract—Unmanned Aerial Vehicles (UAVs) rely on satellite systems for stable positioning. However, due to limited satellite coverage or communication disruptions, UAVs may lose signals for positioning. In such situations, vision-based techniques can serve as an alternative, ensuring the self-positioning capability of UAVs. However, most of the existing datasets are developed for the geo-localization task of the objects captured by UAVs, rather than UAV self-positioning. Furthermore, the existing UAV datasets adopt discrete sampling to synthetic data, such as Google Maps, neglecting the crucial aspects of dense sampling and the uncertainties commonly experienced in practical scenarios. To address these issues, this paper presents a new dataset, DenseUA V, that is the first publicly available dataset tailored for the UAV self-positioning task. DenseUA V adopts dense sampling on UAV images obtained in low-altitude urban areas. In total, over 27K UAV- and satellite-view images of 14 university campuses are collected and annotated. In terms of methodology, we first verify the superiority of Transformers over CNNs for the proposed task. Then we incorporate metric learning into representation learning to enhance the model's discriminative capacity and to reduce the modality discrepancy. Besides, to facilitate joint learning from both the satellite and UAV views, we introduce a mutually supervised learning approach. Last, we enhance the Recall@K metric and introduce a new measurement, SDM@K, to evaluate both the retrieval and localization performance for the proposed task. As a result, the proposed baseline method achieves a remarkable Recall@1 score of 83.01% and an SDM@1 score of 86.50% on DenseUA V. The dataset and code have been made publicly available on https://github.com/Dmmdm1997/DenseUA V.

Index Terms—Unmanned aerial vehicle, geo-localization, transformer, image retrieval.

I. INTRODUCTION

In recent years, Unmanned Aerial Vehicles (UAVs) have become increasingly vital in various applications, such as agricultural operations, ground reconnaissance, and civilian
aerial photography [1], [2], [3], [4]. With the advancements in onboard computing capabilities and lightweight models, different visual algorithms, including object tracking, object detection, and Simultaneous Localization And Mapping (SLAM) [5], [6], [7], [8], have been widely deployed in UAVs. Notably, recent developments in UAV-related visual localization tasks have become very popular. The pioneering University-1652 dataset [1] introduces the use of drone view for cross-view geo-localization, such as drone-view target localization and drone navigation, which extend the applications of UAV geo-localization.

As the applications of UAVs become more widespread, ensuring the stability of UAV's self-positioning has emerged as a crucial concern. In general, this heaviness relies on satellite systems. However, maintaining stable connections between UAVs and satellites is challenging due to factors such as the fixed frequency band of GPS signals, obstructions, and signal interference. To mitigate the above issues, we propose a novel UAV self-positioning task. In contrast to the University-1652 dataset, our focus is on determining the position of UAVs rather than the captured objects, aligning with the goals presented in previous works [9], [10]. However, our approach differs in that we do not employ panoramic images as the UAV-view image. Instead, we offer a simpler solution by vertically orienting the UAV camera towards the ground, thus mitigating view deviation issues.

To facilitate the proposed task, we create a new DenseUA V dataset. The key features of this dataset include:

- Real scene sampling: Unlike existing datasets [1], [11] that rely on synthetic data, e.g., from Google Maps, DenseUA V captures real scenes.
- Dense sampling: Existing datasets primarily focus on mining salient features, such as buildings, overlooking the impact of overlapping regions between adjacent images on positioning. DenseUA V emphasizes dense sampling, enabling high-precision UAV self-positioning.
- Multi-height sampling: DenseUA V incorporates samples captured at three different heights, enhancing the model’s robustness to varying UAV flying heights.
- Multi-time-node satellite images: We introduce multiple temporal satellite images to facilitate the model’s adaptability to spatial changes caused by temporal offsets.
- Multi-scale satellite images: The satellite imagery contains multi-scale images, strengthening the model’s resilience to varying scales of satellite images.

To provide a baseline method for the proposed task, we develop a Transformer-based approach. We also use a
Siamese network that allows the model to learn shared representations across different modalities. Extensive investigations are completed to analyze the impact of various components, such as data augmentation, backbone network, prediction head, and loss function, on the final performance of the baseline method. Second, we introduce metric learning and mutual learning to our baseline method. Metric learning is motivated by the fact that UAV self-positioning involves image matching across different modalities. It addresses the modality discrepancy issue effectively. Moreover, the incorporation of mutual learning facilitates concurrent learning between different modalities, ensuring their alignment throughout the training process.

As shown in Fig. 1, the UAV self-positioning pipeline has four main steps. For pre-processing, we slice the satellite images acquired from 20-level Google Maps and construct a satellite image set. Then, we extract their image features \{F1, F2, ..., FN\} using a vision Transformer model. In the real-time processing stage, the vertical ground camera of the UAV is employed to capture the corresponding UAV-view image, and the associated image feature D_F is obtained. Subsequently, in the matching stage, we calculate the feature similarity between the UAV and satellite gallery images, resulting in the generation of a corresponding feature distance matrix. Lastly, the post-processing stage obtains optimal matching through a positioning strategy, thereby enabling continuous positioning. The evaluation metric is another essential element. Currently, Recall@K [12], [13] is the mainstream evaluation metric for most image retrieval tasks. In general, the Recall@K indicator considers the accuracy of recall results from the perspective of retrieval, and it cannot measure the deviation of spatial distance. Therefore, we propose SDM@K that not only retains the properties of retrieval but also introduces the determination of positioning errors.

In summary, the main contributions of this paper include:

- We construct a low-altitude urban scene dataset, DenseUAV, which is constructed by dense sampling from the UAV’s down-looking camera in real scenarios.
- We propose a UAV-related sub-task called UAV self-positioning, and establish a complete training and inference framework. We also build a Transformer-based strong baseline model to support this task.
- We improve R@K and propose an evaluation metric tailored for UAV self-positioning called SDM@K, which considers the accuracy of UAV self-positioning from both perspectives of retrieval and positioning.
- We extensively evaluate the impact of some key components on the proposed DenseUAV dataset.

The rest of the paper is organized as follows. We first introduce the related work in Sec. II. Then, we present the proposed dataset in Sec. III, and the new evaluation metric in Sec. IV. Next, we introduce the baseline model in Sec. V. The experimental results and analysis are presented in Sec. VI. Last, we draw the conclusion in Sec. VII.

II. RELATED WORK

In this section, we introduce the existing studies by dividing them into three categories: geo-localization datasets, geo-localization methods, and other related methods.

A. Geo-Localization Datasets

1) Ground-to-Aerial Matching: The geo-localization task was initially proposed to solve the ground-to-aerial matching problem. Some pioneering studies of [15], [16], and [17] were proposed to use publicly available resources to build image pairs for ground and aerial view images. Later, CVUSA [9] constructed image pairs from ground-based panoramic images and satellite images, while CVACT [10] added spatial factors to CVUSA, i.e., orientation maps. Recently, VIGOR [14] changed the previous center-to-center image matching and redefined the problem with a more realistic assumption that the query image can be arbitrary in the area of interest.

2) Drone-to-Satellite Matching: With the development of UAVs, the geo-localization task goes beyond ground-based operations. University-1652 [1] introduced the drone-view into cross-view geo-localization, using the drone-view as a transition view to reduce the difficulty of matching between ground and satellite views, and regarded it as a retrieval task. In addition, University-1652 proposed two UAV-based
subtasks, namely drone-view target localization and drone navigation. Further, SUE-200 [11] improved the adaptability of a model in terms of flight altitudes by collecting drone images at four different altitudes. The proposal of DenseUAV is originally inspired by University-1652. However, they have significant differences. University-1652 primarily focuses on identifying and localizing buildings with prominent features, and these buildings are discretely distributed in space. In contrast, DenseUAV is designed to address the problem of self-positioning of UAVs in GPS-denied environments. This requires a model to not only consider prominent features but also be aware of the spatial distribution of objects in images.

B. Deep-Learning-Based Geo-Localization

1) Supervision-Based Methods: This category primarily focuses on different types of supervision learning and enhances the discriminative capability of a model by introducing new loss functions. The commonly used CNN methods are supervised by the cross-entropy loss [18], which involves image feature embedding via classification supervision. In order to improve the matching reliability, triplet loss [19] was utilized to reduce the distance between positive pairs and increase the distance between negative pairs. Additionally, several methods based on triplet loss were proposed to further strengthen the discriminative ability of features [20], [21], [22], [23]. By pulling the distance between positive pairs, contrastive loss [9], [24] could further enhance the performance of a geo-localization model. Moreover, some multi-task supervised learning methods [10], [25], [26] were introduced subsequently to further boost the discriminative ability of the model.

2) Matching-Based Methods: This category focuses on the spatial information asymmetry between different domains. Some existing approaches aim to reduce the variance of different domains through view transformation. For example, Shi et al. used the polar coordinate transformation [27], [28] to map ground panorama images to a top-down perspective. Although this method is highly efficient, it suffers from distortion and information loss. For another example, Toker et al. used GAN to generate images from the target domain [29]. Another focus in this category is to overcome the viewpoint bias and feature misalignment issues between different domains [19], [30]. Some methods have tried to minimize domain differences by utilizing the results of pixel-level segmentation [10], [31]. Also, feature alignment has been used for specific scenes by employing feature chunking. For instance, Local Pattern Network (LPN) [32] proposed a square chunking strategy to extract useful information from the edges and the chunking scheme, resulting in significant performance boosting. FSRA [33] automatically divides the corresponding instances through the distribution of thermal values and performs metric learning on the corresponding regions of the two domains.

C. Other Related Methods

In this part, we introduce some other related methods. First, Yang et al. [34] introduced a general framework to enhance visual computing through multi-level/modal feature integration, which motivates us to carry out multi-source image representation. In addition, the framework of UAV self-positioning is similar to some retrieval tasks. In re-identification [35], [36], [37], [38], representation learning and metric learning often work together to form suitable clustering in the feature representation space. In face recognition [39], [40], it usually includes a huge number of subjects, which requires a model to retain strong fine-grained mining capabilities and emphasize the clarity of classification boundaries. As for text-related retrieval tasks [41], [42], [43], based on the support of a large amount of data pairs, contrastive learning is often used as a supervision method. For geo-localization tasks [1], [9], a model is required to adaptively map images from multiple domain spaces into the same representation space and usually needs to overcome viewing angle bias and spatial scale issues. In fact, UAV self-positioning inherits the challenges of geo-localization, additionally has greater domain differences, and requires the ability to capture the relative position in space while overcoming the problem of dynamic object changes caused by time.

III. THE DENSEUAV DATASET

In this part, we first introduce the key features of the proposed DenseUAV dataset and emphasize its distinctions from the existing datasets. Then we present the sampling method and the composition of the dataset in detail.

A. Characteristics

First, it should be highlighted that the aim of the proposed DenseUAV dataset is UAV self-positioning, which has not been addressed in previous datasets. Additionally, DenseUAV is a dataset designed for low-altitude urban scenes, comprising perspectives from UAV-view and Satellite-view. More details of DenseUAV are shown in Table I. These details encompass a comprehensive analysis of various aspects, such as the quantity

| Datasets | DenseUAV(ours) | SUES-200 [11] | University-1652 [1] | VIGOR [16] | CVUSA [9] | Tsn et al. [15] |
|----------|----------------|---------------|---------------------|------------|-----------|----------------|
| Training | 10 x 255 x 9   | 120 x 51      | 701 x 71.6          | 91k x 53k  | 15.5k x 2 | 15.7k x 2 |
| Platform | Drone, Satellite | Drone, Satellite | Drone, Ground, Satellite | Ground, Aerial | Ground, Satellite | Ground, Aerial |
| Images/Platform | 3 x 6 | 50 x 1 | 54 x 16.64 x 1 | / | 1 + 1 | 1 + 1 |
| Target | UAV | Diverse | Building | User | User | Building |
| Sampling | Dense | Discrete | Discrete | Discrete | Discrete | Discrete |
| Source | Real Scenes | Google Map | Google Map | Google Map | Google Map | Google Map |
| Evaluation | R@K & SDIM@K | R@K & AP & RB & PF | R@K & AP | Meter Accuracy | R@K | PR & AP |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 2. A diagram describing the process of dataset creation. The red circles represent the sampling locations, with an interval of 20 meters, and sampled at 80, 90, and 100 meters altitude, respectively. The green arrows represent the orientation of the camera. T is the current sampling point. T-1 and T+1 are the previous and next sampling points. The orange dotted line represents the overlap area. The upper right corner is the overlooking route, and the black arrow represents the flight direction.

of training data, the data composition platform, the specific number of images encompassed within each platform, the designated positioning target, the sampling method employed, the source of the data, and the associated evaluation indicators. Some notable characteristics of DenseUAV at the data level will be further introduced below.

1) Dense Sampling: Existing datasets [1], [10] usually use discrete sampling methods based on landmark buildings during the dataset construction phase to capture intricate architectural features. Nevertheless, this approach introduces a noticeable disparity between various categories due to discrete sampling. Consequently, the retrieval performance during testing is hindered, as the challenge of accurately retrieving the desired information is not sufficiently pronounced. The proposed DenseUAV dataset differs from previous geolocalization datasets mainly in terms of ‘dense sampling’ which refers to the fact that there will be some overlapping areas between adjacent frames. In the training process, a model is required to not only capture fine-grained features but also recognize spatial relative information. Similarly, interference from neighboring frames significantly affects the testing process thus making the benchmark more challenging.

2) Real Scene: Given the high costs of collecting real-world data, the majority of existing datasets [1], [11] used in the research community are obtained from diverse perspectives using Google Map. Another notable distinction of the DenseUAV dataset is that all the UAV-view images are captured from authentic real-world scenes. The rationale behind this lies in the fact that, in real-world applications, the time gap between the UAV-view image and satellite-view image leads to varying degrees of discrepancy at the same location, posing a significant challenge. Moreover, real data is deemed more suitable for practical scenarios, thereby minimizing the disparities between the dataset and real-world landing scenes.

3) Others: In addition to the aforementioned features, DenseUAV incorporates three different altitude levels for image sampling, which means that the captured UAV images possess varying spatial coverage ranges. Similarly, during the construction of satellite imagery, we also employ three different scales of images. Moreover, to address the spatial information variations arising from temporal disparities, satellite images are generated by using multiple temporal instances.

B. Dataset Construction

After discussing the main characteristics of the dataset, this subsection covers the data collection scheme and composition.

1) Sampling Scheme for UAV-View Images: The proposed DenseUAV dataset considers three main factors to capture UAV images. Firstly, by considering the influence of UAV flight height on image scale, the UAV images are collected at three different heights (80m, 90m, and 100m relative to the ground), ensuring nearly identical latitude and longitude coordinates. This is achieved through the use of DJI Pilot’s waypoint action, with an error control within 1m. Secondly, to address the impact of weather and light levels on image quality, a random weather and random time period sampling method is used. This includes varying weather conditions (sunny and cloudy days) and random sampling between 6:00 am and 6:00 pm. Thirdly, to mitigate perspective bias resulting from changes in UAV camera orientation, a standardized configuration is implemented with the camera facing vertically downward. Additionally, rotational data enhancement techniques are employed to address the UAV flight orientation issue (see Section VI-D). Furthermore, we set a fixed sampling distance of 20m. This serves on both the creation of a uniformly spaced dataset and the enhancement of a model’s ability to learn subtle differences between classes. The upper part of Fig. 2 demonstrates the images captured at adjacent sampling points (T-1, T, and T+1), providing a temporal sequence of images. Meanwhile, the upper right
corner of the figure illustrates the flight path of the UAV, which overlooks the scene during data collection.

2) Acquisition of Satellite-View Images: For satellite images, we collect Google Map images at level 20. To enrich the diversity of geographic information and facilitate a model’s adaptation to spatial changes over time, two years (2020 and 2022) of satellite images are included in the DenseUAV dataset. Furthermore, to improve the model’s resilience to the variations in satellite image scales, three different scales of satellite images are included in the dataset. The visualization results demonstrating different scales are presented in the right part of Fig. 2.

3) Dataset Composition: Regarding the composition of the dataset, we collect real data through UAV sampling at fourteen universities in Zhejiang, China. As presented in Table II, the training set consists of 6768 UAV-view images captured from 2256 sampling points across ten universities and 13536 satellite-view images. Moreover, the test set consists of a query set, including 2331 UAV-view images from 777 sampling points across four universities, and 4662 satellite-view images. Additionally, the gallery set encompasses a total of 27297 images from 3033 sampling points, covering all fourteen universities.

IV. EVALUATION METRICS

In this section, we first introduce the widely used evaluation metric, i.e., Recall@K, in cross-view geo-localization. Then we present the proposed SDM@K metric, which considers both retrieval and positioning performance.

A. Spatial Discrete Index (Recall@K)

In image retrieval, Recall@K (R@K) [12], [13] is the most commonly used evaluation metric. Taking R@1 as an example, whether a sample is a correct match can be expressed as:

\[
I(l_q, l_i) = \begin{cases} 
1 & \text{if } l_q = l_i \\
0 & \text{if } l_q \neq l_i
\end{cases}
\]

(1)

where \(l_q\) corresponds to the category of query, and \(l_i\) is the category corresponding to the \(i\)-th image sorted in the ascending order of the calculated euclidean distance. The resulting value is 1 if it belongs to the same category and 0 if it is not. For all the samples, R@1 is defined as:

\[
R@1 = \frac{1}{|Q|} \sum_{q \in Q} I(l_q, l_i)
\]

(2)

where \(Q\) is the set of all query images and \(|Q|\) denotes the number of images in \(Q\). We can see that the value of R@1 increases when only the category of query and the category of the closest image in the gallery are the same, otherwise, they are regarded as false-matched samples.

B. Spatial Continuity Index (SDM@K)

During UAV operations, a notable characteristic arises from the spatial continuity and density of satellite imagery, resulting in subtle gaps in feature information between adjacent images. As depicted in Fig. 3 (a), using R@1 as the evaluation criterion would only consider the correct satellite images within the red dashed region, denoted as \(I(l_q, l_i) = 1\), while categorizing the satellite images within the blue dashed region as incorrect, indicated by \(I(l_q, l_i) = 0\). Even though the deviation is only 1 sampling point, R@1 will consider it as a wrong match. For UAV self-positioning, the objective is to achieve a closer approximation to the actual location hence minor positioning deviations are inevitable. However, large deviations are undesired, and such spatial differences are not effectively captured by the R@1 metric.

To address the above issue and provide a more accurate measure of positioning accuracy, we propose a new evaluation metric, namely Spatial Distance Metric (SDM). SDM combines the characteristics of Recall@K while also considering the performance of a model in positioning. Specifically, the SDM value of a single query sample is defined as:

\[
SDM_k = \frac{(K - k + 1)}{e^{s \times d_k}}
\]

(3)

where \(d_k = \sqrt{(x_q - x_i)^2 + (y_q - y_i)^2}\), \((K - k + 1)\) is the weight of the \(k\)-th sample, as shown in Fig. 3 (b). The weight is assigned based on the feature distance. A large weight is set for the gallery image that is closer to the query feature. \(K\) denotes the top \(K\) samples in the gallery that have the closest distance to the query features. \(x_q\) and \(x_i\) denote the longitude corresponding to the query and gallery images, respectively, while \(y_q\) and \(y_i\) denote the latitude. In short, \(d_k\) represents the spatial Euclidean distance between two images and \(s\) is an amplification factor. In this paper, \(s\) is set to \(5 \times 10^3\). Last, the SDM values for all \(K\) samples are calculated, and the final SDM@K index is obtained through a normalization process:

\[
SDM@K = \sum_{i=1}^{K} \frac{(K - i + 1)}{e^{s \times d_i}} / \sum_{i=1}^{K} (K - i + 1).
\]

(4)

It is worth mentioning that SDM is a continuous metric distributed between 0 and 1. A larger value indicates a better positioning performance.

SDM@K is tailored for the proposed UAV self-positioning task, which is a fused indicator for both retrieval and positioning tasks. In the real positioning process, we allow a little bit of error, but a large positioning error is not favored and thus should be considered as a false match. SDM can evaluate this well but does not define the error like the Euclidean distance. SDM is nonlinear but exponentially correlated. The reason for this design is that once the positioning error is large, SDM will give a score close to 0, which can be regarded as a false match. Specific visualization examples are shown in Fig. 3 (c). SDM is sensitive to small positioning errors. Once the error is large, as shown in the two cases in the bottom right of Fig. 3 (c), SDM is almost close to 0 and insensitive.

To further discuss the advantages of SDM@K, we analyze it from the perspectives of the sample score calculation method.
Fig. 3. A comparison between R@1 and SDM@K metric. In (b), \(d_1\) to \(d_K\) represent the spatial euclidean distance between the UAV-view image and the most similar K satellite-view images. The calculation is shown in Equation 4, where \(x\) represents the longitude and \(y\) represents the latitude. \(w_1\) to \(w_K\) are weighting factors and \(s\) is an amplification factor. In (c), we compare the evaluation process of R@1 and SDM@1.

Fig. 4. An intuitive comparison between Recall, Euclidean Distance and SDM. The image center is the completely matching position.

Fig. 5. The proportion of error offset sampling intervals located in the test set. 0 on the abscissa represents a completely correct match, 1 represents an offset of one sampling interval, 2 represents an offset of 2 sampling intervals, and the other indicates an offset of more than 3 sampling intervals. The sampling process can be referred to the upper part of Fig. 2.

Fig. 6. This section will introduce the proposed baseline network for UAV self-positioning, which mainly has four parts: Augment, Backbone, Head and Loss. As shown in Fig. 6, first, the model needs to receive the data of the UAV and satellite views as input. Then, the input should perform data enhancement through the Augment module. After the feature extraction step performed in Backbone, feature integration is used in the Head module and mapped to the specified feature space.
Fig. 6. An overview of the proposed framework, including both the training and testing phases. The training phase uses representation, metrics, and mutual learning to extract robust feature presentations. In the testing phase, matching is performed by feature cosine similarity. The right side of the figure is an intuitive explanation of the selection process of positive and negative samples in metric learning. We use the same position feature from another perspective as positive samples, and the rest as negative samples. The hard-mining strategy is applied to the negative sample pairs.

In the training phase, the feature dimension is mapped to the number of categories by the Fully Connected (FC) layer, and its probability distribution is calculated by the softmax operation. Last, a Loss function calculates through three types of supervision methods including representation learning, metric learning, and mutual learning. During inference, this feature is directly used to calculate the cosine similarity to rank the samples. It is worth mentioning that all the weights of the two branches are shared.

A. Data Augmentation

Given the high cost associated with collecting data using drones in real-world scenes, it becomes particularly meaningful to maximize the diversity of limited training data. Therefore, we adopted four data augmentation methods to expand the UAV and satellite view data.

1) UAV Flight Direction: The unpredictability of the UAV’s flight direction presents a challenge during data collection, as it is virtually impossible to capture all possible flight directions. In response to this issue, we propose a novel data augmentation strategy named “random rotate”, which is designed to simulate different UAV flight directions. Rather than simply rotating the entire original image, random rotate employs a more refined approach. In this method, the largest inscribed circle of the original image is identified as the cropping target. Based on the desired rotation angle \( \theta \), a square is extracted from within this circle as depicted in the top of Fig. 7. Random rotate not only expands the data in the UAV’s flight direction but also enhances the model’s generalization capability by obscuring edge information and allowing the model to focus on more generic features. Moreover, by erasing edge information, the model can prioritize central image information. The visualization results of three instances of random rotate are shown in the bottom-left corner of Fig. 7.

2) Satellite View Direction: Solely applying data augmentation to the UAV direction leads to a rapid convergence of the classification loss for satellite-view images to 100% during training, indicating a lack of diversity in the satellite data. In reality, the orientation of the satellite view is also uncertain. Then, we employ the random affine augmentation method for satellite-view data. Unlike random rotate, random affine introduces some empty areas caused by rotation at the image edges, which can introduce some edge noise, thus forcing the model to learn the spatial context information of the satellite imagery.

3) Light Intensity: Images captured by UAVs during different time periods and under varying weather conditions often display variations in brightness. To enhance the model’s robustness to illumination variations, we introduce the data augmentation method of random brightness. The visualization results are depicted in Fig. 7.

4) Spatial Differences: A big challenge in UAV self-positioning is the spatial difference caused by time offset. One of the solutions to this challenge is to perform random erasing on images. By randomly erasing part of an image, the object changes in space caused by time offset can be indirectly simulated, forcing the model to focus on more general and robust features.

B. Backbone Network

The choice of a proper backbone network plays a crucial role in extracting features from input images. The existing
backbones mainly include two categories: CNN-based and Transformer-based. In Section VI-E, the effects of different types of backbones on UAV self-positioning are experimentally investigated. Through our experiments, we observed a consistent phenomenon where the performance of the model trained with a CNN-based backbone network is noticeably inferior to that of the model trained with a Transformer-based backbone network. We attribute this discrepancy primarily to the significant challenges at the data level, including domain discrepancies between different perspectives, and variations in spatial information due to perspective and temporal differences. These factors contribute to the model’s demand for a strong capability to comprehend global context, which aligns well with the inherent characteristics of Transformer models. To strike a balance between performance and inference speed, we adopt the ViT-S model as the backbone network in our baseline model.

C. Prediction Head

The prediction head is responsible for integrating features and compressing them to a specific dimension. In this paper, we divide different prediction heads into two categories: pooling-based and chunking-based. In Section VI-F, we investigate and report the performance of different heads in UAV self-positioning.

1) Pooling-Based: By default, the output of the ViT model consists of two components: a global class token and other local tokens. Typically, the dimensions of the local tokens are represented as (B, N, C), where B denotes the batch size, N denotes the patch size, and C represents the number of channels. To represent an image with a single feature vector, it is necessary to compress the patch dimension (N) of the local tokens. Therefore, various pooling techniques, including MaxPool, AvgPool, AvgMaxPool, GlobalPool and GemPool [44], have been evaluated and the results are reported in Section VI-F.1.

2) Chunking-Based: In recent years, partitioning strategies have been widely used to extract fine-grained features and align regional features. One of the representative methods is Local Pattern Network (LPN) [32], which manually divides an image into multiple regions starting from the center and extending towards the boundary of an image. Originally, LPN was developed for CNN architectures. In this paper, we reconstruct the local tokens obtained from the ViT-S model into a (H, W) format, serving as the LPN head input. Additionally, we refer to a thermal-based adaptive partitioning method, FSRA [3], which is designed for ViT structures, eliminating the need for additional operations. Detailed analysis and comparison are presented in Section VI-F.2.

D. Loss Functions

The baseline model encompasses three distinct supervised learning methods: representation learning, metric learning, and mutual learning. These methods need to calculate losses at three different levels, including the class, feature, and distribution levels.

1) Representation Learning: Representation learning is a common and efficient way for feature extraction. In essence, representation learning maps data to a specified feature space through a supervision signal. The most common one is classification supervision, which has been widely used in cross-view geo-localization [45], face recognition [46], ReID [47], etc. For the baseline model, we use the widely adopted Cross-Entropy (CE) loss [18].

2) Metric Learning: During inference, image retrieval tasks often measure the similarity between images based on the distance between their features. The most commonly used measure is cosine similarity. Therefore, directly supervising the model at the feature level during training is essential. Indeed, metric learning is a method that maps similar samples to adjacent positions in the embedding space by learning the relative relationship between samples to achieve a more effective distance measurement method. Generally, metric learning is not used in isolation for supervised learning tasks, but rather in combination with representation learning methods. To evaluate the effectiveness of metric learning, we conduct experiments using some types of metric learning methods, including Contrastive Loss [48], the standard Triplet Loss [49], Hard-Mining Triplet Loss, Same-Domain Triplet Loss [3], and Soft-Weighted Triplet Loss [50].

Since metric learning naturally involves the selection of positive and negative samples, in this part, we sample the two perspectives in a paired form, that is, to ensure that the drone and satellite images input to the model in each iteration cycle are spatially corresponding. For each anchor, there must be a positive sample from another perspective and \((B - 1) \times 2\) negative samples, where \(B\) is the batch size. The details of the sampling process are shown in the right part of Fig. 6. In order to define the expression of metric learning more clearly, we list several typical metric learning methods as follows. The triplet loss is defined as:

\[
\text{Tri}(a, p, n) = \max(0, D(a, p) - D(a, n) + m)
\]

where \(a\) is the feature of the anchor sample, \(p\) is the positive sample feature, and \(n\) represents the negative sample feature. \(m\) is a predefined margin controlling the minimum distance between similar samples. \(D(a, b)\) denotes the cosine similarity between sample \(a\) and \(b\).

The hard-mining triplet loss function is calculated as:

\[
\text{HMTri}(a, p, n) = \max_{(a, p, n)} \left[ \text{Tri}(a, p, n) \right]
\]

where \(\max_{(a, p, n)}\) means maximizing all possible triples.

Last, the soft-weighted triplet loss function is defined as:

\[
\text{SWTr}(a, p, n) = \log(1 + e^{a \times (D(a, p) - D(a, p))})
\]

We use the soft-weighted triplet loss due to its excellent performance for the proposed task. A comparison of these loss functions is reported in Section VI-G.2.

3) Mutual Learning: Mutual learning has been widely applied to the field of knowledge distillation [51]. However, for the UAV self-positioning task, it is expected that the category vector distributions of the UAV and satellite views of the same category tend to be consistent. Therefore, we introduce the
distribution-level bilateral learning method, which is expressed as:

\[
KL\text{Loss} = KL\text{Div}(O_d || O_s) + KL\text{Div}(O_s || O_d) 
\]

(8)

\[
KL\text{Div}(O_p || O_q) = \sum_{i=1}^{N} O_p(i) \times \log\left(\frac{O_p(i)}{O_q(i)}\right) 
\]

(9)

where \(O_p\) and \(O_q\) respectively represent the probability distribution of the teacher and student category vectors through softmax. Additionally, \(O_d\) represents the class vector output of a UAV image, while \(O_s\) represents the class vector output of a satellite image.

VI. EXPERIMENTAL RESULTS

A. Implementation Details

The backbone networks used in this section were all pre-trained using the timm framework [52], with the additional classification layer removed. During training, the Stochastic Gradient Descent (SGD) optimizer was employed, with an initial learning rate of 0.003 and batch size of 8. The learning rate for the backbone network weights was set to 0.3 of the other weights. The models were trained for a total of 120 epochs, with the learning rate decreasing to 0.1 \times \theta_70\text{-th} and 0.01 \times \theta_{110\text{-th}} epochs, respectively. The input scale for both the UAV-view and satellite-view images was set to 224 \times 224. Regarding the network architecture, we use global representation vectors of 512 dimensions through a fully connected layer, followed by a classification layer for category prediction.

B. Non-Dense Features v.s. Dense Features

Previously, we discussed the significance of dense sampling for UAV self-positioning. In this section, we categorize the dataset into Dense and Non-Dense classes and validate it through experimentation. For the Non-Dense category, we utilize two non-intensive geographic task-related datasets, namely Place [53] for landmark classification tasks and University-1652 for cross-view geo-localization tasks. For the Dense category, we use the dataset proposed in this paper. The experimental results are shown in Table III, which demonstrate that the use of dense data during training has a significant impact on UAV self-positioning task. Correspondingly, the model trained in dense data can significantly improve the positioning performance. We attribute this improvement to the presence of overlapping areas between adjacent frames in the dense dataset, which compels the model to extract more robust spatial distribution information.

| Dataset          | R@1  | R@5  | S@D1 | S@D5 |
|------------------|------|------|------|------|
| Place [53]       | 1.88 | 3.22 | 4.65 | 2.80 |
| University-1652  | 28.77| 42.63| 33.94| 21.07|
| DenseUAV(ours)   | 80.18| 93.99| 84.39| 78.02|

TABLE III

A COMPARISON OF THE POSITIONING PERFORMANCE TRAINED WITH DENSE AND NON-DENSE DATASETS

C. Comparison With The Existing Methods

UAV self-positioning is a subtask within the broader context of geo-localization tasks. Thus the methodology closely aligns with various UAV-based geo-localization assignments, such as drone-view target localization and drone navigation. To facilitate a comprehensive evaluation of existing methods on the DenseUAV dataset, we have adopted some typical approaches utilized in UAV-related tasks for training and testing. The experimental results are reported in Table IV. Notably, conventional CNN-based networks exhibit subpar performance on the DenseUAV dataset which will be further analyzed in Session VI-E. Last, our baseline model exhibits excellent performance without additional module design.

In addition, we trained our baseline model on the University-1652 dataset to verify its generality capability. The experimental results are shown in Table V. Since the baseline model is not tailored for University-1652, the pure structure can not beat the specially designed methods like FSRA. However, when these specific modules are introduced into our baseline, there will be 1-2 points of improvement. This also demonstrates the effectiveness of our baseline structure.

D. Evaluation on Data Augmentation

It is crucial to create diverse scenarios that better align with real-world conditions. This section focuses on the impact of data augmentation techniques on UAV self-positioning tasks.

1) UAV Flight Direction: To validate the efficacy of our proposed random rotate, we conducted experiments as presented in the Random Rotate column of Table VI. The first row in the table represents the use of only random horizontal flip and resize operations at the data level. We can notice that incorporating random rotate data augmentation for the UAV view led to significant improvements across all the metrics.

| Method               | Drone→Satellite | Satellite→Drone |
|----------------------|-----------------|-----------------|
|                      | R@1            | R@5            | AP   | R@1            | R@5            | AP   |
| Instance Loss [1]    | 58.23%         | 62.91%         | 74.47%| 59.45%         | 65.03%         | 74.80%|
| RK-Net [55]          | 66.13%         | 70.23%         | 80.17%| 65.76%         | 79.89%         | 85.38%|
| LCM [54]             | 66.65%         | 70.82%         | 79.89%| 65.38%         | 79.14%         | 84.65%|
| LPN [32]             | 75.93%         | 79.14%         | 86.45%| 74.79%         | 83.63%         | 87.69%|
| PCL [4]              | 79.47%         | 83.63%         | 87.69%| 78.51%         | 88.25%         | 87.87%|
| FSRA [1]             | 82.25%         | 84.82%         | 87.87%| 81.53%         | 84.78%         | 87.59%|
| Our Baseline         | 82.22%         | 84.78%         | 87.59%| 81.49%         | 86.24%         | 87.87%|
| Our Baseline + FSRA  | 83.91%         | 86.24%         | 87.87%| 82.91%         | 86.24%         | 87.87%|

TABLE IV

COMPARISON OF RESULTS AFTER MIGRATING PREVIOUS UAV-BASED GEO-LOCALIZATION METHODS TO DENSEUAV FOR TRAINING

TABLE V

A COMPARISON OF OUR BASELINE MODEL AND SOME EXISTING METHODS ON UNIVERSITY-1652

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
DeiT-S, PVTV2-B2, Swinv2-T, ViT-S, and ViT-B are used as Transformer backbones. Note that the Swinv2-T model uses an input size of 256 × 256, and all the other networks use 224 × 224 inputs. All the above models are trained on the DenseUAV dataset. The weights of all models are pre-trained with Timm [52]. The experimental results are shown in Table VII, including the number of parameters (Params), calculation amount (Macs), inference time (InferTime, for 2000 samples), and retrieval accuracy (R@1 and R@5). In addition, the curve of SDM@K is plotted in Fig. 8 (a).

Based on the experimental results, the following conclusions are drawn. First, the Transformer-based models perform significantly better than the CNN-based models for UAV self-positioning. Second, ViT, the most primitive Transformer model, is more competitive. Last, in terms of both accuracy and inference speed, the ViT-S model reaches a good balance, with only 18.9s/2000 images in inference time.

It can be observed that there is a significant gap in positioning accuracy between CNN-based and Transformer-based methods. To further explore the advantages of Transformers, we visualize the heatmap results of four different types of backbone networks in Fig. 9. Due to its limited receptive field, ResNet-50 focuses mainly on salient feature regions, which is disadvantageous for UAV self-positioning since adjacent frames often have overlapping areas. By solely focusing on salient feature regions, it becomes difficult to achieve accurate self-positioning results. Unlike ResNet, ConvNeXt, with the addition of large receptive field convolutions in its network, can be attributed to the random affine is not well-suited for accurately capturing the real scene of drone rotation.

3) Light Intensity: The impact of light intensity has also been evaluated in the Random Brightness column of Table VI. The results demonstrate that random brightness has minimal impact on the results. This can be attributed to the random time-based data collection, which already encompasses data samples with varying brightness levels. It can also be concluded that lighting is not the main challenge and factor affecting this task.

4) Spatial Difference: The impact of spatial difference has also been evaluated in the Random Erasing column of Table VI. On the whole, random erasing generally improves positioning accuracy. In particular, the effect on satellite view is more significant, increasing R@1 by about 2.5% and SDM@1 by about 2.5%. This is mainly due to the fact that random erasing can create a scene where information is lost in space, which can help the model effectively overcome the challenge of spatial information changes caused by time offsets.

E. Evaluation on Backbone Network

To explore the impact of different backbone networks on the UAV self-positioning task, we adopt some popular backbone networks for experiments, mainly containing two categories: CNN-based and Transformer-based. Among them, the ResNet50, SENet50, EfficientNet-B3, EfficientNet-B5, and ConvNeXt-T models are used as CNN backbones. DeiT-S, PVTV2-B2, Swinv2-T, ViT-S, and ViT-B are used as Transformer backbones. Note that the Swinv2-T model uses an input size of 256 × 256, and all the other networks use 224 × 224 inputs. All the above models are trained on the DenseUAV dataset. The weights of all models are pre-trained with Timm [52]. The experimental results are shown in Table VII, including the number of parameters (Params), calculation amount (Macs), inference time (InferTime, for 2000 samples), and retrieval accuracy (R@1 and R@5). In addition, the curve of SDM@K is plotted in Fig. 8 (a).

Based on the experimental results, the following conclusions are drawn. First, the Transformer-based models perform significantly better than the CNN-based models for UAV self-positioning. Second, ViT, the most primitive Transformer model, is more competitive. Last, in terms of both accuracy and inference speed, the ViT-S model reaches a good balance, which achieves 80.18%/84.39% in R@1/SDM@1, with only 18.9s/2000 images in inference time.

It can be observed that there is a significant gap in positioning accuracy between CNN-based and Transformer-based methods. To further explore the advantages of Transformers, we visualize the heatmap results of four different types of backbone networks in Fig. 9. Due to its limited receptive field, ResNet-50 focuses mainly on salient feature regions, which is disadvantageous for UAV self-positioning since adjacent frames often have overlapping areas. By solely focusing on salient feature regions, it becomes difficult to achieve accurate self-positioning results. Unlike ResNet, ConvNeXt, with the addition of large receptive field convolutions in its network.
Building on the analysis of the aforementioned results, we have derived several key factors contributing to the significant improvement of the Transformer-based method in comparison to the CNN-based approach: 1) While a CNN-based method primarily focuses on local salient features, Transformers demonstrate a more uniform distribution of attention across salient features. This allows Transformers to allocate attention to sub-salient information, aiding in the extraction of hidden feature details. 2) The UAV self-positioning task involves both retrieval and positioning aspects, demanding the model’s attention not only to fine-grained information but also to the establishment of spatial concepts. The heat map analysis indicates that a Transformer-based model appears to grasp the significance of spatial information, particularly emphasizing the center of the image, aligning with the original task settings. 3) Another plausible factor contributing to the success of Transformers is their reliance on global modeling capabilities, enabling it to obtain more robust representations that overcome domain and perspective differences. This is reinforced by the experiments in [3], demonstrating the Transformer’s superior performance over CNN-based methods in UAV-based geo-localization tasks.

F. Evaluation on Prediction Head

The role of a prediction head is to integrate and map the features extracted from the backbone. This paper divides the heads into two groups. The first one is based on pooling methods, and the other one is based on some existing chunking methods. Since the chunking-based method has a limited convergence speed under the condition of the default learning rate, to present the results more fairly, the learning rate of the chunking-based method is set to 0.01 (the default is 0.003). In addition, all the experiments in this section use the ViT-S model as the backbone network. The results are reported in Table VIII and Fig. 8 (b), including the parameters, computation, inference time, and accuracy of different heads.

1) Pooling-Based Heads: Table VIII shows the results of different heads. Among them, MaxPool indicates that the 1D max pooling operation is performed in the patches dimension of local tokens. Similarly, AvgPool represents a 1D average pooling operation. AvgMaxPool is the sum of the results of MaxPool and AvgPool. Global means just using the global class token. GemPool means to adaptively learn weights for local tokens through learnable parameters. Global is the baseline model used in this paper, which has a certain improvement as compared with AvgPool and MaxPool. Due to its adaptive learning method, GemPool has improved R@1 by 2.3% and SDM@1 by 0.6% as compared with the baseline method.

2) Chunking-Based Heads: The main idea of the chunking-based method is to achieve feature alignment between images from different perspectives by blocks. To maintain alignment, the number of blocks is uniformly set to 2. As shown in Table VIII, compared with the baseline, both LPN and FSRA have improved the final performance significantly. Among them, LPN has improved R@1 by 2.9%, and SDM@1 by 1.8%. However, FSRA and LPN increase the inference time slightly.

| Head          | Params | InFeTime | R@1 | R@5 |
|---------------|--------|----------|-----|-----|
| MaxPool       | 23.3M  | 18.9s    | 65.21% | 82.45% |
| AvgPool       | 23.3M  | 19.0s    | 79.62% | 92.96% |
| AvgMaxPool    | 23.3M  | 19.3s    | 77.95% | 93.35% |
| GlobalPool    | 23.3M  | 18.9s    | 80.18% | 93.95% |
| GemPool [31]  | 23.3M  | 19.4s    | 82.45% | 94.21% |
| FSRA(Block=2) [3] | 26.0M | 21.1s    | 82.58% | 94.94% |
| LPN(Block=2) [32] | 26.0M | 21.2s    | 83.05% | 94.99% |
G. Evaluation on Loss Function

Our baseline includes three supervised methods: Representation Learning, Metric Learning, and Mutual Learning.

1) Representation Learning: In the experiments, we evaluate two commonly used classification losses: Cross-Entropy (CE) Loss and Focal Loss [64]. The results are shown in Table IX. Although focal loss provides advantages in terms of sample and weight balancing, we find that the CE loss is more suitable for the task of UAV self-positioning.

2) Metric Learning: Metric learning serves the enhancement of the discriminative capability of the model and mitigates modality discrepancies. To verify the effectiveness of metric learning, we perform experiments using five variations of losses: Contrastive Loss, Triplet Loss, Hard-Mining Triplet Loss, Same-Domain Triplet Loss, Soft-Weighted Triplet Loss. As shown in Table IX, compared to using only representation learning, the introduction of metric learning can achieve significant improvements. Specifically, the introduction of contrastive loss improved the overall indicators. The inclusion of triplet loss leads to a noticeable improvement of 2.9% in R@1 and an increase of 2.4% in SDM@1. Furthermore, the soft-weighted triplet loss achieves an additional performance boosting of 2% in R@1 and 1.6% in SDM@1, as compared with the standard triplet loss.

It is worth discussing the advantages of triplet loss over pairwise supervised contrastive loss for the UAV self-positioning task. In some specific fields such as image segmentation [48], pixel-level contrast supervision can achieve good results. However, the UAV self-positioning task is limited by the volume of data. Making full use of information from different samples for supervision can improve the convergence speed and generalization capability of the model. The form of triplet can better learn the relative relationship between samples, rather than just judging whether two samples are similar.

3) Mutual Learning: Kullback-Leibler (KL) divergence is primarily applied in knowledge distillation scenarios, particularly in teacher-student learning settings. In this paper, we adopt a bidirectional mutual learning approach, where two branches exchange and enhance their knowledge and representations. The mutually supervised learning approach aids in ensuring that the UAV and satellite branches progress simultaneously and mutually influence each other. In our experiment, we introduce KLLoss based on the CE loss and soft-weighted triplet loss. As shown in Table IX, the use of KLLoss leads to further performance boosting, i.e., 1.8% in R@1 and 1.2% in SDM@1.

H. The Impact of Data Source

There are many factors that affect the performance of the UAV self-positioning task. Therefore, this section will combine experiments and analyze the impact of data sources, mainly including flight height, the scale, and the time node of the satellite image.

1) Impact of Flight Altitude: The altitude at which a UAV operates directly impacts the visible range of its captured UAV-view images. Furthermore, increasing the flight altitude can expand the field of view, but it also leads to a larger spatial distance represented by each pixel after image resizing, resulting in a loss of fine-grained information. To investigate the impact of various flight altitudes, we conduct experiments by using the captured UAV images at heights of 80m, 90m, and 100m.

From the experimental results, we can see that with the increase of the flying height of the drone, the indicators i.e., R@1 and SDM@1 show an increasing trend, mainly because the flying height of the drone can make the image contain a wider field of view, that is, contain more abundant spatial information. However, due to the limited flying height of drones, we were unable to collect data at higher altitudes. The impact of higher UAV flight altitudes on positioning performance still needs to be further explored.

2) Impact of The Scale: Intuitively, the scale of the satellite image determines the amount of information contained in the satellite-view images. If the scale is too small, it will increase the difficulty of matching, and if the scale is too large, it will increase the probability of wrong matching. Multi-scale satellite imagery can help the model to improve its robustness to scale changes in the UAV input. To explore the influence of satellite-view scale on UAV self-positioning, this section conducts comparative experiments on satellite images of different scales, and the results are shown in the Scale column of Table X. Among them, the definition of Small, Middle, and Big is as follows. If Middle is used as the base scale (B, B), the scale of Small is (0.75B, 0.75B), and the scale of Big is (1.25B, 1.25B). First, it can be found that in the single scale set, the middle scale achieves the best performance, and the performance of the small and big are similar, which is lower than middle-scale by nearly 3 points.
TABLE X

| Height | Scale | Time | R@1 | R@5 | SDM@1 | SDM@3 | SDM@5 | SDM@10 |
|--------|-------|------|-----|-----|-------|-------|-------|--------|
| 80m    | Small | 2020 | 79.70% | 94.33% | 84.07% | 82.83% | 78.62% | 67.19% |
| 90m    | Middle| 2020 | 80.33% | 93.15% | 84.12% | 81.85% | 77.26% | 65.61% |
| 100m   | Big   | 2020 | 80.82% | 94.63% | 85.25% | 83.10% | 78.49% | 66.77% |
|        |       | 2022 | 76.36% | 95.40% | 81.42% | 69.90% | 55.79% | 40.26% |
|        |       | 2022 | 79.28% | 96.57% | 83.41% | 68.36% | 56.81% | 40.90% |
|        |       | 2022 | 76.49% | 96.05% | 81.55% | 66.33% | 55.12% | 39.53% |
|        |       | 2022 | 80.57% | 96.98% | 84.52% | 79.24% | 72.32% | 57.21% |
|        |       | 2022 | 79.15% | 94.81% | 83.61% | 78.89% | 71.97% | 56.94% |
|        |       | 2022 | 80.18% | 93.99% | 84.39% | **82.51%** | **78.02%** | **66.46%** |

in R@1. This is mainly because middle-scale images contain a moderate amount of spatial scale, which is more suitable for query images. Then, we can find that multi-scale has stronger performance than single-scale universally. Specifically, the simultaneous use of the total three scales, as compared to the use middle scale merely, results in a 0.9% improvement in R@1 and 1.0% in SDM@1. This also makes us think about a problem. In real landing application scenarios, the flying height of the UAV is unknown, so the satellite image library should contain images of as many scales as possible to make up for the uncertainty of the UAV’s flying height.

3) Impact of The Time: The spatial difference arising from time offset poses a significant challenge in UAV self-positioning. To enhance the model’s robustness to this type of difference, this paper constructed the dataset using satellite-view images from two different years (2020 and 2022), while the UAV-view images were acquired in 2021. As shown in the Time column of Table X. The experimental results demonstrate that incorporating multi-time spans yields better results compared to a single-time node. When compared to a single time node, R@1 improves by nearly 9%, and SDM@1 increases by approximately 7 points. These results highlight the importance of considering multi-time spans in addressing the spatial difference caused by time offset. By including satellite images from different years, the model becomes more robust to temporal variations and exhibits improved performance in UAV self-positioning tasks.

I. Visualization

This subsection visualizes the positioning results of the baseline model from the sample level. Fig. 10 illustrates the retrieval results of 4 sample groups within the test set. The UAV-view images (query) are depicted on the left side of the dotted line, while the 6 images closest to the query image in the satellite view are displayed on the right side. The true-matched images are denoted by orange boxes, while the false-matched images are represented by green boxes. Notably, satellite images encompass a span of time, leading to substantial domain variations, whereas UAV images exhibit discrepancies in brightness and shooting height due to temporal inconsistencies. As depicted in the figure, there is a noticeable divergence in both domain and spatial information between UAV and satellite images, posing a significant challenge in this task. Although this paper adopts the method of metric learning and mutual learning to alleviate the matching difficulties caused by domain differences. Some queries still cannot match the corresponding satellite images well. Perhaps some perspective transformation and GAN methods can also be used to alleviate this domain difference.

J. Limitations and Analysis

This paper proposes a novel dataset and a baseline model for UAV self-positioning. However, our work still has certain limitations that should be considered for future studies. We analyze these limitations from the data and methodology perspectives below.

1) Data: The proposed DenseUA dataset is restricted to low-altitude urban environments. In the future, more scenes are worth exploring to enhance the robustness of UAV self-positioning tasks. Such scenes include foggy, rainy, snowy, nights, high altitudes, mountains, rainforests, maritime scenes, etc. By this, a trained model is not limited to a specific scenario or pattern, thereby improving its versatility in various
scenarios. However, the cost of collecting more data from various real-life scenes is very high. Some recently proposed image generation methods could be considered for effective data collection [65].

2) Methodology: We analyze the possible attempts in terms of methodology based on the challenges of the UAV self-positioning task.

- Image interference from adjacent spatial locations. As shown in Fig. 11 (a), DenseUAV is a dense sampling dataset. There are overlapping regions between adjacent frames. So it is necessary to not only fully mine fine-grained information but also improve the representation ability of relative spatial positions.

- Spatial inconsistency caused by inconsistent sampling times between drones and satellites. As shown in Fig. 11 (b), with the migration of time, there are certain changes in the same location, which requires a model to have strong global information utilization capabilities and is good at capturing static fine-grained features.

- Domain differences between drone and satellite images, such as dynamic object changes, perspective shifts, weather changes, lighting and shadows, etc. As shown in Fig. 11 (c), the first example illustrates that dynamic objects such as cars and people have changed due to inconsistent sampling times between drone and satellite images. The second example illustrates information differences due to weather. The third example is to illustrate the loss of information caused by shadows and lighting conditions. The fourth example shows that perspective bias leads to inconsistency in spatial information. This requires a model to selectively ignore some fine-grained features and focus on static objects. It also requires a model to capture global spatial information and establish strong correlations with relevant spatial information.

The proposed baseline method is not specifically designed to address the above challenges. In the future, we hope some targeted methods can be designed to solve these challenges.

VII. CONCLUSION

This study investigated the vision-based self-positioning task of UAVs in low-altitude urban environments. Firstly, we introduced DenseUAV, a novel UAV-based geo-localization dataset, which incorporates real-world scene acquisition and dense sampling. This dataset aims to establish a strong pipeline between UAV high-precision self-positioning and real-world applications. Furthermore, we proposed a new evaluation metric, SDM@K, which offers enhanced effectiveness in evaluating positioning accuracy within real-world scenarios. Additionally, we developed a robust baseline model and a practical process framework for practical applications. At the model level, we partitioned the model into four major components and employed three learning methods for supervision. We conducted a wide spectrum of experiments to validate their effectiveness. Moreover, we conducted experiments focusing on the data source to investigate the impact of flight altitude, satellite image scale, and satellite image time on the positioning results. Last, our proposed baseline model achieved remarkable results on the DenseUAV dataset. It is worth mentioning that due to the challenges posed by modality differences between UAV and satellite data, as well as the characteristics of UAV self-positioning task, we have observed that the Transformer models outperform CNNs significantly.

REFERENCES

[1] Z. Zheng, Y. Wei, and Y. Yang, “University-1652: A multi-view multi-source benchmark for drone-based geo-localization,” in Proc. 28th ACM Int. Conf. Multimedia, Oct. 2020, pp. 1395–1403.

[2] S. Hu and G. H. Lee, “Image-based geo-localization using satellite imagery,” Int. J. Comput. Vis., vol. 128, no. 5, pp. 1205–1219, May 2020.

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

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

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

[6] H. Yu et al., “The unmanned aerial vehicle benchmark: Object detection, tracking and baseline,” Int. J. Comput. Vis., vol. 128, no. 5, pp. 1141–1159, May 2020.

[7] J. Leng, M. Mo, Y. Zhou, C. Gao, W. Li, and X. Gao, “Pareto refocusing for drone-view object detection,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 3, pp. 1320–1334, Mar. 2023.

[8] P. Zhu, J. Zheng, D. Du, L. Wen, Y. Sun, and Q. Hu, “Multi-drone-based single object tracking with agent sharing network,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 10, pp. 4058–4070, Oct. 2021.
L. Liu and H. Li, “Lending orientation to neural networks for cross-view geo-localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 3961–3969.

L. Liu and H. Li, “Lending orientation to neural networks for cross-view geo-localization,” in Proc. IEEE/ICCV Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 5617–5626.

R. Zhu, L. Yin, M. Yang, F. Wu, Y. Yang, and W. Hu, “SUES-200: A multi-height multi-scene cross-view image benchmark across drone and satellite,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 9, pp. 4825–4839, Sep. 2023, doi: 10.1109/TCSVT.2023.3249204.

R. Zhao, W. Ouyang, and X. Wang, “Person re-identification by salience matching,” in Proc. IEEE Int. Conf. Comput. Vis., Dec. 2013, pp. 2528–2535.

L. Zheng, L. Shen, L. Tian, S. Wang, J. Wang, and Q. Tian, “Scalable person re-identification: A benchmark,” in Proc. IJCV, 2015, pp. 1116–1133.

S. Zhu, T. Yang, and C. Chen, “VIGOR: Cross-view image geo-localization beyond one-to-one retrieval,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 5316–5325.

Y. Tian, C. Chen, and M. Shah, “Cross-view image matching for geo-localization in urban environments,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1998–2006.

T. Yin, Y. Li, and Y. Jia, “Learning deep representations for ground-to-aerial geo-localization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2015, pp. 5007–5015.

N. N. Vo and J. Hays, “Localizing and orienting street views using overhead imagery,” in Proc. ECCV, 2016, pp. 494–509.

H. Xu, M. Yang, L. Deng, Y. Qian, and C. Wang, “Neutral cross-entropy loss based unsupervised domain adaptation for semantic segmentation,” in Proc. CVPR, vol. 30, pp. 4516–4525, 2021.

H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, “End-to-end comparative attention networks for person re-identification,” IEEE Trans. Image Process., vol. 26, no. 7, pp. 3492–3506, Jul. 2017.

Z. Zhao, L. Wang, S. Liu, and R. Ji, “Quadreplet loss for deep hashing,” in Proc. ACM ICMLR, 2019, pp. 224–232.

K. Sohn, “Improved deep metric learning with multi-class N-pair loss objective,” in Proc. NIPS, 2016, pp. 1857–1865.

Y. Deng, M. H. Nguyen, and G. H. Lee, “CVM-Net: Cross-view matching network for image-based ground-to-aerial geo-localization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 7258–7267.

Y. Guo, M. Choi, K. Li, F. Boussaid, and M. Bennamoun, “Soft exemplar highlighting for cross-view image-based geo-localization,” IEEE Trans. Image Process., vol. 31, pp. 2094–2105, 2022.

T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “SimCLR: A simple framework for contrastive learning of visual representations,” in Proc. NIPS, 2020, p. 4.

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

Y. Sun et al., “Circle loss: A unified perspective of pair similarity optimization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 6397–6406.

Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, “Optimal feature transport for cross-view image geo-localization,” in Proc. AAAI Conf. Artif. Intell., 2020, vol. 34, no. 7, pp. 11990–11997.

Y. Shi, X. Yu, D. Campbell, and H. Li, “Where am i looking at? Joint location and orientation estimation by cross-view matching,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 4063–4071.

A. Toker, Q. Zhou, M. Maximov, and L. Leal-Taixé, “Coming down to earth: Satellite-to-street view synthesis for geo-localization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 6484–6493.

B. Sun, C. Chen, Y. Zhu, and J. Jiang, “GEOCAPSNET: Ground to aerial view multi-height geo-localization using capsule network,” in Proc. ICME, 2019, pp. 742–747.

R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “NetVLAD: CNN architecture for weakly supervised place recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 5297–5307.

T. Wang et al., “Each part matters: Local patterns facilitate cross-view geo-localization,” IEEE Trans. Circuits Syst. Video Technol., vol. 32, no. 2, pp. 867–879, Feb. 2022.
[57] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.

[58] M. Tan and Q. Le, “EfficientNet: Rethinking model scaling for convolutional neural networks,” in Proc. ICML, vol. 97, 2019, pp. 6105–6114.

[59] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, “A ConvNet for the 2020s,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 11966–11976.

[60] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jegou, “Training data-efficient image transformers & distillation through attention,” in Proc. Int. Conf. Mach. Learn., vol. 139, 2021, pp. 10347–10357.

[61] W. Wang et al., “Pyramid vision transformer: A versatile backbone for dense prediction without convolutions,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 548–558.

[62] Z. Liu et al., “Swin transformer v2: Scaling up capacity and resolution,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 11999–12009.

[63] Y. Sun, T.-S. Chua, Y. Yang, and C. Yan, “Multiple-environment self-adaptive network for aerial-view geo-localization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 1–22.

[64] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 1–22.

[65] T. Wang, Z. Zheng, Y. Sun, T.-S. Chua, Y. Yang, and C. Yan, “Multiple-environment self-adaptive network for aerial-view geo-localization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 1–22.

Ming Dai received the B.S. degree in automation and the M.S. degree in mechanical engineering from China Jiliang University, Hangzhou, China, in 2020 and 2023, respectively. Currently, he is pursuing the Ph.D. degree with the School of Automation, Southeast University. His primary research interests include image retrieval and object detection.

Enhui Zheng received the Ph.D. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 2006. He is currently an Associate Professor with the Department of Automation, China Jiliang University, Hangzhou. He is also the Deputy Secretary-General of the Zhejiang Model Radio Sports Association. His research interests include UAV application, image processing, deep learning, and simultaneous localization and mapping.

Zhenhua Feng (Senior Member, IEEE) received the Ph.D. degree from the Centre for Vision, Speech and Signal Processing (CVSSP), University of Surrey, U.K., in 2016. He is currently a Senior Lecturer in machine learning and computer vision with the School of Computer Science and Electronic Engineering, University of Surrey. He has published more than 80 scientific papers in top-tier conferences and journals, such as IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, International Journal of Computer Vision, CVPR, ICCV, IJCAI, and ACL. His current research interests include pattern recognition, machine learning, and computer vision. He has received the 2017 European Biometrics Industry Award from the European Association for Biometrics (EAB) and the 2018 AMDO Best Paper Award for Commercial Applications. He also served as the Program Chair for the 2022 BMVC, the Area Chair for BMVC from 2021 to 2023 and CVMP from 2022 to 2023, and a Senior Program Committee Member for the 2021 IJCAI. He serves as an Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS and Complex and Intelligent Systems.

Lei Qi received the Ph.D. degree from the Department of Computer Science and Technology, Nanjing University, in 2020. He is currently an Assistant Researcher with the School of Computer Science and Engineering, Southeast University, China. His research interests include some ML methods, such as domain generalization, domain adaptation, and semi-supervised learning. For applications, he mainly focuses on person re-identification and semantic segmentation.

Jiedong Zhuang received the B.S. degree in automation and the M.S. degree in control engineering from China Jiliang University, Hangzhou, China, in 2019 and 2022, respectively. Currently, he is pursuing the Ph.D. degree with the School of Information and Electronic Engineering, Southeast University. His research interests mainly include model pruning, pedestrian re-identification, and image retrieval.

Wankou Yang (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the School of Computer Science and Technology, Nanjing University of Science and Technology (NUST), China, in 2002, 2004, and 2009, respectively. From July 2009 to August 2011, he was a Postdoctoral Fellow with the School of Automation, Southeast University, China, where he has been an Assistant Professor with the School of Automation, since September 2011. His research interests include pattern recognition, computer vision, and machine learning.