A Scene Graph Encoding and Matching Network for UAV Visual Localization

Ran Duan, *Member, IEEE,* Long Chen, Zhaojin Li, Zeyu Chen, Bo Wu*,

***Abstract*—This paper tackles the visual localization of unmanned aerial vehicles (UAVs) in the presence of multi-source and cross-view images are involved. We present a lightweight end-to-end scene graph encoding and matching network that finds the best matches for the airborne camera views from the reference image maps. The scene graph addresses the challenges of encoding the semantic scene by aggregating the image convolutional features into global and structured semi-global descriptors. The principal contributions of this paper are as follows: First, we develop a new network architecture that embeds a non-local block and a modified vector of locally aggregated descriptors network (NetVLAD) into a backbone convolutional neural network (CNN). The main component of the modified NetVLAD is a cluster similarity masking graph (CSMG) encoder, which is proposed to replace the feature-cluster residuals computing in NetVLAD with cluster consensus feature aggregation and structure-aware scene graph extraction. In addition, a global descriptor is extracted by a non-local block to label each image with a discriminative global feature descriptor. Second, we develop a new triplet loss for the network training procedure to learn the features at different semantic levels. The proposed global descriptor and CSMG encoder are trained together according to a weighted sum of cosine triplet losses. Third, the global descriptor from the non-local block and semi-global descriptor from the CSMG encoder work hierarchically for coarse-to-fine image retrieval and can achieve real-time efficiency and favorable accuracy of image searching and matching from the reference image map. We train and test the model on two challenging benchmark datasets. We also test the pre-trained model on a dataset collected by a Fixed-wing UAV to further evaluate the model's generalizability. The benchmark evaluations and ablation experiments show that the developed method outperforms state-of-the-art methods and achieves superior performance in the real-time matching of UAV images and reference image maps for UAV visual localization. Open-source code is available on GitHub: https://github.com/rduan036/scene-graph-matching-demo.git.**

***Index Terms*—Scene graph, end-to-end network, image matching, UAV visual localization**

I. Introduction

Unmanned aerial vehicles (UAV) visual localization is essential for a range of remote sensing applications, including aerial mapping, surveying, and environmental monitoring. Determining the location through autonomous image georeferencing, also known as visual place retrieval (VPR) in the computer vision and robotics fields, is an emerging topic in smart geo-informatics that aims to find the best match of a scene in pre-built geo-reference maps. The most fundamental task is to encode the image features into unique descriptors and perform similarity measures. The more local details the image descriptor contains, the more accurate the image matching is. While significant progress has been made in recent years, mainly driven by deep learning technology, visual localization in geoscience and remote sensing applications remains a challenge, particularly on edge platforms such as UAVs [1]–[6]. The outputs of these methods are usually fixed-length vectors that can be efficiently matched using standard distance metrics, making them suitable for large-scale place recognition. The efficient recall of the scene and accurate matching between the query–reference image pair require algorithms to encode the image with discriminative models while possessing appropriate generalization capability to deal with feature variation caused by the multi-source and cross-view data. Thus, accurate place retrieval based on multi-source and cross-view imagery is associated with several challenges. For instance, many UAVs require the capability to obtain geo-location feedback by matching the onboard views to satellite imagery or pre-recorded map data. The query and reference images usually feature considerable variations in terms of the landscape, lighting condition, and angle of view.

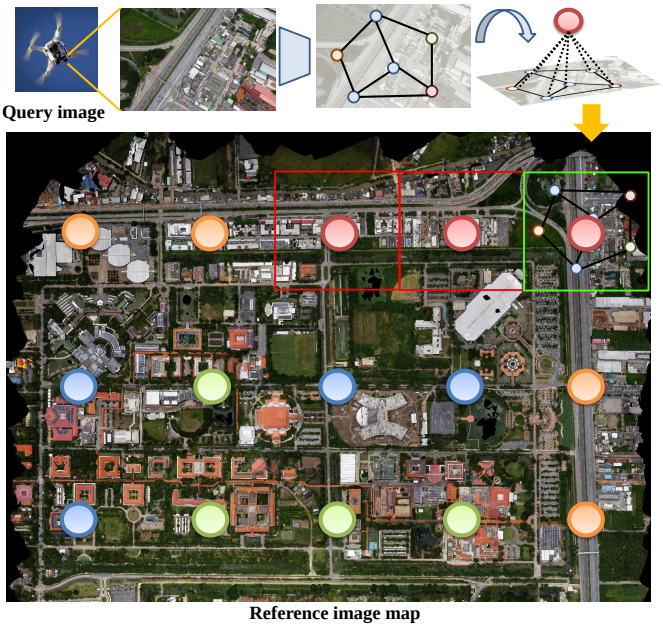


Fig. 1. Retrieval image obtained via graph-based feature aggregation. The hierarchical structure allows the graph to have both global and semi-global interpretations of the image features, which benefit both efficient recall and accurate matching of images.

The authors are with the Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Kowloon, Hong Kong, P.R. China.

*Corresponding author (bo.wu@polyu.edu.hk).

In this work, we address one of the critical issues of UAV visual localization, i.e., the VPR problem under multi-source, cross-view conditions, by encoding the image into a scene graph with both semi-global and global features for UAV-based accurate and efficient place retrieval. The concept is illustrated in Fig. 1. When drones are used to align captured scenes with satellite imagery, the task becomes challenging due to alterations in scene details. These alterations are often caused by spatiotemporal environmental changes, including image rotation and geometric distortion resulting from the drone's shooting angle. Current state-of-the-art (SOTA) methods, which employ image retrieval frameworks based on global descriptors, often interpret these as significantly different scenes due to the lack of local feature matching. However, introducing local features could dramatically increase the algorithmic complexity of image retrieval and matching. Moreover, local features, which are highly susceptible to changes in image details, often lack generalization ability, leading to network overfitting. As a result, most SOTA methods opt for global descriptors. The motivation behind this research is to improve matching accuracy by designing a type of local descriptor that not only maintains the network's generalization ability but also ensures efficient matching.

Local descriptors [7]–[11] are pixel-level descriptions containing detailed local appearance and spatial locations. These local descriptors can help build a discriminative model for each image, but the encoding ability could be insufficient for large-scale image data, and retrieving them is even more problematic. The use of local descriptors for the entire image would cause overfitting, which would be disastrous for the retrieval process. Considering that descriptors at a low semantic level usually lead to matching ambiguity, the geometric consensus of local descriptors can guide the rejection of mismatches during image-to-image matching. Moreover, the global descriptor represents the entire image scene at a high semantic level as a generative model, which can significantly reduce the retrieval workload [12]–[14]. However, this encoding may cause a loss of the spatial information of the image, which is useful for differentiating between similar scenes. Considering the above issues, it is crucial to design an appropriate feature aggregation strategy that balances the discriminative character of the local descriptors and the generative character of the global descriptors of an image in a specific application. A notable work [15] presented a non-local block that adopts a self-attention mechanism at different semantic levels to bridge the local features with the global semantic understanding of the image. The versatile network architecture is embedded into the existing network without changing the dimension between the last and next layers.

The vector of locally aggregate descriptors (VLAD) [16] is a global descriptor that accumulates the residuals between each local feature and the centroid of each group. The final representation is achieved by concatenating the descriptor of each cluster. NetVLAD [17] is a notable method in this context. It is a CNN architecture that integrates the VLAD descriptor into the network as a trainable layer, which frames the clustering problem in VLAD as a learnable classification problem and extracts the high-level deep features from a CNN. However, these algorithms treat each local feature equivalently, which often struggle with changes in viewpoint, illumination, and environmental conditions. This suppresses the most desirable features and amplifies some unsatisfactory features, which may exacerbate the matching ambiguity.

Here, we propose an approach that integrates non-local neural networks with a cluster similarity masking graph (CSMG) encoder modified from NetVLAD to extract both semi-global and global features and propose a scene graph along with a hierarchical matching strategy for the place retrieval task. The approach involves extracting a group of semi-global descriptors using non-local blocks and clustering them into hypernodes of a graph. The convolutional neural network (CNN) features that fall into the same cluster are denoted as consensus features. The spatial information of the consensus features indicates the spatial distribution of a certain semantic scene on the image. In the matching step, the node with the highest semantic level is used for fast filtering of the candidates. Then, structure-aware graph matching is applied according to corresponding semi-global descriptors. Overall, our contributions are as follows:

- A new end-to-end network architecture that integrates a backbone CNN, a non-local block for global feature encoding, and a CSMG encoder that replaces the NetVLAD output by aggregating semi-global features into a structure-aware scene graph.
- A hierarchical coarse-to-fine graph matching strategy for efficient image retrieval.
- Performance demonstration and ablation study via model evaluations on two challenging place recognition benchmarks and one UAV image dataset.

II. Related Work

Autonomous image georeferencing using image matching, either based on 2D images or 3D point clouds (including SAR) [18]–[25], [25], [26], is often performed through i) mathematical descriptor generation, ii) reference image search and similarity score computation, and iii) prior knowledge check and retrieval of image output [27]. As both the descriptor construction and reference image search are resource-demanding and time-consuming, deploying the VPR algorithms on mobile devices to achieve real-time tasks is still challenging. One study proposed hierarchical localization [6], [28], which leverages a global scene descriptor to achieve rough localization and is refined by the descriptors of the local features. Benefiting from the prior point cloud, the representative images of the locations are extracted and clustered, and the query images could be roughly grouped using the k-nearest neighbor algorithm. However, how to simultaneously generate the descriptors for reference images and query images is still an open question.

Scene descriptor generation involves extracting high-level visual features to better depict each segment of an image [29], [30]. Different from the local keypoint descriptors, the scene graph utilizes the information on the whole image to extract the mathematical visual descriptors. Before the advent of deep learning, the aggregated local features were mainly dominated

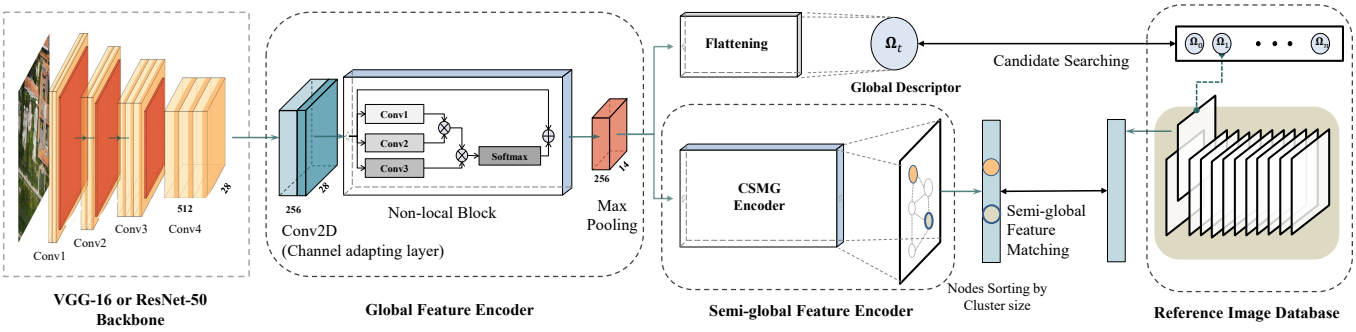


Fig. 2. Method overview. The scene graph consists of a global descriptor and a structure-aware scene graph. The global descriptor represents the global semantic features, while the graph contains semi-global features with encoded spatial relations. The output of the backbone network is first processed by the global descriptor encoder so that the local features extracted by the convolutional layer are associated with the global semantic information. The descriptor of a global descriptor is directly obtained from the flattened non-local feature. The descriptors of each node of the scene graph are then extracted by the CSMG encoder, which is modified from NetVLAD. Details of the modification are given in Fig. 3.

by handcrafted features such as SIFT, SURF, and AKAZE [7], [8], [27], [31], which have now been replaced by learned features. SuperPoint, a milestone in learned-feature methods, has been verified to be capable of recording deep features and thus resilient to illumination or seasonal changes [32]–[34]. However, the deep features extracted using a CNN are limited by the perspective field, making each feature unrepresentative of its neighborhood. Therefore, the non-local neural network [15], [35] has been proposed to integrate the information across a long distance to refine the original features to a non-local feature. The network provides insight into the method of extending the local receptive field to the upper layer by weighting the features at each position according to the product similarity of features.

NetVLAD is a popular baseline method for place recognition and image retrieval. Several studies [36]–[39] have extended or modified the original NetVLAD method, but these studies focused on replacing the backbone or adding more layers to improve the efficiency or accuracy of NetVLAD. The output descriptor does not fundamentally differ from the encoding method of NetVLAD, that is, after clustering, only the sum of the residuals between local features and the cluster center is calculated, while all other information related to geometric space is discarded. Because in cross-view matching, the local details can vary significantly between views. The way NetVLAD encoding features can be particularly problematic in the UAV-satellite image matching case. To address this problem, some studies have suggested two-stage approaches that utilizing NetVLAD to retrieve some image candidates and then conducting fine matching using local feature descriptors is a feasible means for achieving accurate localization [5], [40].

In conclusion, the NetVLAD-based SOTA methods face many challenges when the application involves multi-source and cross-view data due to their feature encoding and matching strategy. In order to take advantage of the clustering encoding of NetVLAD while retaining the image structure features after clustering, we propose a new encoding method. Our method first compares its similarity with the class of features learned by the network. When two features belong to the same class, they are considered similar features. This is different from directly comparing the similarity of features in terms of calculating distance like most SOTA methods do because introducing the class of features is equivalent to learning the range of feature variation. This is more general and robust compared to directly measuring the distance between two features. We also consider geometric information by proposing a graph encoding framework, which can be important for multi-source and cross-view place recognition.

III. End-to-End Scene Graph Encoding and Matching

A. Method Overview

The proposed method represents an image in a hierarchical form, comprising the semi-global descriptors and a global descriptor. The system overview is given in Fig. 2. The system architecture can be divided into two loosely coupled parts: 1) a CNN-based feature encoder consisting of the first four blocks of the pre-trained VGG16 [41] or ResNet-50 [42] backbone; 2) a graph encoder consisting of a global descriptor encoder based on a non-local network [15] and a CSMG encoder modified from NetVLAD [17]. First, we reroute the outputs of the Conv4 layer to the global descriptor encoder to upgrade the local convolutional features to the semi-global level. A channel-adapting convolutional layer and a max pooling layer are embedded into the front-end and back-end of the non-local block to adjust the input and output feature dimensions. Then, the output is directly flattened into a global descriptor as the global descriptor of the graph. Concurrently, the same output is fed into the CSMG encoder. The feature clusters trained by CSMG are used to assign the consensus features (grouped by the clusters) to a structure-aware scene graph. Instead of directly summing the feature cluster residuals as NetVLAD does, our work provides insight by computing the semantic centers of consensus features to extract the spatial relationship of the learned clusters in an embedded form. Meanwhile, we sort the nodes of the graph based on the size of their semantic feature distribution in the image before flattening the feature tensors to the semi-global descriptor so that we can eliminate the impact of image rotation. While introducing the structural information is a reasonable solution to improve

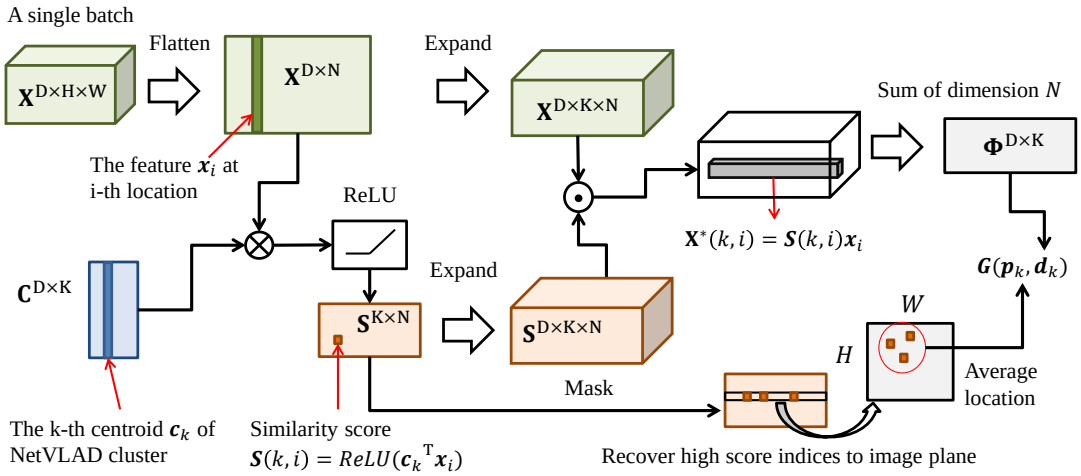


Fig. 3. CSMG encoder: Re-encodes the NetVLAD feature to a scene graph via cluster similarity masking. Compared with the original output of NetVLAD, our output retains both convolutional features and the structure information of the semantic features. The features are reweighted according to their similarity to the cluster centroids. This also allows us to embed the operations of Fig. 4 into the end-to-end network.

TABLE I
NOTATION LIST.

Symbol	Definition
[B,D,H,W]	Size of tensor [batch, channel, height, width]
K	Number of clusters
N	Number of features N=H×W
$\mathbf{x}$	Local image feature $\mathbf{x} \in \mathbb{R}^D$
$\mathbf{X}$	Feature tensor $\mathbf{X} = [\mathbf{x}_1, ..., \mathbf{x}_N] \in \mathbb{R}^{D \times N}$
$\mathbf{W}$	Weight matrix
$\mathbf{c}$	A cluster centroid vector
$\mathbf{C}$	Tensor of cluster centroids $\mathbf{C} = [\mathbf{c}_1, ..., \mathbf{c}_K]$
$\mathbf{S}$	Similarity score tensor
$\mathbf{d}$	Node descriptor
$\mathbf{\Phi}$	Tensor of node descriptors $\mathbf{\Phi} = [\mathbf{d}_1, ..., \mathbf{d}_K]$
$\mathbf{f}$	Flattened $\mathbf{\Phi}$, $\mathbf{f} = [\mathbf{d}_1^T, ..., \mathbf{d}_K^T]^T$
$\mathbf{\Omega}$	global descriptor descriptor
$\mathbf{p}$	2D positions of nodes
L	Loss

the matching accuracy, retrieving the whole graph in a large reference image set is challenging. To address this problem, the proposed system first finds the shortlist candidate graphs from the reference image set via global descriptor retrieval and then performs graph matching to score the candidates.

Reference image maps are usually different areas of a global map with defined geographic information. When the best match is found in the reference image set, the map area that overlaps significantly with the UAV image is also identified. Then the precise geo-location of the UAV will be determined via common photogrammetric techniques. Therefore, the proposed method can be used for state update in EKF-based localization methods or loop-closure detection in visual odometry (VO) methods [43], [44].

B. Preliminaries

The general notations used in this paper and their definition have been listed in Table I. The specific definition of a symbol with subscript and superscript is given in the relevant text.

The output of the *non-local block* [15] encoding the feature is as follows:

$$\mathbf{z}_i = \mathbf{W}_z \mathbf{y}_i + \mathbf{x}_i, \tag{1}$$

where "$+\mathbf{x}_i$" denotes a residual connection, weight matrix $\mathbf{W}_z$ computes a position-wise embedding on $\mathbf{y}_i$, $\mathbf{y} = softmax(\mathbf{x}^T \mathbf{W}_\theta^T \mathbf{W}_\phi \hat{\mathbf{x}}) g(\hat{\mathbf{x}})$, $g(\cdot)$ is a linear embedding, and $\hat{\mathbf{x}} = maxpool(\mathbf{x})$. This operation shares some similarities with the transformer's self-attention mechanism. The operation helps convolution-based networks to have a perception of the global knowledge of the image. This block does not change the input and output dimensions, so that it can be flexibly embedded into the convolutional network layers.

NetVLAD: Recall that in NetVLAD, a $K \times D$ vector $\mathbf{V}$ that represents the sum of the feature-cluster residual is defined as

$$\mathbf{V}(j,k) = \sum_{i=1}^{N} \overline{\alpha}_k(\mathbf{x}_i)(\mathbf{x}_i(j) - \mathbf{c}_k(j)), \tag{2}$$

where $j = 1, ..., D, k = 1, ..., K, \overline{\alpha}_k(\mathbf{x}_i) = \frac{e^{\mathbf{W}_k^T \mathbf{x}_i + \mathbf{b}_k}}{\sum_{k'} e^{\mathbf{W}_k^T \mathbf{x}_i + \mathbf{b}_{k'}}}$, $\mathbf{c}_k \in \mathbb{R}^D$ is a cluster center trained by NetVLAD, and $(\mathbf{x}_i(j) - \mathbf{c}_k(j))$ is the residual of the i-th feature and the k-th cluster centroid at the j-th dimension. Using the sum of the feature–centroid residual allows for efficient encoding of the semantic meaning of the whole image but also makes the output of the network features lose the spatial information of the image.

We redesign the feature aggregation of NetVLAD, as shown in Fig. 4. A group of consensus features can be found for each cluster centroid by computing its vector similarity. These features actually represent the distribution of a higher semantic feature in the image. Thus, we can obtain a scene graph in which each node is the dispersion center of a semi-global feature over the image space. Moreover, the descriptors of these semi-global features can be extracted simply by finding the weighted average of these features, and the weights are computed according to the similarity between features and cluster centroids.

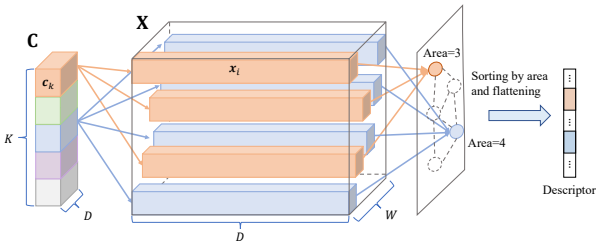


Fig. 4. Encoding the spatial locations for nodes by averaging the 2D location of the masked features on the $H \times W$ plane for each cluster.

To do so, we expect the output of the NetVLAD to be

$$\mathbf{X}^*(k) = \sum_i \overline{\alpha}_k(\mathbf{x}_i)\mathbf{x}_i, \;\; s.t. \; \mathbf{c}_k^{\mathrm{T}}\mathbf{x}_i > 0. \tag{3}$$

However, Eq. 3 is a discontinuous process for the network. To address this problem, we propose a CSMG encoder, which masks the output of the soft assignment (Eq. 3) using the score of the similarity between the features and the cluster centroids.

C. Cluster Similarity Masking Graph Encoder

We provide the implementation details of the cluster masking process. The output $\mathbf{X}^*$ of Eq. 3 is redesigned as a $B \times D \times K \times N$ tensor given by

$$\mathbf{X}^*(k,i) = ReLU(\mathbf{c}_k^{\mathrm{T}}\mathbf{x}_i) \odot \mathbf{x}_i, \tag{4}$$

where $\odot$ denotes the element-wise product. The volume with D dimension at (i,k) of the output represents the i-th feature $\mathbf{x}_i$ weighted according to the score of the similarity between the feature and the k-th cluster centroid $\mathbf{c}_k$. In Fig. 3, we show the full workflow of cluster similarity masking for a single batch ($B = 1$). When the batch size is considered, we can expand the cluster centroid tensor $\mathbf{C}$ to size $B \times D \times K$. We can compute the dot-product similarity $\mathbf{S}(B \times K \times N)$ along the dimension D (can be implemented by the function $torch.bmm()$ in PyTorch). The rectified linear unit (ReLU) is used to set the non-positive value to zero while keeping the whole process differentiable for network training. Then, we expand both $\mathbf{S}$ and $\mathbf{X}$ to the same size $B \times D \times K \times N$. The cluster masking process is achieved via matrix element-wise multiplication between $\mathbf{S}$ and $\mathbf{X}$.

Equation 4 shows that the output features are masked by zero for each cluster if they are not similar. Thus, we can obtain a similarity-weighted semantic descriptor for each node of the graph in Fig. 4 by summing up the dimension N of $\mathbf{X}^*$:

$$\mathbf{d}_k = \sum_i^N \mathbf{X}^*(k,i). \tag{5}$$

For the global descriptor $\mathbf{\Omega}$, we obtain it by flattening the output of the Non-local Block layer:

$$\mathbf{\Omega} = Flatten(\mathbf{x}_g), \tag{6}$$

where $\mathbf{x}_g$ is a $256 \times 14 \times 14$ tensor encoded by the Global Feature Encoder as Fig. 2 shows.

D. Graph Generation and Node Sorting

Let the graph of an image $\mathbf{G}(\mathbf{p}, \mathbf{\Phi}, \mathbf{\Omega})$ contain the node spatial locations $\mathbf{p}$, tensor of node descriptors $\mathbf{\Phi}$, and global descriptor $\mathbf{\Omega}$. The average spatial location of top-ranked consensus features of k-th cluster is denoted as $\mathbf{p}(k)$. To extract the spatial location from the new output $\mathbf{X}^*$, we find indices $\{i\}$ for all non-zero volumes of the slice $\mathbf{X}^*(k)$ and recover them to 2D locations on the $H \times W$ plane. The coordinate of k-th node is given as

$$\mathbf{p}(k) = \overline{\{(i/W, i\%W) | \forall i, \mathbf{S}(k,i) > \epsilon\}}, \tag{7}$$

where the similarity threshold ϵ is an adaptive value that aims to select the top-ranked features.

Although networks like GNN can provide accurate graph matching, it requires more precise definitions of nodes and edges. If we introduce the GNN into our workflow, it will break the end-to-end structure and increase the complexity of the network. Therefore, what we expect is still a flattened descriptor, which makes the process of retrieval or matching simpler and more efficient. In order to eliminate the impact of image rotation on the descriptor while retaining the structural information of the graph, we propose a method of sorting first and then flattening, as shown in Fig. 4. The nodes are sorted according to the proportion of each cluster's image features in the entire image, and then all feature vectors are flattened and added to a descriptor. This method is somewhat similar to PCA, but way simpler. If there is a main feature in a scene, such as a building or a river, then even if the image is taken from different angles, their encoded features will be ranked first by the semi-global descriptor. The relative relationship of different scenes in the scene is also preserved due to sorting.

E. Network Training and Cosine Triplet Loss

We train the descriptor encoder independently for image retrieval tasks. For the training process, the tensor $\mathbf{\Phi}$ will be directly flattened into a single image descriptor $\mathbf{f}$ for computing the triplet loss. Let tuples $(\mathbf{f}^a, \{\mathbf{f}_i^p\}, and \{\mathbf{f}_j^n\})$ be the anchors and the positive and negative samples, respectively. The anchors are obtained from the satellite images, while the positive and negative samples are extracted from the drone images.

A triplet loss with *weakly supervised ranking loss* presented by NetVLAD is the prior choice for the image retrieval task, where the loss function is given as

$$L_{tri} = \sum_j (l(\min_i(L2(\mathbf{f}^a, \mathbf{f}_i^p) + \alpha - L2(\mathbf{f}^a, \mathbf{f}_j^n)))), \tag{8}$$

where l is the hinge loss ($l(x) = max(x, 0)$), $L2(\cdot)$ is L2-norm, and α is the margin.

Cosine embedding loss [45] is also a candidate loss function because we use the vector similarity measure. However, as we need to maximize the gap between positive and negative samples, we need two cosine embedding loss functions with positive and negative labels, which are similar to triplet loss. We propose a cosine triplet loss that replaces the L2-norm in Eq. 8 with cosine similarity:

$$L_{cos} = \sum_j (l(\min_i(1 - cos(\mathbf{f}^a, \mathbf{f}_i^p) + \alpha + cos(\mathbf{f}^a, \mathbf{f}_j^n)))), \tag{9}$$

In addition, because two semantic levels of the descriptors with different dimensions are extracted in our method, to train the two parallel sub-blocks of the network together, our loss function contains two terms:

$$L = \beta L_{cos}^{\mathbf{\Omega}} + (1-\beta)L_{cos}, \tag{10}$$

where $L_{cos}^{\mathbf{\Omega}}$ computes the cosine embedding loss of the global descriptor descriptor, and β is the weight assigned to the global descriptor. Since our search strategy is to match global descriptor first, to ensure accuracy, the weight of the $L_{cos}^{\mathbf{\Omega}}$ is expected to be greater than or equal to 0.5. However, if the weight is too large, it will lead to slow convergence of the model during the training process and overfitting of the global descriptor. Therefore, in our implementation, we use a typical value of $\beta = 0.5$ which simply takes the average loss.

The output features can be directly used as descriptors of the image for matching after node sorting and tensor flattening, which is adopted in the network forward process.

F. Retrieval and Matching Process

Candidate graph extraction via global descriptor retrieval can be efficiently conducted by computing the dot-product similarity between the query image and the reference image of the encoded reference images. We use the superscripts q and r to denote query and reference images, respectively. Let $FS(\cdot)$ be the node sorting and feature tensor flattening operation. For the query image, the network output contains $(\mathbf{\Phi}^q, FS(\mathbf{\Omega}^q))$. The reference image contains the pre-encoded scene graphs of reference images (gallery) $\mathbf{G}^r = \{(\mathbf{\Phi}^r, FS(\mathbf{\Omega}^r))\}$. The overall workflow of the image retrieval process is given by Algorithm 1.

Algorithm 1: Retrieval Workflow

Data: Query image descriptors $(\mathbf{\Phi}^q, FS(\mathbf{\Omega}^q))$, reference image set $\mathbf{G}^r$.
Result: Top N matched reference ID in the reference image set.
Initialization:;
$simscore \leftarrow [\mathbf{\Omega}^q]^{\mathrm{T}}[\mathbf{\Omega}_1^r, ..., \mathbf{\Omega}_N^r]$;
$indices \leftarrow TopN(simscore)$;
for *each* t *in the* $indices$ **do**
  $scores[t] \leftarrow$ **Cosine similarity**$(FS(\mathbf{\Phi}^r), FS(\mathbf{\Phi}^q))$
end
Top-N matches $\leftarrow Sort(scores)$

IV. Experiment

We tested the proposed method on three UAV satellite VPR benchmarks for performance evaluation: the University-1652 dataset, the Aerial-view Large-scale Terrain-Oriented (ALTO) dataset, and the SenseFly dataset. The primary challenge associated with these datasets is cross-view image matching. For the University-1652 [46] and ALTO datasets [47], the reference images were remote images captured by satellites, while the query images were obtained by the airborne camera with time and view heading/angle differences. The reference image in SenseFly was a global colored 3D point cloud reconstructed from LiDAR and RGB camera views. Network training and evaluation were conducted, and the proposed method was compared with state-of-the-art (SOTA) methods in terms of the performances on the University-1652 and ALTO datasets, while the tests were extended to the SenseFly dataset without extra training.

The University-1652 dataset is a multi-view multi-source benchmark containing 1,652 university buildings worldwide, 701 of which are in the training dataset. The image data contains a satellite image (orthophoto map) and multiple UAV views from different headings and angles at low altitudes. In addition, the Google Earth files of these locations are provided. This helps the model to learn the viewpoint-invariant features. However, the authors used a synthetic drone (3D engine in Google Earth) to collect aerial images. The drone view rotated precisely around a fixed landmark so that the view-point-invariant feature learned by the network could concentrate near the center of rotation.

The ALTO dataset also contains satellite–UAV image pairs and was used for the general place recognition competition (GPR-Competition, https://sites.google.com/andrew.cmu.edu/gpr-competition/home) at the International Conference on Robotics and Automation (ICRA) 2022. A total of 13,781 high-altitude airborne camera images and 3,742 satellite images were collected. Most of the scenes in the dataset are natural terrains, including forests and lawns. However, the gap between the satellite maps and the drone images in the low-level semantic features of the dataset can be very large, because the natural scenery changes over time. Extracting invariant features from these images is rather difficult.

The SenseFly dataset is relatively small compared with the other two datasets. A total of 443 UAV images were collected by a fixed-wing drone flying over the campus on a specific path. Nonetheless, the map still contained various landscapes, and the UAV views were more continuous than those of the University-1652 dataset. Therefore, SenseFly can be used for testing the performance of the proposed network pre-trained on University-1652 for UAV visual localization tasks. However, the images in this dataset were characterized by many missing segments after the 3D point cloud was converted into an orthophoto map. This attribute is not present in the training data and is a strong test of the generation capability of the proposed model.

All experiments were conducted using UAV views as query images to match the pre-encoded reference image maps. The reference image map encoding was carried out using a trained network, which was detached from the network training process. The top 1 and top 5 recall results are denoted as $R@1$ and $R@5$, respectively.

A. Training and Hyperparameter Tuning

We first investigated the basic performance of the proposed network on the University-1652 benchmark. The training loss and testing accuracy under varying cluster numbers were evaluated. The detailed settings of the training process are given in Table II and Fig. 5. In the following content,

TABLE II
THE SETTING OF NETWORK TRAINING.

Training platform	RTX3090
Transforming and augmenting	resize, random corp, padding
Input image tensor size	$3 \times 224 \times 224$
Batch size	128
Output size (flattened)	number of clusters $\times$ 256
Optimizer/learning rate	Adam/0.001
Epoch	100

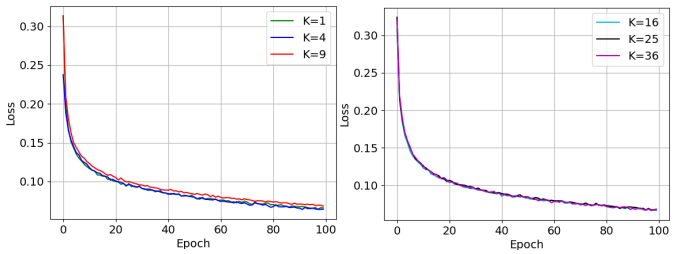


Fig. 5. Network training: epoch and loss. K is the number of clusters. After 80 epochs, the rate of decrease in training loss has reduced to an average of 0.001 per epoch. Considering the prevention of model overfitting, we believe that the model training should have reached the condition of convergence at this point.

we will analyze and discuss several important indicators and parameters during the model training process.

Model testing during the training period: The results of recall accuracy (Fig. 6 and Fig. 7) were based on the direct matching of the flattened network output **f**(Fig. 3) according to the dot-product similarity.

Model size and computational complexity: The proposed network model was divided into two parts. The backbone of the proposed model was the first four convolutional blocks (around 5.5% of the parameters) of the whole VGG16, and we did not need to retrain the weights of these parts. In Table III, we used the third-party toolbox **torchstat** in pytorch to analyze the parameter size and floating-point operations (FLOPs) of the proposed model as well as the SOTA method NetVLAD. The results show that by replacing the NetVLAD with the proposed model (using 4 clusters), the number of parameters and FLOPs drop from 140.6M and 94.2G to 8.9M and 15G, respectively, making it a lightweight network. Most of the computation load was on the GPU because most of the processes were embedded into the network. The full model (including the image retrieval process) ran at around 300 frames per second (FPS) on an RTX3090 GPU and around 37 FPS on Nvidia Jetson Xavier NX.

Impact of the loss and the number of clusters: We evaluated the properties of the proposed network under varying numbers of clusters. The top-1 and top-5 recall accuracies on the test set of the University-1652 benchmark over 100 training epochs are given in Fig. 6. The results showed a higher recall accuracy under a smaller number of clusters. We also plotted the loss (Fig. 7 left), and the results showed that no significant correlation existed between the number of clusters and accuracy. The highest top-1 recall accuracy achieved by the network within 100 epochs was plotted with respect to the loss and the number of clusters (Fig. 7). The results suggest that recall accuracy and loss were negatively

TABLE III
MODEL PARAMETER SIZE AND FLOPS (K=4, BACKBONE=VGG16, 1M=1×10^6, 1G=1×10^9)).

Model	NetVLAD	Our	Our(Remove backbone)
Parameter	140.6M	8.9M	1.3M
FLOPs	94.2G	15.0G	1.03G

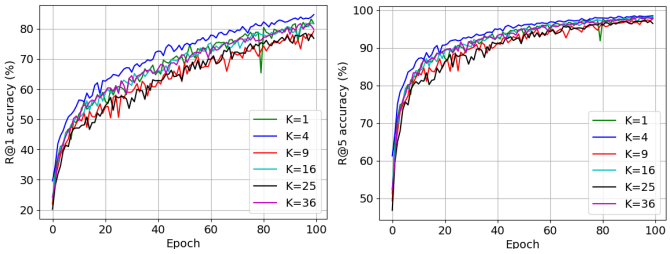


Fig. 6. Network training: Epoch and recall accuracy. K is the number of clusters. The model achieved the best accuracy on the test set when setting the number of clusters to 4.

correlated, and the number of clusters was more likely to affect the convergence speed of the model training. When the number of clusters was 1, the cluster centroid became the feature centroid of all images. The case in which the number of clusters is 1 represents a feature normalization over the whole training dataset. It is also equivalent to the condition in which the number of clusters is maximized. When only a few clusters were used for feature learning, the network tended to learn the semantic scene at a global level. The classification performance of each cluster centroid for semi-global features under an increasing number of clusters was observed. Increasing the number of clusters in the first small range made the model converge more slowly, and the situation improved with a further increase in the cluster number. We have added more experiments in the parameter range ($K = 2 \sim 8$) where the model performs better to fine-tune the parameter of the number of clusters. In our experiments, the highest accuracy was achieved when the number of clusters was 6, followed by 4. Considering that increasing the number of clusters will increase the size of the model, the reasonable choice is to set the number of clusters to around 4 for efficient model training and model weight reduction. When the hardware platform is very powerful and a larger backbone network is used, using a relatively large number of clusters with more training epochs is reasonable. When the number of clusters increases to a certain amount (Fig. 5 right), we find that it no longer significantly affects the training results. This is likely because when the cluster density is too high, different classes may correspond to very similar local image features, and the clustering results tend to stabilize, which can be intuitively understood by referring to Fig. 4. Thus, increasing the number of clusters does not make any positive contribution to the convergence of loss, as many of the feature classes learned by the network under dense clustering are similar and redundant. In the calculation of loss, it manifests as the addition of some duplicate terms, which does not significantly impact the final average.

Effect of the backbone output layer selection: A notable

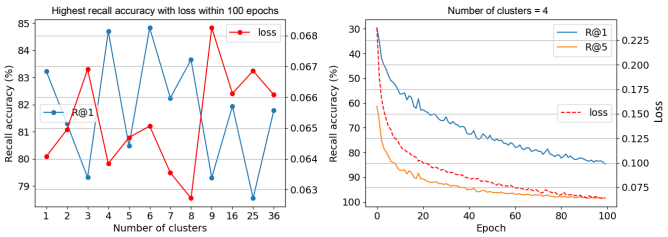


Fig. 7. The impacts of loss and number of clusters on recall accuracy. The results show a strong negative correlation between loss and recall accuracy. In contrast, the effect of the number of clusters on the convergence speed of the model training is more significant.

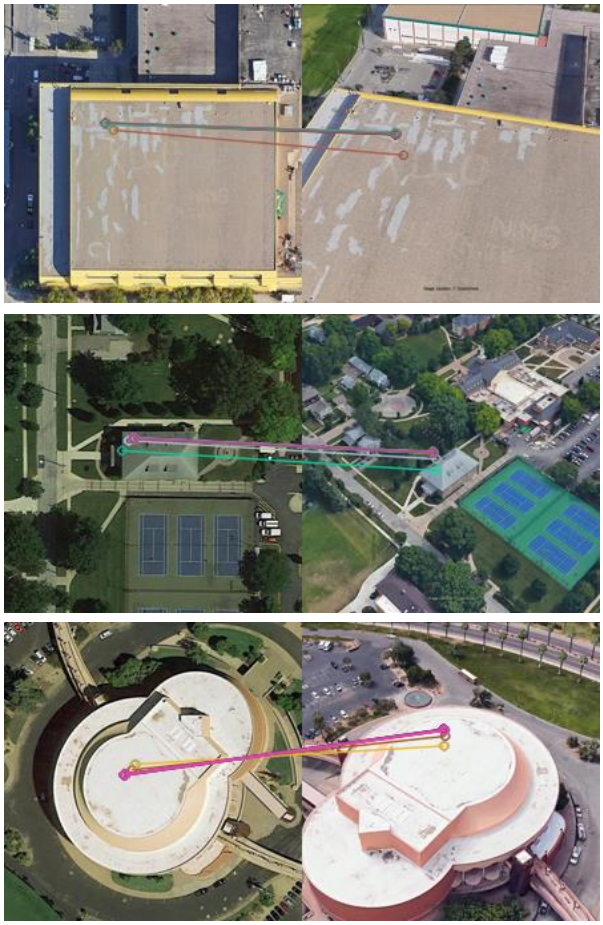


Fig. 8. The visualization of the node matching results ($K = 4$) on the University-1652 dataset.

study based on NetVLAD [17] experimentally investigated the effects of performing backpropagation down to a certain layer of the AlexNet backbone and suggested that overfitting may occur below the conv2 layer. In addition, the final version of NetVLAD based on VGG16 was trained from Conv4, which means that the pre-training weights of the first three convolutional blocks of VGG16 were retained. This choice also applies to our configuration, and we used a non-local block to replace the Conv5 block of VGG16 or ResNet-50.

Effect of the training data: The main drawback of this dataset was that its drone images were not real-world data. The dataset was obtained by setting a manually designed spiral flight path in the Google Earth 3D engine, which made the training data potentially biased and can lead to overfitting (which happened for $K = 16$ near 80 epoch, Fig. 7). The large viewpoint variation generated by the spiraling flight made it difficult for the model to learn the view-point-invariant features in the image edge regions. Moreover, the rotation of the landscape also caused the features learned by the proposed model to be evenly distributed around the image center. Consequently, the spatial locations of the graph nodes gathered in the center with small location differences. We present some scene graph-matching results of the proposed encoder in Fig. 8. When the non-local block was applied, this rotation did not affect the classification and extraction of the structure-aware scene graph. It only scaled down the graph, but the structured information of the graph itself was still present and played an important role in the retrieval process. This is demonstrated in Fig. 8, where the structural correspondences are shown.

Location distribution of the graph nodes: The spatial location of a graph node is the distribution center of a certain class of image features, which is fundamentally different from the traditional feature points. Therefore, their coordinates only represent the semantic structure and cannot be used as a precise reference for scene location.

Dimensions of the descriptor: We directly used the feature dimensions recommended by NetVLAD ($D = 256$) for each semi-global feature because the method that we used to train the clustering centroids was similar to theirs. Thus, the dimension of the output semi-global descriptor is the number of clusters times the feature dimension $K \times D$.

B. Benchmark Evaluation

Model configuration: Because the model convergence speed was not our main concern, we chose the model with four clusters according to our parameter tuning experiment (Fig. 7), and we increased the epoch during training until the network converged. The model configuration for benchmark evaluation is given in Table IV. This configuration is shared when using VGG16 or ResNet-50 as a backbone.

Evaluation protocol: For University-1652, we used all satellite images (1,651 images in total) as the map library. Eight hundred thirty-six of them contained tested UAV images, and the rest were used as distractors. Each satellite map corresponds to 54 drone images sampled uniformly along the spiral flight trajectory. Regarding the ALTO dataset, the satellite and UAV images had a high degree of overlap (more than 50%) within five consecutive frames. Therefore, we sampled one in every five images when building the reference map library (92 satellite images were sampled), while all UAV images were used for testing (1,684 images in total).

Comparison with SOTA results: We demonstrated the overall performance of the proposed method through two benchmark evaluations. The benchmark evaluation results of the University-1652 and ALTO datasets are compared with the recorded SOTA results in Table V and Table VI, respectively. The sampled results are shown in Fig. 9 and Fig. 10, respectively. We provide the results of retrieving satellite

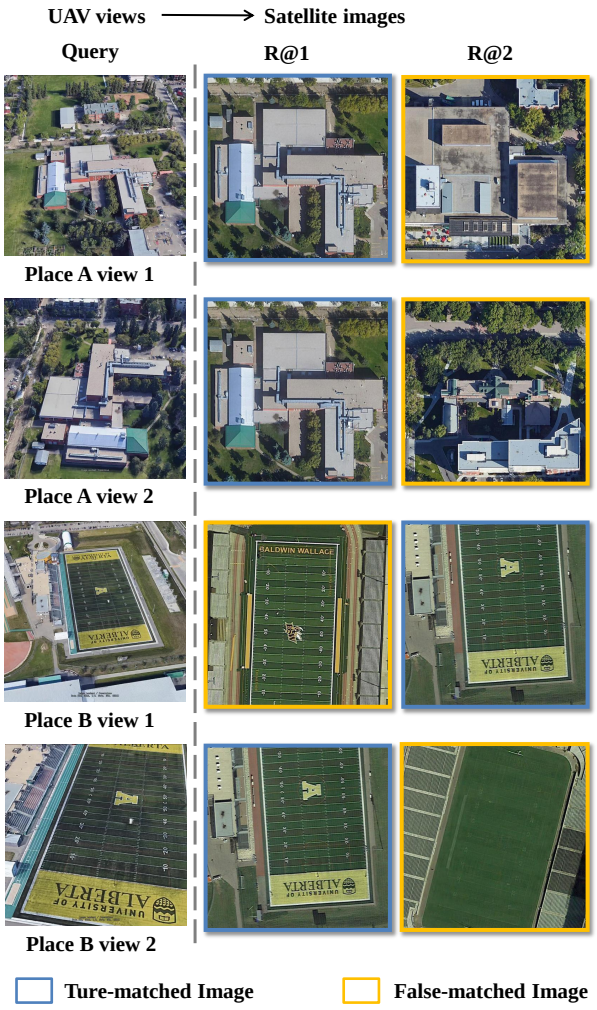


Fig. 9. Sampled results from University-1652 benchmark evaluation.

TABLE IV
THE MODEL CONFIGURATION FOR BENCHMARK EVALUATION.

Block Name	Output (Shape)	Training
Conv1	tensor(112,112,64)	No
Conv2	tensor(56,56,128)	No
Conv3	tensor(28,28,256)	No
Conv4 (remove pooling)	tensor(28,28,512)	No
global descriptor encoder	tensor(14,14,256)	Yes
CSMG encoder	tensor(4,256), points(4,2)	Yes

images from UAV views using VGG16 and ResNet-50. The proposed method achieved remarkable recall accuracy in both the University-1652 and ALTO datasets and outperformed the SOTA methods in retrieving satellite images from UAV views.

C. Ablation Study and Contribution Demonstration

We investigated whether the proposed scene graph was the key factor responsible for the high recall accuracy and on what level it contributed to the overall performance. To separately evaluate the impact of the introduction of the non-local block and the CSMG encoder, we conducted controlled experiments by removing one of these parts from the whole workflow and performed the same benchmark evaluation on

TABLE V
BENCHMARK EVALUATION AND SOTA METHODS COMPARISON ON UNIVERSITY-1652.

Method	Backbone	R@1
Contrastive Loss [48]	VGG16	52.39
TripletLoss (M = 0.3) [46]	ResNet-50	55.18
Soft Magin Triplet Loss [49]	VGG16	53.21
Instance Loss [50]	ResNet-50	58.23
LCM [51]	ResNet-50	66.65
SAFA [52]	VGG16	68.27
RK-Net [53]	VGG16	66.13
LPN [54]	ResNet-50	75.93
LPN+USAM [40]	ResNet-50	77.07
FSRA [55]	Vit-S	82.25
Ours	VGG16	85.26
Ours	ResNet-50	**87.36**

TABLE VI
ALTO BENCHMARK EVALUATION.

Method	Backbone	R@1	R@5
NetVLAD	VGG16	34.4	76.8
SAFA	VGG16	36.7	88.5
RK-Net	VGG16	31.2	74.4
LPN	ResNet-50	39.5	82.3
FSRA	Vit-S	46.4	89.7
Ours	VGG16	47.2	91.1
Ours	ResNet-50	**52.3**	**97.5**

the ALTO dataset (Fig. 11). The results indicated that both of the proposed parts contributed to the final performance. In addition, in Fig. 12, we plotted the confusion matrix for the global descriptor, semi-global descriptor, and overall, respectively. For the global descriptor, although the diagonal shows relatively clear highlights compared to other areas (very high true positive), there are also highlights in other areas of the matrix, which means that the false positive of the matching result of using this descriptor is not good enough. However, using semi-global for matching results in much fewer false positives, only appearing in a few adjacent frames. This is reasonable on the ALTO dataset, because the image data is continuously collected during flight, and adjacent frames are likely to contain the same feature-rich and highly recognizable scenes (such as buildings, roads, valleys, etc., illustrated in Fig. 10). The overall results indicated that the proposed global and structure-aware semi-global descriptors offer deterministic matching results.

D. VPR using Pre-trained Model for UAV

We evaluated the generalization ability of the proposed methods by conducting a test on the SenseFly dataset using the model previously trained on the University-1652 benchmark. The colored 3D point cloud was converted into an orthophoto and divided into 30 reference maps by 5×6 grid zones. Because the point cloud-to-optical pixel ratio was still sparse and there were blind areas in the 3D reconstruction, the reference maps were corrupted and contained discontinuous patches or missing pixels. Thus, the dataset was suitable for validating the generalizability of the trained models. In our test, 231 out of 443 UAV images were correctly matched to the corresponding zones in the global map (52.1% R@1 accuracy). The retrieval results with the visualization of the scene graph

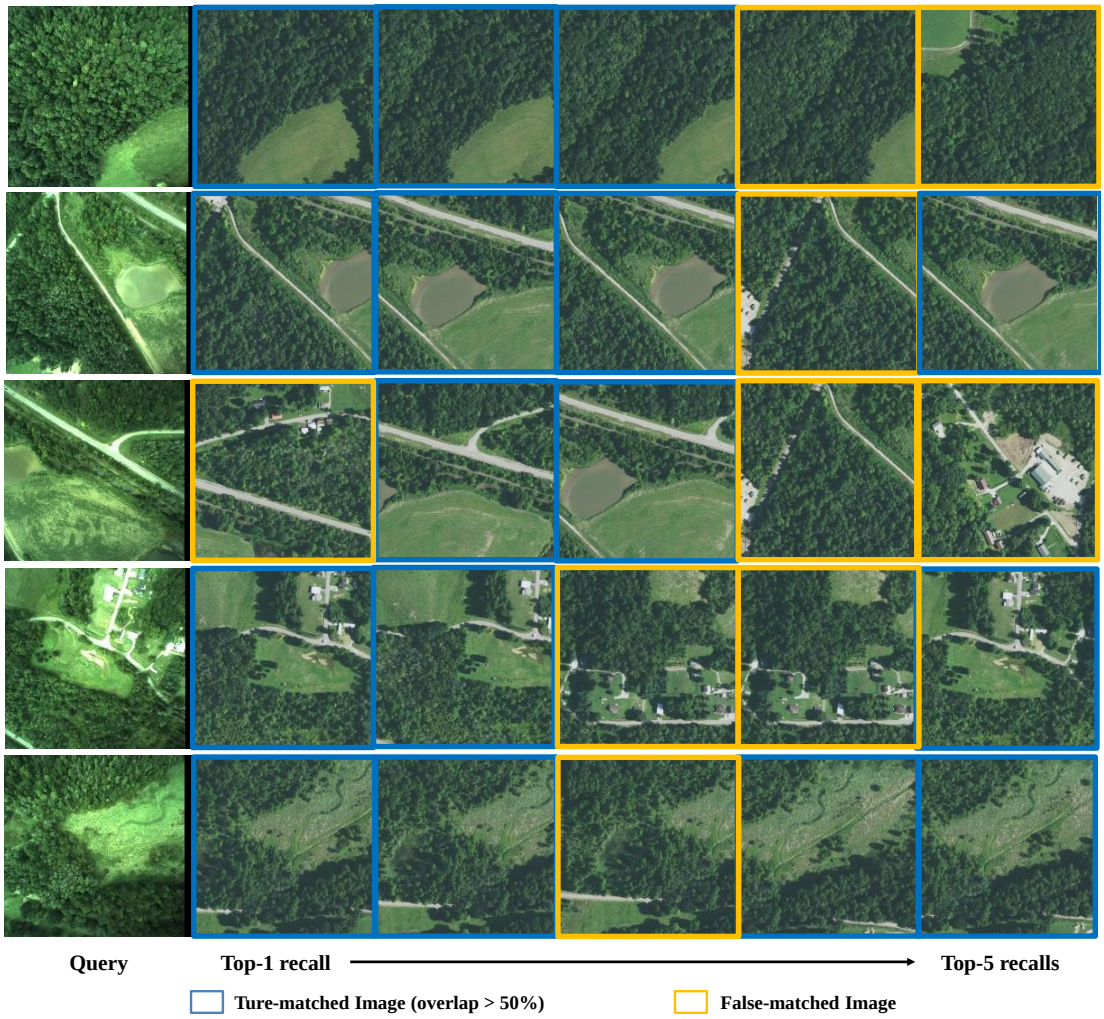


Fig. 10. Sampled results from ALTO benchmark evaluation.

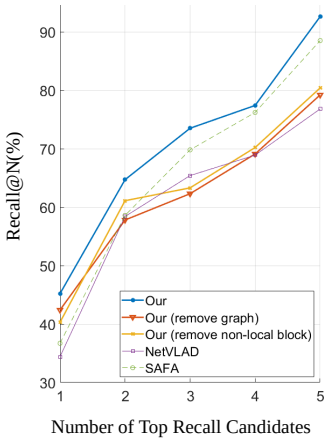


Fig. 11. Ablation study: Benchmark evaluation on ALTO after the disabling of certain components.

matching (Fig. 13) demonstrated that the proposed method could recall the reference image with favorable matching performance on the cross-view image pairs.

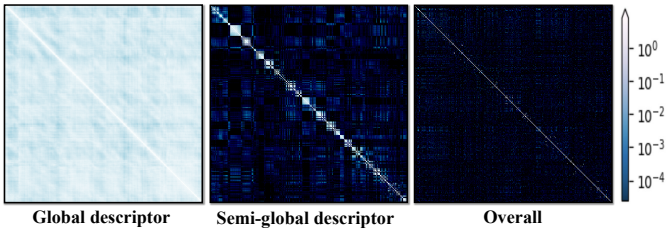


Fig. 12. Confusion matrices in different stages. The elements (i, j) represent the normalized vector dot product similarity between reference features i and j extracted from reference images.

V. Conclusion

We propose a novel network architecture to encode UAV images into scene graphs and match them with the reference image maps in an end-to-end manner for UAV cross-view visual localization. The backbone of the network is the first four convolutional blocks of VGG16 or ResNet-50 with pre-trained weights. Our contribution is a scene graph encoding network that is embedded into the backend of the CNN, which consists of a non-local block and a CSMG encoder

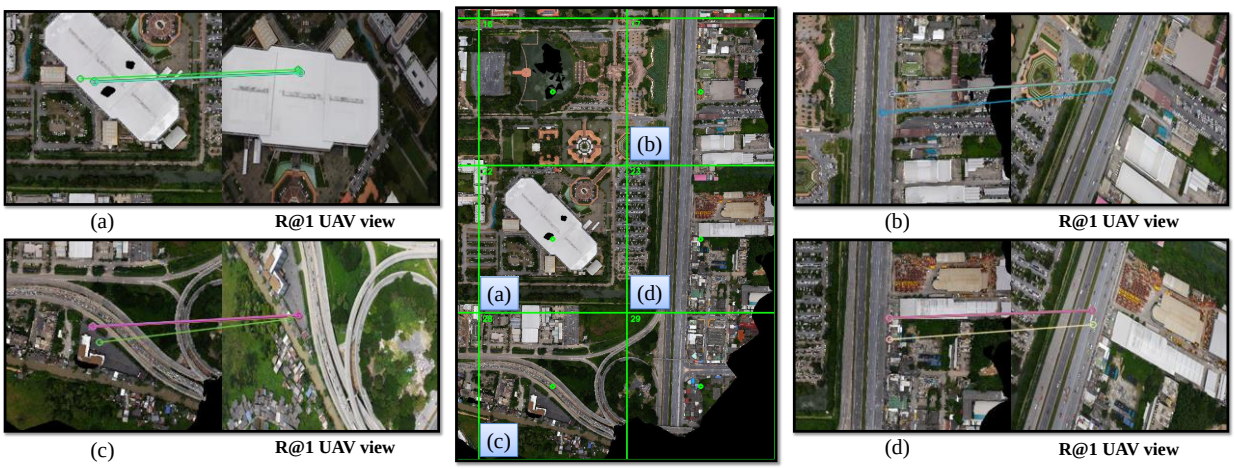


Fig. 13. Place retrieval results. The green grid marks the cropped reference images. The pre-trained model was used to rebuild the reference image data, in which the reference images were obtained from the gridded global map. The results showed that the proposed method could recall the place accurately.

modified from NetVLAD, to encode the global and semi-global descriptors, respectively. Different from the original NetVLAD, this modified network does not output the sum of the residuals of features and clustering centroids. The CSMG output is a graph representing the spatial relationship of semi-global features, in which the nodes are the weighted sum of the consensus features in each cluster. Their spatial locations on the image are the semantic centers of each cluster. Before the final flattening output of the network, we sort the semantics represented by each node in the graph according to the proportion of each semantic feature in the image. This allowed us to match images by taking into account the structure information of the semantic features. In addition, global and semi-global descriptors are retrieved hierarchically so that the system can perform efficient coarse-to-fine image retrieval. We conducted experiments on three drone datasets. Two of them were benchmarks for UAV place retrieval and contained drone images and satellite maps, and the other dataset included drone images and 3D maps. The experimental results indicated that the developed network model yielded promising matching results in retrieving reference images with UAV images and outperformed the SOTA approaches with respect to top-1 recall accuracy. This proposed network can suitably run on edge devices as the model is lightweight, and it has the potential to improve the ability of vision-based global localization and navigation error correction for UAVs. Future work will explore incorporating the graph structure into the loss function and training on image segmentation datasets to obtain models with better semantic segmentation capabilities for accurate matching and localization based on geo-referencing.

ACKNOWLEDGMENT

This work was supported by a grant from the Research Grants Council of Hong Kong (Project No: PolyU 15210520) and grants from The Hong Kong Polytechnic University (Project No: 1-ZVN6, Project No: P0044685, Project No: P0046112).

REFERENCES

[1] X. Wan, Y. Shao, S. Zhang, and S. Li, "Terrain aided planetary uav localization based on geo-referencing," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–18, 2022.

[2] L. Cheng, Y. Yuan, N. Xia, S. Chen, Y. Chen, K. Yang, L. Ma, and M. Li, "Crowd-sourced pictures geo-localization method based on street view images and 3d reconstruction," *ISPRS journal of photogrammetry and remote sensing*, vol. 141, pp. 72–85, 2018.

[3] 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, 2021.

[4] J. Phillips, J. Martinez, I. A. Bârsan, S. Casas, A. Sadat, and R. Urtasun, "Deep multi-task learning for joint localization, perception, and prediction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4679–4689.

[5] P.-E. Sarlin, C. Cadena, R. Siegwart, and M. Dymczyk, "From coarse to fine: Robust hierarchical localization at large scale," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 716–12 725.

[6] P.-E. Sarlin, F. Debraine, M. Dymczyk, R. Siegwart, and C. Cadena, "Leveraging deep visual descriptors for hierarchical efficient localization," in *Conference on Robot Learning*. PMLR, 2018, pp. 456–465.

[7] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. IEEE, 1999, pp. 1150–1157.

[8] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Lecture notes in computer science*, vol. 3951, pp. 404–417, 2006.

[9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: An efficient alternative to sift or surf," in *2011 International conference on computer vision*. Ieee, 2011, pp. 2564–2571.

[10] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 224–236.

[11] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, vol. 1, no. 1-22. Prague, 2004, pp. 1–2.

[12] Sivic and Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proceedings ninth IEEE international conference on computer vision*. IEEE, 2003, pp. 1470–1477.

[13] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 3304–3311.

[14] R. Arandjelovic and A. Zisserman, "All about vlad," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2013, pp. 1578–1585.

[15] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[16] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 9, pp. 1704–1716, 2011.

[17] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 5297–5307.

[18] B. Wu, L. Xie, H. Hu, Q. Zhu, and E. Yau, "Integration of aerial oblique imagery and terrestrial imagery for optimized 3d modeling in urban areas," *ISPRS journal of photogrammetry and remote sensing*, vol. 139, pp. 119–132, 2018.

[19] L. Ye and B. Wu, "Integrated image matching and segmentation for 3d surface reconstruction in urban areas," *Photogrammetric Engineering & Remote Sensing*, vol. 84, no. 3, pp. 135–148, 2018.

[20] Q. Zhu, Y. Li, H. Hu, and B. Wu, "Robust point cloud classification based on multi-level semantic relationships for urban scenes," *ISPRS journal of photogrammetry and remote sensing*, vol. 129, pp. 86–102, 2017.

[21] X. Ge, H. Hu, and B. Wu, "Image-guided registration of unordered terrestrial laser scanning point clouds for urban scenes," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9264–9276, 2019.

[22] M. A. Uy and G. H. Lee, "Pointnetvlad: Deep point cloud based retrieval for large-scale place recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4470–4479.

[23] G. Elbaz, T. Avraham, and A. Fischer, "3d point cloud registration for localization using a deep neural network auto-encoder," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4631–4640.

[24] P. Kadam, M. Zhang, S. Liu, and C.-C. J. Kuo, "R-pointhop: A green, accurate, and unsupervised point cloud registration method," *IEEE Transactions on Image Processing*, vol. 31, pp. 2710–2725, 2022.

[25] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl, "Semantic match consistency for long-term visual localization," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 383–399.

[26] D. Zhu, J. Li, and G. Li, "Rfi source localization in microwave interferometric radiometry: A sparse signal reconstruction perspective," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 58, no. 6, pp. 4006–4017, 2020.

[27] X. Zhang, L. Wang, and Y. Su, "Visual place recognition: A survey from deep learning perspective," *Pattern Recognition*, vol. 113, p. 107760, 2021.

[28] A. Irschara, C. Zach, J.-M. Frahm, and H. Bischof, "From structure-from-motion point clouds to fast location recognition," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 2599–2606.

[29] X. Chang, P. Ren, P. Xu, Z. Li, X. Chen, and A. Hauptmann, "A comprehensive survey of scene graphs: Generation and application," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 1, pp. 1–26, 2021.

[30] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5410–5419.

[31] M. Okawa, "Vector of locally aggregated descriptors with kaze features for offline signature verification," in *2016 IEEE 5th Global Conference on Consumer Electronics*. IEEE, 2016, pp. 1–5.

[32] X. Wu, X. Tian, J. Zhou, P. Xu, and J. Chen, "Loop closure detection for visual slam based on superpoint network," in *2019 Chinese Automation Congress (CAC)*. IEEE, 2019, pp. 3789–3793.

[33] P.-E. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superglue: Learning feature matching with graph neural networks," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 4938–4947.

[34] S. Deng, Q. Dong, B. Liu, and Z. Hu, "Superpoint-guided semi-supervised semantic segmentation of 3d point clouds," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 9214–9220.

[35] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on image processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[36] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 141–14 152.

[37] J. Yu, C. Zhu, J. Zhang, Q. Huang, and D. Tao, "Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 2, pp. 661–674, 2019.

[38] Y. Tang, X. Zhang, L. Ma, J. Wang, S. Chen, and Y.-G. Jiang, "Non-local netvlad encoding for video classification," in *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, 2018, pp. 0–0.

[39] Z. Li, C. D. W. Lee, B. X. L. Tung, Z. Huang, D. Rus, and M. H. Ang, "Hot-netvlad: Learning discriminatory key points for visual place recognition," *IEEE Robotics and Automation Letters*, vol. 8, no. 2, pp. 974–980, 2023.

[40] J. Lin, Z. Zheng, Z. Zhong, Z. Luo, S. Li, Y. Yang, and N. Sebe, "Joint representation learning and keypoint detection for cross-view geo-localization," *IEEE Transactions on Image Processing*, vol. 31, pp. 3780–3792, 2022.

[41] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[42] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[43] R. Duan, D. P. Paudel, C. Fu, and P. Lu, "Stereo orientation prior for uav robust and accurate visual odometry," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 3440–3450, 2022.

[44] R. Duan, D. P. Paudel, C.-Y. Wen, and P. Lu, "Filtering 2d-3d outliers by camera adjustment for visual odometry," *IEEE Transactions on Instrumentation and Measurement*, vol. 72, pp. 1–12, 2023.

[45] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, and W. Liu, "Cosface: Large margin cosine loss for deep face recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5265–5274.

[46] 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.

[47] I. Cisneros, P. Yin, J. Zhang, H. Choset, and S. Scherer, "Alto: A large-scale dataset for uav visual place recognition and localization," *arXiv preprint arXiv:2207.12317*, 2022.

[48] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, "Learning deep representations for ground-to-aerial geolocalization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 5007–5015.

[49] S. Hu, M. Feng, R. M. Nguyen, and G. H. Lee, "Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7258–7267.

[50] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, M. Xu, and Y.-D. Shen, "Dual-path convolutional image-text embeddings with instance loss," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 16, no. 2, pp. 1–23, 2020.

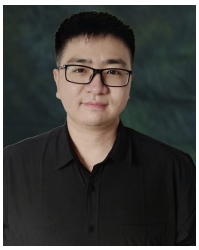
[51] L. Ding, J. Zhou, L. Meng, and Z. Long, "A practical cross-view image matching method between uav and satellite for uav-based geo-localization," *Remote Sensing*, vol. 13, no. 1, p. 47, 2020.

[52] 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.

[53] J. Lin, Z. Zheng, Z. Zhong, Z. Luo, S. Li, Y. Yang, and N. Sebe, "Joint representation learning and keypoint detection for cross-view geo-localization," *IEEE Transactions on Image Processing*, vol. 31, pp. 3780–3792, 2022.

[54] T. Wang, Z. Zheng, C. Yan, j. Zhang, Y. Sun, B. Zheng, and Y. Yang, "Each part matters: Local patterns facilitate cross-view geo-localization," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[55] 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*, vol. 32, no. 7, pp. 4376–4389, 2021.



Ran Duan received a B.S. degree in Communication Engineering from the School of Information Engineering, Southwest University of Science and Technology in 2013 and a Master's degree in Computer Vision and Robotics from the University of Burgundy, France (European VIBOT program) in 2015. From 2015 to 2017, he worked as a research associate at Nanyang Technological University (NTU), Singapore. In 2022, he earned his Ph.D. degree from the Department of Aeronautical and Aviation Engineering (AAE) at the Hong Kong Polytechnic University. He is currently a Research Assistant Professor in the Department of Land Surveying and Geo-Informatics at the Hong Kong Polytechnic University. His research areas include visual navigation, edge-AI, and UAVs.

Long Chen received the Ph.D. degree in photogramemtry and planetary mapping from The Hong Kong Polytechnic University, Hong Kong, in 2023. He is currently a Research Associate with the Hong Kong Polytechnic University. His research interests include real-time photogrammetry and image processing, 3D mapping, and robotic vision.

Zhaojin Li received the B.S. degree in remote sensing from the Wuhan University of China. She is currently pursuing a Ph.D. degree with a major in photogrammetry and remote sensing with the Hong Kong Polytechnic University. Her research interests include planetary mapping, photogrammetry and computer vision.

Zeyu Chen received B.S. degrees in East China Jiaotong University, China, and Troy University, U.S, and the MSc degree in the Hong Kong Polytechnic University. He is currently pursuing a Ph.D. degree with a major in planetary remote sensing with the Hong Kong Polytechnic University. His research interests include planetary mapping, automatic rock detection, and machine learning.

Bo Wu is a professor with the Department of Land Surveying and Geo-Informatics, the Hong Kong Polytechnic University. His research interests are mainly in planetary remote sensing and mapping, photogrammetry and robotic vision.