

# Joint Representation Learning and Keypoint Detection for Cross-view Geo-localization

Jinliang Lin, Zhedong Zheng, Zhun Zhong, Zhiming Luo, Shaozi Li, Yi Yang, Nicu Sebe

**Abstract**—In this paper, we study the cross-view geo-localization problem to match images from different viewpoints. The key motivation underpinning this task is to learn a discriminative viewpoint-invariant visual representation. Inspired by the human visual system for mining local patterns, we propose a new framework called RK-Net to jointly learn the discriminative Representation and detect salient Keypoints with a single Network. Specifically, we introduce a Unit Subtraction Attention Module (USAM) that can automatically discover representative keypoints from feature maps and draw attention to the salient regions. USAM contains very few learning parameters but yields significant performance improvement and can be easily plugged into different networks. We demonstrate through extensive experiments that (1) by incorporating USAM, RK-Net facilitates end-to-end joint learning without the prerequisite of extra annotations. Representation learning and keypoint detection are two highly related tasks. Representation learning aids keypoint detection. Keypoint detection, in turn, enriches the model capability against large appearance changes caused by viewpoint variants. (2) USAM is easy to implement and can be integrated with existing methods, further improving the state-of-the-art performance. We achieve competitive geo-localization accuracy on three challenging datasets, *i.e.*, University-1652, CVUSA and CVACT. Our code is available at <https://github.com/AggMan96/RK-Net>.

**Index Terms**—Geo-localization, Representation learning, Keypoint, Attention.

## I. INTRODUCTION

ROSS-VIEW geo-localization refers to inferring the geographical location from images of different viewpoints and is usually viewed as an image retrieval task [1]–[4]. Given a query image collected from one platform, *e.g.*, a drone, the system aims to retrieve the images of the target location from candidate images collected by another platform, *e.g.*, a satellite. Since satellite-view data are usually accompanied by detailed GPS metadata, we can efficiently infer the location of the query image. Cross-view geo-localization has been applied to a wide range of real-world tasks, including drone navigation

This work is supported by the National Natural Science Foundation of China (No. 61876159, No. 61806172), the EU H2020 project AI4Media No. 951911 and the PRIN project CREATIVE Prot. 2020ZSL9F9.

J. Lin, Z. Luo (corresponding author) and S. Li (corresponding author) are with the Department of Artificial Intelligence, Xiamen University, Xiamen 361005, China (e-mail: jinlianglin@stu.xmu.edu.cn, zhiming.luo@xmu.edu.cn, szlig@xmu.edu.cn).

Z. Zheng is with NExT++, School of Computing, National University of Singapore, Singapore 118404. E-mail: zdzheng@nus.edu.sg. Z. Zheng was partially supported by Zhejiang Lab's International Talent Fund for Young Professionals (No.ZJ2020GZ021).

Y. Yang is with the College of Computer Science and Technology, Zhejiang University, China, 310027. E-mail: yangics@zju.edu.cn

Z. Zhong and N. Sebe are with the Department of Information Engineering and Computer Science (DISI), University of Trento, Trento 38123, Italy (e-mail: zhunzhong007@gmail.com, niculae.sebe@unitn.it).

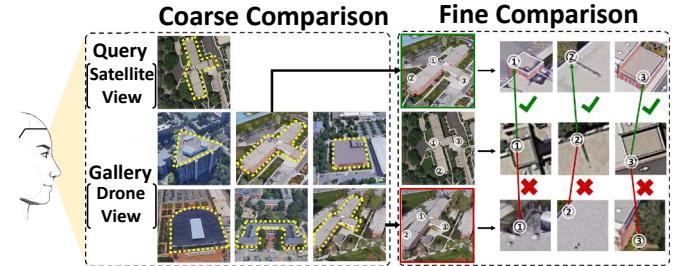


Fig. 1: Illustration of our motivation. We are inspired by the human visual system in distinguishing images: given two images, people usually focus on the whole image at a coarse level (**left**) and then compare salient regions at a fine comparison stage (**right**). The positive/negative image of the query is in the green/red box at the final stage.

[5]–[8], event detection [9]–[13], drone delivery [14], [15], and so on. Compared with GPS devices with a 2~15 meter position error range [16], the primary advantage of cross-view geo-localization is that it benefits from more fine-grained and accurate environmental information for the target location. In addition, it can be utilized as an independent auxiliary tool to improve user localization when the GPS signal is missing or is relatively weak.

The key aim underpinning cross-view geo-localization is to extract discriminative features, which remains challenging due to the small inter-class difference. Specifically, this is because most architectures/locations share similar building styles and homogeneous appearances, which are difficult to distinguish from the coarse level. The way in which the human visual system [17] distinguishes two similar images greatly inspires us. As shown in Fig. 1, when looking for the differences between two buildings, the human visual system first focuses on the general properties of the architecture, such as shape, style, and color. If it is difficult to identify the positive image from the global feature in a rough comparison, the human visual system further extracts some view-invariant keypoints with discriminative fine-grained information to determine the positive candidates.

Despite the great success of deep models in cross-view geo-localization [3], [18]–[24], view-invariant keypoints containing discriminative information have not been well explored. Inspired by the processes of the human visual system and classical hand-crafted descriptors (*e.g.*, SIFT [25]–[27] and LBP [28]–[30]), we propose a novel framework called RK-Net to jointly learn Representation and detect Keypoints with one

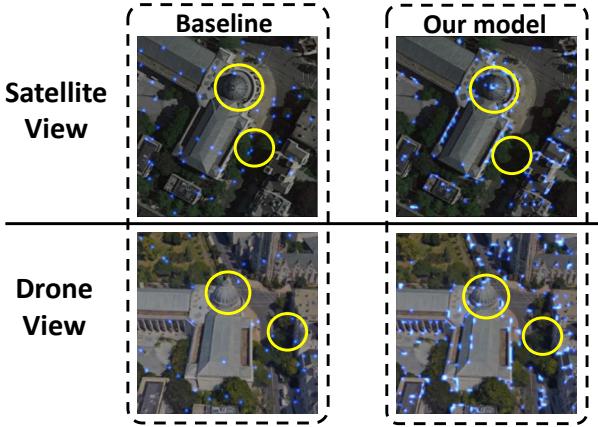


Fig. 2: A comparison of keypoint detection by the baseline approach [18] and our method. As shown in yellow circles, our method generally devotes more attention to discriminative regions, *e.g.*, the roof, while the baseline model focuses on “nonspecial” areas, *e.g.*, vegetation. It is worth noting that our model automatically extracts keypoints from corresponding regions for both satellite-view images and drone-view images without extra supervision.

Network. RK-Net contains a novel Unit Subtraction Attention Module (USAM), which is a plug-and-play module. As shown in Fig 3, the key operation of USAM is Unit Subtraction Convolution (USC). Without the prerequisite of extra keypoint annotation, USC can effectively and efficiently detect keypoints to help the model learn a discriminative representation. Specifically, USC explicitly enables the comparison between the center point and the surrounding points in the receptive field of the convolution kernel. *RK-Net leads to interaction between keypoints and visual representation.* The keypoint information extracted by our module contains fine-grained features of the target, helping to extract discriminative representations. Representation learning, in turn, encourages the model to obtain more representative keypoints from USAM. As shown in Fig. 2, the baseline model [18] usually focuses on some “nonspecial” areas (such as vegetation) while ignoring the distinctive parts of buildings (such as roofs). Instead, our model can effectively mine more salient keypoints and assign higher confidences to discriminative regions that are essential to distinguish different buildings/locations.

To summarize, our contributions are threefold:

- We present a novel framework, called RK-Net, to jointly learn discriminative representation and detect keypoints for Cross-view Geo-localization without the prerequisite of extra annotations.
- We design a Unit Subtraction Attention Module (USAM) as the primary component of RK-Net that can automatically discover representative keypoints. The keypoints force the model to focus on salient regions, yielding robust features against viewpoints.
- Our framework can be integrated with most existing methods and can significantly improve the performance, *e.g.*, boosting the global-based model [18] and the

ad hoc part-based model [24]. Extensive experiments show that the proposed method achieves competitive results on three cross-view geo-localization datasets, *i.e.*, University-1652, CVUSA and CVACT.

## II. RELATED WORK

**Cross-view Image-based Geo-localization.** Image-based geo-localization has attracted significant attention as a result of its numerous applications. Due to the large viewpoint changes of images from different platforms, the development of cross-view geo-localization has encountered a bottleneck in the method of hand-crafted feature matching [31]–[33]. Benefiting from the use of deep convolutional neural networks, recent geo-localization works focus on learning deep representations for both ground and aerial images to improve the performance of geo-localization. Workman *et al.* [19] are the first to adopt deep learned features for the cross-view matching task. Specifically, [19] uses a network pretrained on ImageNet [34] and Places [35] to extract features for the cross-view localization, which can effectively distinguish the target between two geographic regions. Moreover, [19] finds that the discrimination of the model representation can be further improved by minimizing the feature distance between positive pairs of ground-view images and aerial images [20]. To leverage orientation information, Liu *et al.* [3] design a Siamese network to explicitly encode the orientation information of the images. To handle the problem of orientation misalignment in cross-view geo-localization, [36] designs a Dynamic Similarity Matching (DSM) module to measure the feature similarity of the image pair. Shi *et al.* [21] propose a spatial-aware layer that exploits the spatial information to improve the localization performance. In the view of image generation, Krishna *et al.* [22] utilize the conditional GANs [37] to synthesize images from one view to the other, which minimizes the domain gap between the two views. Toker *et al.* [23] also create realistic street views from satellite images and localize the corresponding query street-view simultaneously in an end-to-end manner. Recently, Wang *et al.* [24] propose a feature-level partition strategy to make use of the contextual information of neighboring areas. There are some works that devote attention to metric learning and design different losses to train a discriminative network, *e.g.*, weighted soft margin ranking loss [38] for fast training convergence, contrastive loss [1] motivated by face verification [39], foreground loss [40], orientation regression loss for learning orientation-aware representations, and instance loss [18] inspired by cross-modality retrieval [41].

**Attention Mechanism in Geo-localization.** The attention mechanism is an effective technique to reassign the available resources to the most important part/region of an image. Over the past years, the attention mechanism has been used in a wide range of tasks which require identifying subtle contrastive clues from different images, such as person re-identification [42]–[46], fine-grained image recognition [47]–[49], and geo-localization [4], [50], [51]. To exploit attention for objects and patches of interest, Altwaijry *et al.* [50] inject the Spatial Transformer module [52] into a Siamese network to explore a set of possible matching patches. In addition, Tian *et*

*al.* [4] present a two-stage framework by taking advantage of image classification and object detection. Specifically, [4] employs Faster R-CNN [53] to detect buildings in the query and reference images and represents the images by the dominant sets constructed by features inferred from patches of buildings. This approach can achieve good geo-localization accuracy and is able to be generalized for images at unseen locations. Both methods [4], [50] enhance the robustness of object features to visual transformations by exploring specific landmark areas with an extra off-the-shelf detection network. To address the challenge of temporal variation in scenes for cross-view image geo-localization, [54] proposes a semantically driven data augmentation technique and a multi-scale attention module to enable the network to hallucinate unseen objects. Cai *et al.* [51] propose a context-based attention module (FCAM), which sequentially re-weights features by using channel and spatial attention submodules. Yang *et al.* [55] propose the L2LTR network based on Transformer to model global dependencies. Specifically, a self-cross attention mechanism is designed to interact within cross-layer patches, which can ensure effective information flow across Transformer blocks. Different from existing works, we focus on automatically discovering remarkable salient keypoints from feature maps and encouraging the model to pay attention to salient regions, yielding more discriminative visual representations.

**Keypoint detection.** Traditional handcrafted feature detectors are widely used for keypoint detection. In Harris [56], the first and second-order derivatives of images are computed to excavate the geometric structures. To speed up keypoint retrieval, FAST [57] counts the number of brighter or darker pixels around a point followed by a decision tree to improve performance and efficiency. Integrating detectors and descriptors, SIFT [25] looks for scale-invariant corners or blobs by convolving the image with Gaussian filters over multiple scale levels. The later SURF [58] aims to accelerate the detection process by using an approximation of the Hessian matrix and integral images. In MSER [59], the images are binarized at various thresholds, and the stable regions are selected as keypoints. As newer classical algorithms, KAZE [60] and its extension, A-KAZE [61], apply the Hessian matrix to a nonlinear diffusion scale space, which is computed at multiple scales. With the advent of deep learning, recent works learn to detect and describe keypoints by convolutional neural networks. To extract robust keypoints under severe weather and illumination changes, TILDE [62] trains a multiple piecewise linear regression model. In Lenc *et al.* [63], a feature covariant constraint is introduced to train a keypoints detector. SuperPoint [64] is an encoder-decoder architecture that is trained through a self-supervised mechanism. Savinov *et al.* [65] use a ranking scheme of point responses and quadruple image patches to train a model for keypoint detection. LF-Net [66] embeds the entire feature extraction pipeline and estimates the position, scale and orientation of features by jointly optimizing the detector and descriptor, which can be trained end-to-end with just a collection of images.

In contrast, the proposed method primarily differs from existing methods as follows: (1) Additional annotations, such as camera pose, depth, and so on, are not required with our

method. (2) Orientation estimation is not conducted in our method. (3) The proposed method takes the point-to-point relationship into account.

### III. METHODOLOGY

This section introduces the proposed joint Representation learning and Keypoint detection Network (RK-Net). In our RK-Net, the key component is the proposed Unit Subtraction Attention Module (USAM). We first review the traditional convolution in Sec. III-A. Then, in Sec. III-B, we introduce the core operator of USAM, *i.e.*, Unit Subtraction Convolution (USC), and discuss the differences from existing works. In Sec. III-C, we present how to generate the keypoint attention mask with our USC, followed by the residual attention fusion with the generated keypoint attention mask in Sec. III-D. In Sec. III-E, we illustrate the baseline model [18] and the model equipped with USAM, where the latter model enables us to jointly learn the representation and to detect keypoints during training.

**Overall Framework.** An overview of our RK-Net is shown in Fig. 3. We embed the proposed USAM between different stages of the network, each of which takes the features from the previous stage as input and outputs the features produced by USAM to the next stage. USAM consists of feature aggregation, keypoint mask generation by USC and residual attention fusion, which encourages the model to extract a more discriminative representation for cross-view geo-localization.

#### A. Traditional Convolution Review

In a traditional convolutional neural network, the convolution operation is conducted on feature maps that are represented by a three-dimensional form (*i.e.*, height  $h$ , width  $w$ , and channel  $c$ ). In the following section, we discuss the convolution method in a 2D spatial map for better explanation and understanding. The traditional convolution mainly includes two steps: 1) **Element-wise product**, where a matrix sampled from an input feature map and the values of the convolution kernel are multiplied element by element; 2) **Summation**, where all the values obtained in the first step are summed. Given an input feature map  $F$  and the kernel weight  $\psi$ , the output of the convolution operation can be obtained by:

$$F'(i, j) = \sum_{u, v \in A} \psi(u, v) \cdot F(i - u, j - v), \quad (1)$$

where  $i$  and  $j$  denote the coordinates of an element in terms of the height and width dimensions, respectively.  $\psi$  is the weight value of the convolution kernel.  $u$  and  $v$  represent the locations in the local receptive field area  $A$ . For example, if the size of the convolution kernel is  $3 \times 3$ , area  $A$  is  $\{-1, 0, 1\}$ .

#### B. Unit Subtraction Convolution

In our task, the keypoints are discriminative positions that are important to distinguish the targets. In general, a keypoint has a high response value in the feature map and has a large difference from its surrounding points. The traditional convolution uses a weighted sum to aggregate values in a

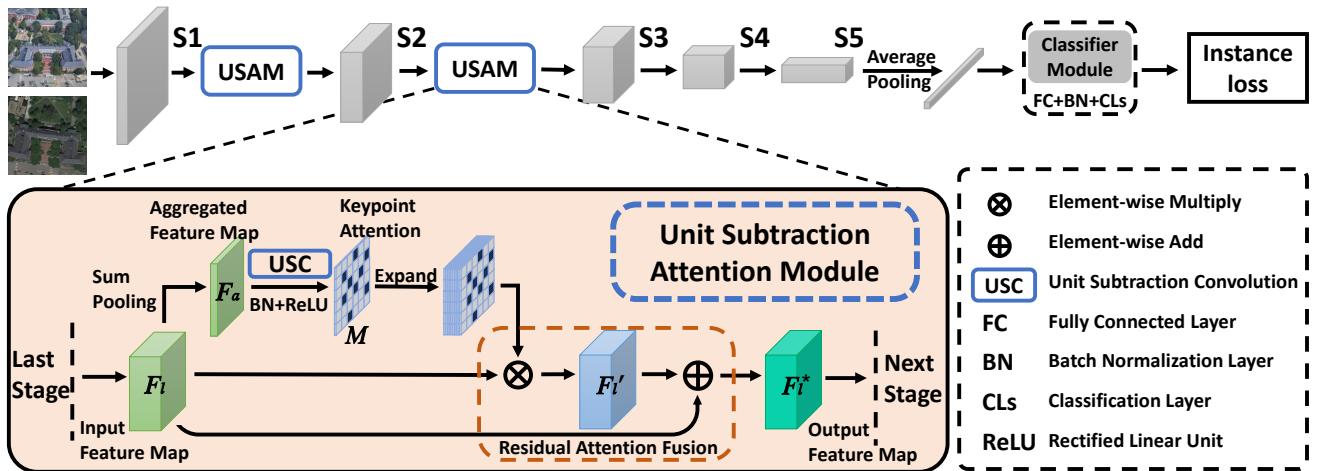


Fig. 3: Overview of the proposed RK-Net framework. The backbone of our network is ResNet-50, which contains five stages. We embed the proposed Unit Subtraction Attention Module (USAM) behind stage 1 (S1) and stage 2 (S2). We remove the original classifier of ResNet-50 and insert one 512-dimensional fully connected layer (FC), one batch normalization layer (BN) and one classification layer (CLS) to form a new classifier block. The instance loss is adopted to train our model. The details of USAM are shown in the orange box, which consist of feature aggregation, keypoint mask generation by the proposed Unit Subtraction Convolution (USC), and residual attention fusion.

region, which ignores the point-to-point relationship between an element and its surroundings. Considering this fact, we propose the Unit Subtraction Convolution (USC) to extract the keypoints from the feature map. By replacing the multiplication operation of traditional convolution with a subtraction operation, our USC can be formulated as:

$$F'(i, j) = \sum_{u, v \in A} (F(i, j) - F(i - u, j - v)). \quad (2)$$

Comparing Eq. 1 with Eq. 2, we can observe that there are two differences between our USC and the traditional convolution. *First*, USC does not introduce any learning parameter, while the traditional convolution requires a parametric kernel to produce the output. *Second*, USC compares the relationship between the center element and its adjacent elements by the subtraction operation instead of summing the values in an area  $A$  with the weights of the kernel. By doing so, if an element has a high value in a feature map and has a large margin from the surrounding elements, it can have a high positive value with USC and can be regarded as a keypoint. To reduce the computational cost and efficiently implement USC in practice, we convert Eq. 2 to:

$$F'(i, j) = K \cdot F(i, j) - \sum_{u, v \in A} F(i - u, j - v). \quad (3)$$

In this way, USC is divided into two parts. The first part is the multiplication between the input feature map and a weight  $K$ , where the value of  $K$  is equal to the size of the convolution kernel. For instance, if the size of the convolution kernel is  $3 \times 3$ ,  $K$  is set to 9. The second part is performing the traditional convolution on the input feature map using a kernel with a fixed weight of 1. That is, the second part can be obtained by Eq. 1 with  $\psi = 1$ . Consequently, USC can be implemented quickly with the convolution operation that is

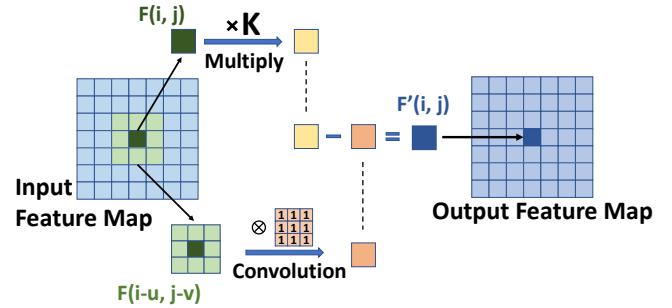


Fig. 4: Example of the proposed Unit Subtraction Convolution. The weight  $K$  is equal to the size of the convolution kernel. In this example, the kernel size is  $3 \times 3$  and  $K = 9$ .

built in the existing deep-learning tools. An example of USC is illustrated in Fig. 4.

**Discussion.** Some existing works [67], [68] have applied the unsharp mask method to generate a masking layer for their deep learning networks. Although both our module and their works use the position information between the center pixel and its surrounding pixels, our method has two main differences from this line of work: **1)** Different inputs. Our module can be implemented on any feature map and can be embedded in different layers of the network. The works involving the unsharp masking layer are implemented on the input original images rather than the feature maps, which constitutes a preprocessing layer. **2)** Different filter settings. There are no learning filter parameters in our proposed USC, while existing works require learnable filters.

### C. Keypoint Attention Mask

In this paper, USC is performed on a single feature channel and can be injected into any convolutional layer of a network.

Given an input image  $I \in \mathbb{R}^{3 \times H \times W}$ , we first extract an intermediate feature map from a certain convolutional layer  $l$ , by feeding the input image into a network. The intermediate feature map is denoted as  $F_l \in \mathbb{R}^{c \times h \times w}$ , where  $c$ ,  $h$  and  $w$  indicate channel numbers, height and width, respectively. To take all elements along the feature channels into consideration and efficiently compute the keypoint attention map, we apply the sum-pooling operation along the channel axis to aggregate the feature map (see Fig. 3). This simple method has been verified to be effective in highlighting salient/informative regions in the feature map [69]. The aggregated feature map can be obtained by

$$F_a = \text{SumPool}(F_l), \quad (4)$$

where  $\text{SumPool}(\cdot)$  is the sum-pooling operation performed on a feature map along the channel axis. Given the aggregated feature map  $F_a \in \mathbb{R}^{1 \times h \times w}$ , we take it as input for the proposed USC and generate the keypoint attention mask by:

$$M = \text{ReLU}(\text{BN}(\text{USC}(F_a))), \quad (5)$$

where USC is the Unit Subtraction Convolution proposed in Sec. III-B. BN is the batch normalization [70], and ReLU is the rectified linear unit activation function [71].  $M \in \mathbb{R}^{1 \times h \times w}$  is the produced keypoint attention mask, which is employed to generate a discriminative feature map that focuses on keypoint regions in Sec. III-D. *It is worth noting that the keypoint attention mask is a soft mask without a predefined threshold.*

#### D. Residual Attention Fusion

Given the generated keypoint attention map  $M$ , we aim to utilize it to highlight important regions in the feature map. Since  $M$  is a soft mask, we can regard it as a weight function and use the element-wise product to re-weight the values in the feature map. We first reduplicate the channel of  $M$  to the same size of  $F_l \in \mathbb{R}^{c \times h \times w}$  and obtain the weighted feature map by:

$$F'_l = F_l \cdot \text{Expand}(M, c), \quad (6)$$

where  $c$  is the channel size of  $F_l$ . To avoid the phenomenon that the soft mask might potentially discard important information of the original feature map  $F_l$ , we adopt a residual structure [72], [73] to produce the final feature map,

$$F_l^* = F_l + F'_l. \quad (7)$$

We call this process *residual attention fusion*. In this way, we can preserve important information in the original feature map while enforcing the model to pay attention to keypoint regions to produce a discriminative representation. Note that our USAM only contains very few learning parameters that are learnable affine parameters in the BN [70] layer and requires only a negligible computational cost. Therefore, USAM can be viewed as a learning parameter-free module. As shown in Alg. 1, we present the PyTorch-like pseudocode of the proposed Unit Subtraction Attention Module (USAM) to illustrate our method in detail.

---

**Algorithm 1** Pseudocode of USAM in a PyTorch-like style.

---

**Inputs:** A feature map  $F_l$ , kernel size  $n$

**Outputs:** A new feature map  $F_l^*$  produced by USAM.

```

1: # generate the aggregated feature map.
2:  $F_a = F_l.\text{sum}(1, \text{keepdim=True})$ 
3: # calculate the weight K of USC.
4:  $K = n * n$ 
5: # initialize a kernel with a fixed weight of 1 for USC.
6: kernel = parameter(
    data = torch.ones(1, 1, n, n), requires_grad = False)
7: # unit subtraction convolution for  $F_a$ .
8:  $F_{usc} = K * F_a - \text{conv2d}(F_a, \text{kernel})$ 
9: # batch normalization layer.
10:  $M = \text{BatchNorm2d}(F_{usc})$ 
11: # rectified linear unit layer.
12:  $M = \text{ReLU}(M)$ 
13: # residual attention fusion.
14:  $F'_l = F_l * M.\text{expand\_as}(F_l)$ 
15:  $F_l^* = F'_l + F_l$ 
16: Return  $F_l^*$ 

```

---

#### E. Model Training

**Baseline Model.** Following [18], we use a three-branch CNN network as the baseline model. The branches are designed for the satellite-view, drone-view and street-view images, respectively. The parameters of the satellite-view branch and the drone-view branch are shared since both satellite-view and drone-view images are from the aerial viewpoint. In this paper, we adopt ResNet-50 [72] as the backbone and replace the last classification layer with a new classifier block, which consists of a fully connected layer (FC), a batch normalization layer (BN) and a classification layer (Cls). The parameters of the classifier block are shared for all branches.

**Loss function.** Our method utilizes the instance loss [41] as the loss function to train the model. Concretely, instance loss regards each location as an individual class and trains the model in a classification manner. Given an image  $I_v^y$ , which is from the  $v$  view (satellite, drone or ground) and belongs to class  $y$ , the instance loss can be formulated by

$$p(I_v^y) = \text{SoftMax}(\text{Cls}(F_v(I_v^y))), \quad (8)$$

$$L_{\text{instance}} = -\log(p(I_v^y)[y]), \quad (9)$$

where  $F_v$  is the feature extractor of the  $v$  view and  $\text{Cls}$  is the shared classifier layer.

**Training with USAM.** Given the baseline model, we aim to equip it with our proposed Unit Subtraction Attention Module (USAM) for joint representation learning and keypoint detection. Specifically, we regard the five convolution residual blocks of ResNet-50 as five stages and embed USAM at the end of stage 1 and stage 2. With the USAM, the output of each stage is recomputed by aggregating the original feature map with a new feature map re-weighted by the generated attention mask. The output is then forwarded to the next stage. After injecting the USAM into the network, the model can be trained in the same manner used in the baseline without further modification. Note that the increase in running

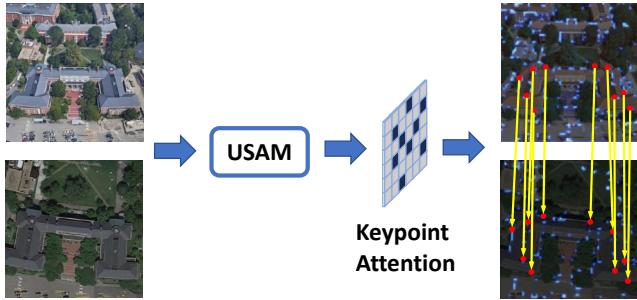


Fig. 5: Illustration of detecting keypoints by USAM. Red points represent detected keypoints, and yellow lines indicate the corresponding relationship between the keypoints from two images. Note that the keypoints extracted by USAM are only used to enhance the feature discrimination of images rather than to detect actual mapping.

time is limited (+7.5% for training and +1.6% for inference). During training, the proposed USAM helps the model extract discriminative representations that are robust to viewpoint variants. The representation learning, in turn, encourages the model to extract more accurate keypoints in the feature map. *That is, RK-Net is gradually improved with the interaction of two related tasks.* In Fig. 5, we provide an example of detecting keypoints using USAM. We can observe that USAM can extract the salient keypoints from images of different views. More visualizations can be found in Sec. IV-E

## IV. EXPERIMENT

### A. Datasets and Evaluation Protocol

We conduct experiments on three datasets, *i.e.*, University-1652 [18], CVUSA [20] and CVACT [3].

**University-1652** [18] is a multi-view multi-source dataset, including satellite-view images, drone-view images and ground-view images collected from three different platforms. Instead of selecting landmarks as the target locations, the dataset selects 1,652 ordinary architectures of 72 universities around the world as targets. Overall, every building has 1 satellite-view image, 54 drone-view images, and 3.38 real street-view images on average. There are 701 architectures of 33 universities in the training set, while the other 951 architectures of 39 universities form the testing set. There is no overlap between universities of the training and testing sets. The dataset can be used to evaluate two new tasks, *i.e.*, drone-view target localization (Drone  $\rightarrow$  Satellite) and drone navigation (Satellite  $\rightarrow$  Drone). **CVUSA** [20] & **CVACT** [3] are both large-scale datasets, and each dataset contains 35,532 ground and satellite training image pairs. A total of 8,884 cross-view image pairs are provided for testing in CVUSA and validating in CVACT (denoted as CVACT\_val). Moreover, CVACT also provides 92,802 image pairs for testing (denoted as CVACT\_test). Note that all ground images of the dataset are panoramas, and both street-view and satellite-view samples are high-resolution images.

**Evaluation Protocol.** We adopt the recall accuracy at top  $K$  (Recall@K) and the average precision (AP) as the evaluation

metrics to evaluate the model performance. The value of Recall@ $K$  is 1 if positive images appear in the top  $K$  of the ranking list. In this paper, we evaluate  $K = 1$  for University-1652 and  $K = 1$  and  $K = \text{Top}1\%$  for CVUSA and CVACT, where Top1% indicates the top 1% samples of the ranking list. The average precision represents the area under the Precision-Recall curve. We report the mean AP (mAP) over all queries.

### B. Implementation Details

**Model Detail.** We employ ResNet-50 [72] with pretrained weights on ImageNet [34] as our backbone network, which has 5 stages in total. By default, we set the kernel size of our USAM to  $3 \times 3$  and insert USAM after stage 1 and stage 2, respectively. Following [18], we add a 512-dimensional fully connected layer and a classification layer to replace the original classifier dedicated for ImageNet and initialize the new classifier with kaiming initialization [74].

**Training Detail.** We resize the input images to a size of  $256 \times 256$  for both the training and testing phases. During training, we adopt horizontal flipping and random cropping as data augmentation. Following [24], we use polar transform for the CVUSA and CVACT datasets but not for the University-1652 dataset. Examples are shown in Fig. 6. The stochastic gradient descent optimizer (SGD) with momentum=0.9 and weight decay=0.0005 is employed to update the model. We train the network for 360 epochs with a mini-batch of 16 in total. For the learning rate, we use 0.01 and 0.001 for the newly added layers and the original backbone layers, respectively. The learning rate decays by 0.1 after 200 epochs for all layers. The dropout rate is 0.65. During testing, we extract the output of the pooling-5 layer as the feature and use the Euclidean distance to measure the similarity between the query image and gallery images.



Fig. 6: Aerial examples for University-1652, CVUSA and CVACT. Polar transform is only applied to CVUSA and CVACT.

### C. Comparison with State-of-the-art Methods

**Results on University-1652.** We first compare our method with state-of-the-art methods on University-1652, including approaches with different losses (*i.e.*, Contrastive Loss [1], Triplet Loss [75], Soft Margin Triplet Loss [38] and Instance Loss [18]), LCM [76], SAFA [77] and LPN [24]. LPN [24] explicitly considers the local information during training. The

TABLE I: Comparison with state of the art on University-1652.  $M$  denotes the margin of the triplet loss. w/ G indicates use of the extra Google Image set during training. Drone→Satellite denotes the drone-view target localization task, and Satellite→Drone indicates the drone navigation task.

Method	Drone→Satellite		Satellite→Drone	
	R@1	mAP	R@1	mAP
Contrastive Loss [1] [18]	52.39	57.44	63.91	52.24
Triplet Loss ( $M = 0.3$ ) [75] [18]	55.18	59.97	63.62	53.85
Triplet Loss ( $M = 0.5$ ) [75] [18]	53.58	58.6	64.48	53.15
Soft Margin Triplet Loss [38] [18]	53.21	58.03	65.62	54.47
Instance Loss [18]	58.49	63.13	71.18	58.74
LCM [76]	66.65	70.82	79.89	65.38
SAFA [77]	68.27	72.06	80.16	68.11
SAFA (w/ G) [77]	69.34	73.15	82.60	69.78
LPN [24]	74.18	77.39	85.16	73.68
LPN (w/ G) [24]	75.93	79.14	86.45	74.79
Instance Loss + USAM	<b>65.63</b>	<b>69.68</b>	<b>78.32</b>	<b>64.87</b>
Instance Loss + USAM (w/ G)	<b>66.13</b>	<b>70.23</b>	<b>80.17</b>	<b>65.76</b>
SAFA+USAM	<b>70.89</b>	<b>74.56</b>	<b>82.88</b>	<b>70.20</b>
SAFA+USAM (w/ G)	<b>72.19</b>	<b>75.79</b>	<b>83.23</b>	<b>71.77</b>
LPN+USAM	<b>77.07</b>	<b>80.09</b>	<b>85.16</b>	<b>74.06</b>
LPN+USAM (w/ G)	<b>77.60</b>	<b>80.55</b>	<b>86.59</b>	<b>75.96</b>

TABLE II: Comparison with competitive methods on CVUSA & CVACT\_val. \*:using extra orientation information. +:using transformer-based backbone.

Method	CVUSA		CVACT_val	
	R@1	R@Top1%	R@1	R@Top1%
Hybrid-Net [50]	45.22	92.32	36.89	87.81
CVM-Net [38]	18.80	91.54	20.15	87.57
VGG global pooling [21]	31.53	95.09	28.98	91.72
Orientation* [3]	40.79	96.08	46.96	92.01
Instance Loss [18]	43.91	91.78	35.24	87.34
Regmi <i>et al.</i> [22]	48.75	95.98	-	-
Siam-FCANet34 [51]	-	98.30	-	-
CVFT [21]	61.43	99.02	61.05	95.93
Rodrigues [54]	75.95	99.42	73.19	97.45
LPN [24]	85.79	99.41	79.99	97.03
SAFA [77]	89.84	99.64	81.03	98.17
DSM [36]	91.96	99.67	82.49	97.32
Polar-L2LTR+ [55]	94.05	99.67	84.89	98.37
Instance Loss+USAM	<b>52.50</b>	<b>96.52</b>	<b>40.53</b>	<b>89.12</b>
SAFA+USAM	<b>90.16</b>	<b>99.67</b>	<b>82.40</b>	<b>98.00</b>
LPN+USAM	<b>91.22</b>	<b>99.67</b>	<b>82.02</b>	<b>98.18</b>

comparison results are reported in Table I. It is noted that the results of those methods that are anterior to the dataset publication are borrowed from [18] except SAFA. For SAFA, we reimplement it on the University-1652 dataset with the provided source code. We can make the following observations. First, without explicitly considering the local information, the baseline model with Instance Loss [18] produces the best results. Second, when adding our USAM, the results of Instance Loss are significantly improved. For example, without using the extra Google training set, “Instance Loss + USAM” achieves 65.63% in Recall@1 accuracy and 69.68% in mAP for “Drone → Satellite” and 78.32% in Recall@1 accuracy and 64.87% in mAP for “Satellite → Drone”. This is clearly 6.55% higher than the Instance Loss for “Drone → Satellite” and 6.13% higher for “Satellite → Drone” in terms of mAP. Our method also benefits from SAFA [77]. For example, “SAFA + USAM” boosts R@1 accuracy from 69.34% to 72.19%

(+2.85%) in the drone-view target localization task (Drone → Satellite) when using the extra Google training set. Third, LPN [24] produces largely higher results than other methods. Nonetheless, when combining our USAM with LPN [24], we can obtain further improvement. For example, when using extra Google data, “LPN [24]+Ours” improves the mAP from 79.14% to 80.55% for “Drone → Satellite” and from 74.79% to 75.96% for “Satellite → Drone”. The above observations demonstrate the effectiveness of the proposed USAM and that USAM is a flexible module that can be embedded in different models to improve performance.

**Results on CVUSA & CVACT.** We also compare our approach with other competitive methods on the CVUSA and CVACT datasets. As shown in Table II and Table III, we observe a similar phenomenon as for the results on University-1652. That is, 1) LPN [24] surpasses most CNN-based methods by a large margin; 2) our USAM can consistently improve the results of Instance Loss [18], SAFA [77] and LPN [24]. Specifically, for CVUSA, when injecting USAM into the network, the Recall@1 accuracy is improved from 43.91% to 52.50% for Instance Loss [18], from 89.84% to 90.16% for SAFA [77] and from 85.79% to 91.99% for LPN [24]. Similarly, improvement is also observed in CVACT (on the validation set, 35.24% to 40.53% for Instance Loss [18], 81.03% to 82.40% for SAFA [77] and 79.99% to 82.02% for LPN [24]); 3) with LPN, our framework (“LPN+USAM”) produces very competitive results, which surpass those of SAFA [77] by 1.38% and 0.99% in recall@1 accuracy for CVUSA and CVACT\_val, respectively. In addition, ‘LPN+USAM’ obtains better R@Top1% accuracy than DSM [36] on CVACT\_val and CVACT\_test. (4)our method largely outperforms Hybrid-Net [50], which uses a spatial transformer as the attention module, when using the instance loss as the baseline. Note that although Polar-L2LTR [55] achieves the best performance, it uses a Transformer-based backbone that is stronger than ResNet-50. Therefore, it is unfair to directly compare Polar-L2LTR with existing CNN-based models. The above results support the effectiveness and flexibility of our USAM under different settings.

TABLE III: Results on the test set of CVACT. \*:using extra orientation information. +:using transformer-based backbone.

Method	CVACT_test	
	R@1	R@Top1%
CVM-Net [38]	5.41	54.53
Instance loss [18]	11.25	52.42
Orientation* [3]	19.21	60.69
CVFT [21]	26.12	71.69
LPN [24]	35.03	84.27
DSM [36]	35.63	84.75
SAFA [77]	55.50	94.49
Polar-L2LTR+ [55]	60.72	96.12
Instance loss+USAM	<b>13.42</b>	<b>55.69</b>
LPN+USAM	<b>37.71</b>	<b>87.04</b>
SAFA+USAM	<b>56.16</b>	<b>95.22</b>

#### D. Evaluation

**Effect of injecting USAM into different stages.** We regard

TABLE IV: Effect of adding USAM into different stages of the ResNet-50 network.

Stage	Drone→Satellite R@1	Satellite→Drone mAP	Drone→Satellite R@1	Satellite→Drone mAP
w/o USAM	58.49	63.13	71.18	58.74
1	65.63	69.97	78.55	64.91
2	65.59	69.69	78.23	64.50
3	64.92	68.01	77.92	64.03
4	62.73	66.98	75.41	62.74
5	60.43	64.87	73.28	60.17
1+2	<b>66.13</b>	<b>70.23</b>	<b>80.17</b>	<b>65.76</b>
1+2+3	63.75	68.32	78.22	64.43
1+2+3+4	63.18	67.42	77.83	63.72
1+2+3+4+5	59.94	64.34	74.19	59.31

TABLE V: Effect of using different kernel sizes for USAM.

Kernel Sizes	Drone→Satellite R@1	Satellite→Drone mAP	Drone→Satellite R@1	Satellite→Drone mAP
w/o USAM	58.49	63.13	71.18	58.74
3 × 3	<b>66.13</b>	<b>70.23</b>	<b>80.17</b>	<b>65.76</b>
5 × 5	63.28	67.62	76.75	63.51
7 × 7	61.11	65.46	75.61	61.90
9 × 9	60.59	64.79	74.32	60.58

the five residual blocks of ResNet-50 as 5 stages. In Table IV, we investigate the impact of injecting USAM into different stages. We first evaluate the performance by embedding USAM after only one of the five stages. We find that injecting USAM into any stage of the network can improve the results of the baseline (“w/o USAM”). Specifically, injecting USAM after a relatively shallow stage (*i.e.*, stage 1, 2 and 3) can achieve higher performance. The main reason for this is that the deep stages (*i.e.*, stage 4 and 5) mainly contain high-level semantic information, which is not suitable to detect keypoints. We then study the effect of adding USAM into multiple stages. The best results are achieved by jointly injecting USAM after stage 1 and stage 2. Applying USAM to deeper stages hampers the results. In our experiments, we use the same training setting for corresponding CNN architectures for a fair comparison, whether or not we apply the USAM module.

**Sensitivity to the kernel size.** The kernel size is an important parameter of our USAM. To find the appropriate kernel size, we compare the results of using different kernel sizes for USAM. Table V shows that the best results are obtained when kernel size = 3 × 3. Using a larger kernel size reduces the performance.

**Ablation study on different components in USAM.** There are two components in our USAM, *i.e.*, Unit Subtraction Convolution (USC) and Batch Normalization (BN). In Table VI, we study the effectiveness of these two components by removing one of them from USAM. When using only BN, we directly apply BN to the feature map. When using only USC, we use the Min-Max Normalization to scale the generated attention mask. We observe that removing each of them largely reduces the performance, especially USC. This demonstrates the effectiveness of the proposed USC and shows

TABLE VI: Ablation study on different components in the proposed USAM.

Operation	BN	USC	Drone→Satellite R@1	Satellite→Drone mAP	Drone→Satellite R@1	Satellite→Drone mAP
-	-	-	58.49	63.13	71.18	58.74
✓	-	-	59.39	63.79	73.32	59.82
-	✓	-	62.89	67.4	76.03	63.66
✓	✓	-	<b>66.13</b>	<b>70.23</b>	<b>80.17</b>	<b>65.76</b>

TABLE VII: Effect of different feature aggregation functions.

Feature Aggregation	Drone→Satellite R@1	Satellite→Drone mAP	Drone→Satellite R@1	Satellite→Drone mAP
w/o Pooling	59.94	64.34	73.05	60.00
Max-Pooling	63.44	67.57	77.89	64.03
Average-Pooling	65.47	69.57	79.74	65.28
Sum-Pooling	<b>66.13</b>	<b>70.23</b>	<b>80.17</b>	<b>65.76</b>

TABLE VIII: Evaluation of different attention fusion strategies.  $M$  is the keypoint attention map.  $F_l$  and  $F_l^*$  are the input feature map and output feature map of USAM, respectively.  $c$  is the channel size of  $F_l$ .

Fusion Strategy	Drone→Satellite R@1	Satellite→Drone mAP	Drone→Satellite R@1	Satellite→Drone mAP
Non: $F_l^* = F_l$	58.49	63.13	71.18	58.74
Mul: $F_l^* = F_l \cdot \text{Expand}(M, c)$	54.19	58.83	68.05	55.50
Add: $F_l^* = F_l + \text{Expand}(M, c)$	65.13	69.33	77.89	64.99
Res: $F_l^* = F_l + F_l \cdot \text{Expand}(M, c)$	<b>66.13</b>	<b>70.23</b>	<b>80.17</b>	<b>65.76</b>

that the improvement of our USAM mainly depends on the proposed USC rather than batch normalization.

**Effect of different feature aggregation functions.** In our USAM, we use sum-pooling to aggregate the feature map (Eq. 4). To study the effectiveness of this strategy, we compare different aggregation methods in Table VII. The performance confirms that 1) it is necessary to aggregate the feature map and 2) feature aggregation with sum-pooling achieves the best results compared to max-pooling and average-pooling.

**Effect of the attention fusion strategy.** To verify the effect of the proposed residual attention fusion, we compare it with two other fusion strategies, namely, the multiplication fusion strategy and the addition fusion strategy. For the multiplication fusion strategy, we directly use the weighted feature map as the new representation for the next stage. For the addition fusion strategy, we add the produced keypoint attention mask to the original feature map as the new representation. The details and results of these two fusion strategies are reported in Table VIII. We find that it is important to retain the original feature map. When only using the weighted feature map (“Mul”), the results are clearly decreased. In addition, it is better to apply the generated keypoint attention map to produce a weighted feature map (“Res”) instead of directly adding it with the original feature map (“Add”).

**Comparison with different forms of attention generation methods.** In our USC, we use the subtraction operation to generate attention. To verify its effectiveness, we additionally compare it with four attention generation operations as studied

TABLE IX: Comparison with different forms of attention generation methods in USC on University-1652.

Attention Generation	Drone→Satellite		Satellite→Drone	
	R@1	mAP	R@1	mAP
Baseline	58.49	63.13	71.18	58.74
Addition	62.52	66.71	73.61	62.19
Hadamard Product	63.26	67.27	76.03	64.71
Dot Product	63.37	67.48	77.89	66.11
Concat+Mean	62.99	67.25	76.89	64.69
Concat+Max	63.98	67.98	76.18	64.14
Subtraction	<b>66.13</b>	<b>70.23</b>	<b>80.17</b>	<b>65.76</b>

TABLE X: Impact of input image size on University-1652.

Image Size	Drone→ Satellite		Satellite→Drone	
	R@1	mAP	R@1	mAP
224*224	63.96	69.19	77.89	64.76
256*256	66.13	70.23	80.17	65.76
320*320	<b>69.08</b>	<b>72.90</b>	<b>82.45</b>	<b>69.91</b>
384*384	68.05	71.97	80.74	69.17
512*512	68.10	71.53	80.96	69.35

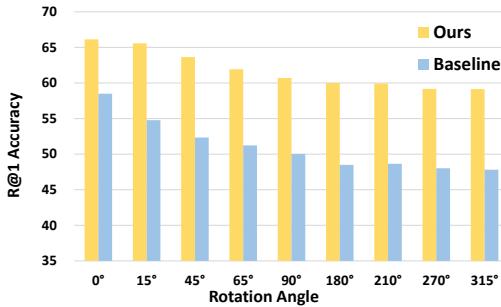


Fig. 7: Evaluation of rotating images on “Drone → Satellite” of University-1652.

in [78], including addition, Hadamard product, dot product and concatenation. To maintain the dimension of the generated attention mask, we use the mean operation or max operation after concatenation (denoted as Concat-Mean and Concat-Max, respectively). From the results in Table IX, we can observe that (1) all the attention operations can boost the model performance and (2) the subtraction operation used in our USC achieves the best improvement for both settings. This shows that our USAM is more suitable for the cross-view geolocation task.

**Effect of the input image size.** To evaluate the effect of image size, we maintain the ratio between width and height as 1:1 and vary the input size from  $224 \times 224$  to  $512 \times 512$ . The results in Table X show that (1) increasing the image size from  $224 \times 224$  to  $320 \times 320$  can clearly improve the performance and (2) assigning an input size that is too large (*e.g.*,  $512 \times 512$ ) leads to a reduction. On the other hand, using a large image size exponentially increases the computational cost. Therefore, considering the balance between speed and accuracy, we adopt  $256 \times 256$  as the input size in our experiments.

**Effect of rotating images.** We conduct experiments to study the effect of rotating images. In this experimentation, we only rotate the query images and keep the gallery images

TABLE XI: Evaluation of computation costs. # **Params**: number of parameters, **FLOPs**: floating point operations.

Method	# Params (M)	FLOPs (G)	Drone→Satellite	
			R@1	mAP
Instance loss [18]	48.426	24.425	58.49	63.13
Instance loss + Ours	<b>48.426</b>	<b>24.425</b>	<b>65.63</b>	<b>69.68</b>
LPN [24]	52.655	24.436	74.18	77.39
LPN + Ours	<b>52.655</b>	<b>24.436</b>	<b>77.07</b>	<b>80.09</b>

unchanged. We rotate the query images between  $0^\circ$  to  $315^\circ$ . A comparison between our method and the baseline is shown in Fig. 7. We can make the following observations. (1) The results of both methods decrease with increasing rotating degree. (2) Our method consistently outperforms the baseline in all rotating cases. (3) The performance reduction of the baseline exceeds that of our model after rotating. For instance, the R@1 accuracy is reduced by 10% for the baseline and 7% for our method, when rotating the images by  $315^\circ$ . The above observations verify that our method is more robust to rotating variations.

**Computation cost of USAM.** To verify the lightweight property of USAM, we conduct experiments by calculating the number of parameters (# Params) and floating point operations (FLOPs) of the network. The results in Table XI show that our USAM can significantly improve the performance without increasing the computation costs. Indeed, our USAM only introduces very few learnable parameters that are produced by BN layers. The extra computation costs are negligible compared to the parameters of the overall network.

### E. Visualization

To better understand our USAM, we also present one qualitative experiment on visualizing keypoint heatmaps generated by USAM. Specifically, we visualize the keypoint attention mask generated by USC, which comes from the USAM module behind stage 2 (S2). The generated attention mask is scaled to  $[0, 1]$  by the min-max normalization. We then use the same threshold (0.4) for the baseline model [18] and our model to produce the visualization of keypoints. This operation ensures a fair comparison between the baseline and our RK-Net. **(A)** In Fig. 8 (a), we provide the visualization of keypoint heatmaps from satellite views (first row) and drone views (second row). We can observe that our method can effectively extract keypoints from discriminative regions of buildings / locations for different view images, regardless of the target scale in the image. **(B)** Then, in Fig. 8 (b), we show the visualization results of the satellite-view image and drone-view images from different shooting angles for the same object. The figures show that our method can still discover remarkable keypoints despite the appearance changes caused by rotation variants. **(C)** In Fig. 8 (c), we show the keypoint heatmaps generated in different epochs. The results illustrate that with the increase of training, our model can extract an increasing number of significant keypoints from discriminative regions, *e.g.*, the roof, and reduce disturbance from the “nonspecial” area, *e.g.*, greensward. This demonstrates the mutual benefit of joint representation learning and keypoint detection. **(D)**

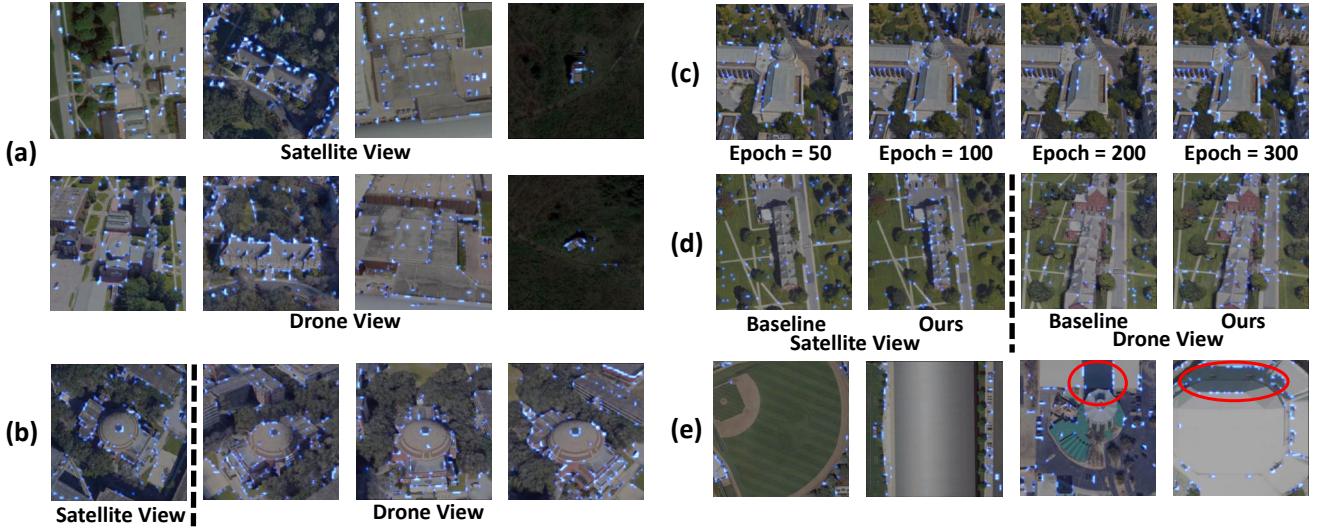


Fig. 8: Visualization of keypoints produced by our USAM. (a) Results from different views. (b) Results from different angles of the drone camera. (c) Results generated in different epochs. (d) Comparison of results between baseline [18] and our method. (e) Failure results.

We also compare the keypoint heatmaps generated by the baseline model [18] and our method. As shown in Fig. 8 (d), our method can extract more salient keypoints and pay more attention to the target building, while the baseline fails to focus on discriminative regions. It should be highlighted that the number of keypoints is almost the same between the baseline and our method, which illustrates that the improvement of our method is obtained from learning with keypoints of the structure region instead of merely using more keypoints. (E) In addition, we show some failure cases of our method in Fig. 8 (e). We observe that our module is unable to perform well in the following two cases: 1) the image texture is smooth and does not exhibit obvious change; 2) there are shadows caused by buildings in the images. For the first case, as shown in the first and second images of Fig. 8 (e), the proposed method cannot mine sufficient keypoints for discrimination. On the other hand, for the second case, as shown in the red areas of the third and fourth images of Fig. 8 (e), our method mistakes the shadows of buildings as part of the target and extracts keypoints from the regions under the shadow. (F) We show examples of different thresholds in Fig. 9. We can observe that our RK-Net consistently discovers more discriminative keypoints than the baseline under different thresholds. (G) Finally, we show an animation from the drone view in Fig. 10, which is composed of images of the same architecture from different shooting angles. The animation further verifies that our model can discover view-invariant keypoints.

## V. FURTHER ANALYSIS AND DISCUSSION

### A. Connection to Human Visual System (HVS)

We further elaborate the relevance between our motivation and HVS in this subsection. In practice, humans commonly recognize objects by a coarse-to-fine process [17], which inspires us to consider both global and local information

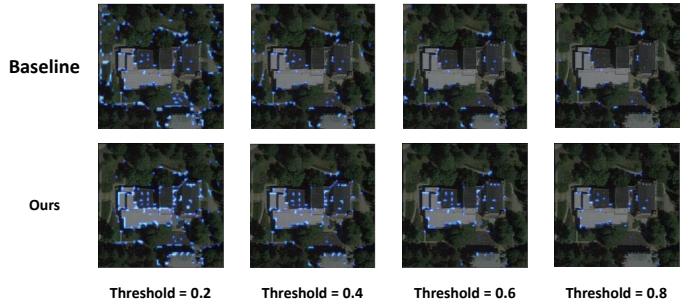


Fig. 9: Visualizations of generating keypoints with different thresholds.

Fig. 10: Frame-to-frame keypoint animation detected by our USAM from a drone view (click on the image to see animation using Adobe Reader). We note that the input video clip is from the test set, and that the model has not seen the building before.

during representation learning. Although we do not explicitly divide the matching process into global comparison and local comparison, our designed model fuses both the global features (coarse) and local features (fine), enabling us to potentially

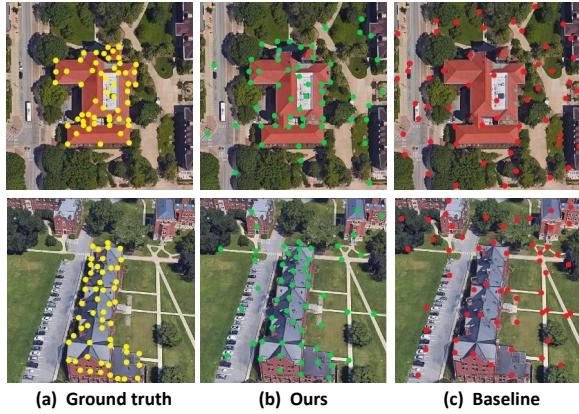


Fig. 11: Keypoint comparisons between the baseline approach [18] and our method. The first row images are satellite-view, and the second row images are drone-view. (a) are the ground truth keypoint labels. (b) and (c) are predicted keypoints from our model and the baseline [18].

perform the coarse-to-fine matching process. Generally, the general properties of images, which are extracted by the backbone network, can be viewed as global features with coarse-grained information and are dominant in the matching process. If two images are similar, it is difficult to identify them only by using the general properties. Therefore, humans further excavate keypoints with discriminative fine-grained information to differentiate two images. In our work, we extract the keypoints with fine-grained features by the process of Unit Subtraction Convolution (USC), which can play a key role in fine-grained comparison. Finally, by pooling operation and Unit Subtraction Attention Module (USAM), our designed model can fuse both the global features and local features to perform a coarse-to-fine matching process.

### B. Keypoint Evaluation

To further validate the effectiveness of USAM, we conduct extra experiments that produce the same number of keypoints (60 top-ranked keypoints) for the baseline and our method and add a quantitative evaluation of keypoints. Specifically, we invite several experts to annotate the keypoints for the testing set of University-1652 and collect 50 annotated samples for validation. We use the percentage of correct keypoints (PCK) metric [79] to evaluate the performance of keypoint detection. Our method achieves 77.6% in PCK accuracy, outperforming the baseline by 34.9%. We also show some qualitative examples for the keypoint annotation between our method and the baseline in Fig. 11. The figure shows that the keypoints extracted by our method are closer to the ground truth.

## VI. CONCLUSION

In this work, we introduce the RK-Net framework, which explores joint learning in keypoint detection and representation learning for cross-view geo-localization. The main idea underpinning RK-Net is to find salient regions to discriminate

between different locations, which is also aligned with the human visual system. In our RK-Net, we propose a novel Unit Subtraction Module (USAM) to automatically mine remarkable keypoints from feature maps for extracting viewpoint-invariant representations. Extensive experiments show that our method can achieve competitive results on three benchmarks, *i.e.*, University-1652, CVUSA and CVACT. In the future, we will investigate other potential applications, such as vehicle re-identification, product retrieval and other fine-grained retrieval tasks.

## REFERENCES

- [1] T.-Y. Lin, Y. Cui, S. Belongie, and J. Hays, “Learning deep representations for ground-to-aerial geolocalization,” in *Proc. CVPR*, 2015, pp. 5007–5015. [1](#), [2](#), [6](#), [7](#)
- [2] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, “Netvlad: Cnn architecture for weakly supervised place recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, pp. 1437–1451, 2018. [1](#)
- [3] L. Liu and H. Li, “Lending orientation to neural networks for cross-view geo-localization,” in *Proc. CVPR*, 2019, pp. 5617–5626. [1](#), [2](#), [6](#), [7](#)
- [4] Y. Tian, C. Chen, and M. Shah, “Cross-view image matching for geolocation in urban environments,” in *Proc. CVPR*, 2017, pp. 1998–2006. [1](#), [2](#), [3](#)
- [5] P. Zhu, J. Zheng, D. Du, L. Wen, Y. Sun, and Q. Hu, “Multi-drone based single object tracking with agent sharing network,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, pp. 4058–4070, 2020. [1](#)
- [6] V. J. Hodge, R. Hawkins, and R. Alexander, “Deep reinforcement learning for drone navigation using sensor data,” *Neural Computing and Applications*, pp. 1–19, 2020. [1](#)
- [7] H. Wei and L. Wang, “Visual navigation using projection of spatial right-angle in indoor environment,” *IEEE Transactions on Image Processing*, vol. 27, pp. 3164–3177, 2018. [1](#)
- [8] D.-G. Sim, S.-Y. Jeong, D.-H. Lee, R.-H. Park, R.-C. Kim, S. U. Lee, and I. C. Kim, “Hybrid estimation of navigation parameters from aerial image sequence,” *IEEE Transactions on Image Processing*, vol. 8, pp. 429–35, 1999. [1](#)
- [9] Y. Ke, R. Sukthankar, and M. Hebert, “Event detection in crowded videos,” in *Proc. ICCV*, 2007, pp. 1–8. [1](#)
- [10] J. Choi, G. Sharma, M. Chandraker, and J.-B. Huang, “Unsupervised and semi-supervised domain adaptation for action recognition from drones,” in *Proc. WACV*, 2020, pp. 1706–1715. [1](#)
- [11] A. M. Algamdi, V. Sanchez, and C.-T. Li, “Dronecaps: Recognition of human actions in drone videos using capsule networks with binary volume comparisons,” in *Proc. ICIP*, 2020, pp. 3174–3178. [1](#)
- [12] Y. Yan, Y. Yang, D. Meng, G. Liu, W. Tong, A. G. Hauptmann, and N. Sebe, “Event oriented dictionary learning for complex event detection,” *IEEE Transactions on Image Processing*, vol. 24, pp. 1867–1878, 2015. [1](#)
- [13] S. Deng, S. Li, K. Xie, W. Song, X. Liao, A. Hao, and H. Qin, “A global-local self-adaptive network for drone-view object detection,” *IEEE Transactions on Image Processing*, vol. 30, pp. 1556–1569, 2021. [1](#)
- [14] H. Zhang, G. Wang, Z. Lei, and J.-N. Hwang, “Eye in the sky: Drone-based object tracking and 3d localization,” in *Proc. ACM MM*, 2019, pp. 899–907. [1](#)
- [15] T. Peng, Q. Li, and P. Zhu, “Rgb-t crowd counting from drone: A benchmark and mmccn network,” in *Proc. ACCV*, 2020, pp. 497–513. [1](#)
- [16] M. Modsching, R. Kramer, and K. ten Hagen, “Field trial on gps accuracy in a medium size city: The influence of built-up,” in *3rd workshop on positioning, navigation and communication*, 2006, pp. 209–218. [1](#)
- [17] S. Kastner and L. G. Ungerleider, “Mechanisms of visual attention in the human cortex,” *Annual Review of Neuroscience*, vol. 23, pp. 315–341, 2000. [1](#), [10](#)
- [18] Z. Zheng, Y. Wei, and Y. Yang, “University-1652: A multi-view multi-source benchmark for drone-based geo-localization.” in *Proc. ACM MM*, 2020, pp. 1395–1403. [1](#), [2](#), [3](#), [5](#), [6](#), [7](#), [9](#), [10](#), [11](#)
- [19] S. Workman and N. Jacobs, “On the location dependence of convolutional neural network features,” in *Proc. CVPRW*, 2015, pp. 70–78. [1](#), [2](#)

- [20] S. Workman, R. Souvenir, and N. Jacobs, "Wide-area image geolocalization with aerial reference imagery," in *Proc. ICCV*, 2015, pp. 3961–3969. [1](#), [2](#), [6](#)
- [21] Y. Shi, X. Yu, L. Liu, T. Zhang, and H. Li, "Optimal feature transport for cross-view image geo-localization," in *Proc. AAAI*, 2020, pp. 11990–11997. [1](#), [2](#), [7](#)
- [22] K. Regmi and M. Shah, "Bridging the domain gap for ground-to-aerial image matching," in *Proc. ICCV*, 2019, pp. 470–479. [1](#), [2](#), [7](#)
- [23] A. Toker, Q. Zhou, M. Maximov, and L. Leal-Taixé, "Coming down to earth: Satellite-to-street view synthesis for geo-localization," in *Proc. CVPR*, 2021, pp. 6484–6493. [1](#), [2](#)
- [24] T. Wang, Z. Zheng, C. Yan, and Y. Yang, "Each part matters: Local patterns facilitate cross-view geo-localization," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, pp. 867–879, 2022. [1](#), [2](#), [6](#), [7](#), [9](#)
- [25] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. ICCV*, 1999, pp. 1150–1157. [1](#), [3](#)
- [26] L.-C. Chiu, T.-S. Chang, J.-Y. Chen, and N. Y.-C. Chang, "Fast sift design for real-time visual feature extraction," *IEEE Transactions on Image Processing*, vol. 22, pp. 3158–3167, 2013. [1](#)
- [27] W.-L. Zhao and C.-W. Ngo, "Flip-invariant sift for copy and object detection," *IEEE Transactions on Image Processing*, vol. 22, pp. 980–991, 2013. [1](#)
- [28] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 971–987, 2002. [1](#)
- [29] G. Zhao, T. Ahonen, J. Matas, and M. Pietikainen, "Rotation-invariant image and video description with local binary pattern features," *IEEE Transactions on Image Processing*, vol. 21, pp. 1465–1477, 2012. [1](#)
- [30] M. Enzweiler and D. M. Gavrila, "A multilevel mixture-of-experts framework for pedestrian classification," *IEEE Transactions on Image Processing*, vol. 20, pp. 2967–2979, 2011. [1](#)
- [31] F. Castaldo, A. Zamir, R. Angst, F. Palmieri, and S. Savarese, "Semantic cross-view matching," in *Proc. ICCVW*, 2015, pp. 1044–1052. [2](#)
- [32] T.-Y. Lin, S. Belongie, and J. Hays, "Cross-view image geolocalization," in *Proc. CVPR*, 2013, pp. 891–898. [2](#)
- [33] M. Bansal, H. S. Sawhney, H. Cheng, and K. Daniilidis, "Geolocalization of street views with aerial image databases," in *Proc. ACM MM*, 2011, pp. 1125–1128. [2](#)
- [34] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2015. [2](#), [6](#)
- [35] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Proc. NeurIPS*, 2014, pp. 487–495. [2](#)
- [36] Y. Shi, X. Yu, D. Campbell, and H. Li, "Where am i looking at? joint location and orientation estimation by cross-view matching," in *Proc. CVPR*, 2020, pp. 4063–4071. [2](#), [7](#)
- [37] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. NeurIPS*, 2014, pp. 2672–2680. [2](#)
- [38] S. Hu, M. Feng, R. M. H. Nguyen, and G. H. Lee, "Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *Proc. CVPR*, 2018, pp. 7258–7267. [2](#), [6](#), [7](#)
- [39] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. CVPR*, 2006, pp. 1735–1742. [2](#)
- [40] L. Wei, S. Zhang, W. Gao, and Q. Tian, "Person transfer gan to bridge domain gap for person re-identification," in *Proc. CVPR*, 2018, pp. 79–88. [2](#)
- [41] 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*, vol. 16, pp. 1–23, 2020. [2](#), [5](#)
- [42] S. Li, S. Bak, P. Carr, and X. Wang, "Diversity regularized spatiotemporal attention for video-based person re-identification," in *Proc. CVPR*, 2018, pp. 369–378. [2](#)
- [43] W. Li, X. Zhu, and S. Gong, "Harmonious attention network for person re-identification," in *Proc. CVPR*, 2018, pp. 2285–2294. [2](#)
- [44] L. Wei, S. Zhang, H. Yao, W. Gao, and Q. Tian, "Glad: Global-local-alignment descriptor for scalable person re-identification," *IEEE Transactions on Multimedia*, vol. 21, no. 4, pp. 986–999, 2018. [2](#)
- [45] L. Wei, X. Liu, J. Li, and S. Zhang, "Vp-reid: Vehicle and person re-identification system," in *Proc. ACM ICMLR*, 2018, pp. 501–504. [2](#)
- [46] J. Xu, R. Zhao, F. Zhu, H. Wang, and W. Ouyang, "Attention-aware compositional network for person re-identification," in *Proc. CVPR*, 2018, pp. 2119–2128. [2](#)
- [47] J. Fu, H. Zheng, and T. Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *Proc. CVPR*, 2017, pp. 4476–4484. [2](#)
- [48] T. Xiao, Y. Xu, K. Yang, J. Zhang, Y. Peng, and Z. Zhang, "The application of two-level attention models in deep convolutional neural network for fine-grained image classification," in *Proc. CVPR*, 2015, pp. 842–850. [2](#)
- [49] B. Zhao, X. Wu, J. Feng, Q. Peng, and S. Yan, "Diversified visual attention networks for fine-grained object classification," *IEEE Transactions on Multimedia*, vol. 19, pp. 1245–1256, 2017. [2](#)
- [50] H. Altwaijry, E. Trulls, J. Hays, P. Fua, and S. Belongie, "Learning to match aerial images with deep attentive architectures," in *Proc. CVPR*, 2016, pp. 3539–3547. [2](#), [3](#), [7](#)
- [51] S. Cai, Y. Guo, S. Khan, J. Hu, and G. Wen, "Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss," in *Proc. ICCV*, 2019, pp. 8390–8399. [2](#), [3](#), [7](#)
- [52] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. NeurIPS*, 2015, pp. 2017–2025. [2](#)
- [53] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 1137–1149, 2017. [3](#)
- [54] R. Rodrigues and M. Tani, "Are these from the same place? seeing the unseen in cross-view image geo-localization," in *Proc. WACV*, 2021, pp. 3752–3760. [3](#), [7](#)
- [55] H. Yang, X. Lu, and Y. Zhu, "Cross-view geo-localization with layer-to-layer transformer," in *Proc. NIPS*, 2021, pp. 1–12. [3](#), [7](#)
- [56] C. Harris, M. Stephens *et al.*, "A combined corner and edge detector," in *Proc. AVC*, 1988, pp. 10–5244. [3](#)
- [57] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. ECCV*, 2006, pp. 430–443. [3](#)
- [58] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer vision and image understanding*, vol. 110, pp. 346–359, 2008. [3](#)
- [59] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image and vision computing*, vol. 22, pp. 761–767, 2004. [3](#)
- [60] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "Kaze features," in *Proc. ECCV*, 2012, pp. 214–227. [3](#)
- [61] P. F. Alcantarilla and T. Solutions, "Fast explicit diffusion for accelerated features in nonlinear scale spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 7, pp. 1281–1298, 2011. [3](#)
- [62] Y. Verdie, K. Yi, P. Fua, and V. Lepetit, "Tilde: A temporally invariant learned detector," in *Proc. CVPR*, 2015, pp. 5279–5288. [3](#)
- [63] K. Lenc and A. Vedaldi, "Learning covariant feature detectors," in *Proc. ECCV*, 2016, pp. 100–117. [3](#)
- [64] D. DeTone, T. Malisiewicz, and A. Rabinovich, "Superpoint: Self-supervised interest point detection and description," in *Proc. CVPRW*, 2018, pp. 224–236. [3](#)
- [65] N. Savinov, A. Seki, L. Ladicky, T. Sattler, and M. Pollefeys, "Quad-networks: unsupervised learning to rank for interest point detection," in *Proc. CVPR*, 2017, pp. 3929–3937. [3](#)
- [66] Y. Ono, E. Trulls, P. Fua, and K. M. Yi, "Lf-net: learning local features from images," in *Proc. NIPS*, 2018, pp. 6237–6247. [3](#)
- [67] J. Carranza-Rojas, S. Calderon-Ramirez, A. Mora-Fallas, M. Granados-Menani, and J. Torrents-Barrena, "Unsharp masking layer: injecting prior knowledge in convolutional networks for image classification," in *Proc. ICANN*, 2019, pp. 3–16. [4](#)
- [68] W. Ye and K. Ma, "Blurriness-guided unsharp masking," *IEEE Transactions on Image Processing*, vol. 27, pp. 4465–4477, 2018. [4](#)
- [69] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *Proc. ICLR*, 2016, pp. 1–13. [5](#)
- [70] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. ICML*, 2015, pp. 448–456. [5](#)
- [71] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. AISTATS*, 2011, pp. 315–323. [5](#)
- [72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, 2016, pp. 770–778. [5](#), [6](#)
- [73] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, "Residual attention network for image classification," in *Proc. CVPR*, 2017, pp. 6450–6458. [5](#)

- [74] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. ICCV*, 2015, pp. 1026–1034. [6](#)
- [75] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, "Large scale online learning of image similarity through ranking," in *Proc. PRIA*, 2009, pp. 11–14. [6, 7](#)
- [76] 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, p. 47, 2021. [6, 7](#)
- [77] Y. Shi, L. Liu, X. Yu, and H. Li, "Spatial-aware feature aggregation for image based cross-view geo-localization," in *Proc. NIPS*, 2019, pp. 10 090–10 100. [6, 7](#)
- [78] H. Zhao, J. Jia, and V. Koltun, "Exploring self-attention for image recognition," in *Proc. CVPR*, 2020, pp. 10 073–10 082. [9](#)
- [79] M. Andriluka, L. Pishchulin, P. V. Gehler, and B. Schiele, "2d human pose estimation: New benchmark and state of the art analysis," in *Proc. CVPR*, 2014, pp. 3686–3693. [11](#)



**Shaozi Li** received a B.S. degree from Hunan University, a M.S. degree from Xi'an Jiaotong University, and a Ph.D. degree from the National University of Defense Technology. He currently serves as the chair and professor of the Cognitive Science Department of Xiamen University, the vice director of the technical committee on Collaborative Computing of CCF, and the vice director of the Fujian Association of Artificial Intelligence. He is also a senior member of ACM and the China Computer Federation (CCF). His research interests cover Artificial Intelligence and Its Applications, Moving Objects Detection and Recognition, Machine Learning, Computer Vision, Multimedia Information Retrieval, etc. He has directed and completed more than twenty research projects, including several National 863 Programs, National Nature Science Foundation of China and Ph.D. Programs Foundation of Ministry of Education of China.



**Jinliang Lin** received a B.S. degree from the College of Engineering from Shantou University, China, in 2019. He is currently a master's student in the Department of Artificial Intelligence from Xiamen University, China. His research interests include cross-view geo-localization and fine-grained visual recognition.



**Zhedong Zheng** received a Ph.D. degree from the University of Technology Sydney, Australia, in 2021 and a B.S. degree from Fudan University, China, in 2016. He is currently a postdoctoral research fellow at NExT++, School of Computing, National University of Singapore. He was an intern at Nvidia Research (2018) and Baidu Research (2020). His research interests include robust learning for image retrieval, generative learning for data augmentation, and unsupervised domain adaptation.



**Yi Yang** received a Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He is currently a professor with the College of Computer Science and Technology, Zhejiang University. He was a professor with University of Technology Sydney, Australia and a Post-Doctoral Researcher with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. His current research interests include machine learning and its applications to multimedia content analysis and computer vision, such as multimedia analysis and video semantics understanding.



**Zhun Zhong** received his Ph.D. degree in the Department of Artificial Intelligence of Xiamen University, China, in 2019. He was also a joint Ph.D. student at University of Technology Sydney, Australia. He is now a postdoc at University of Trento, Italy. His research interests include person re-identification, novel class discovery, data augmentation and domain adaptation.



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



**Nicu Sebe** is Professor in the University of Trento, Italy, where he is leading the research in the areas of multimedia analysis and human behavior understanding. He was the General Co-Chair of the IEEE FG 2008 and ACM Multimedia 2013. He was a program chair of ACM Multimedia 2011 and 2007, ECCV 2016, ICCV 2017 and ICPR 2020. He is a general chair of ACM Multimedia 2022 and a program chair of ECCV 2024. He is a fellow of IAPR.