GEOCAPSNET: AERIAL TO GROUND VIEW IMAGE GEO-LOCALIZATION USING CAPSULE NETWORK

Bin Sun[1], Chen Chen[2], Yingying Zhu[1], Jianmin Jiang[1]

1. Shenzhen University, China, 518060
2. Department of Electrical and Computer Engineering, University of North Carolina at Charlotte
2161230410@email.szu.edu.cn, chen.chen@uncc.edu, {zhuyy, jianmin.jiang}@szu.edu.cn

ABSTRACT

The task of cross-view image geo-localization aims to determine the geo-location (GPS coordinates) of a query ground-view image by matching it with the GPS-tagged aerial (satellite) images in a reference dataset. Due to the dramatic changes of viewpoint, matching the cross-view images is challenging. In this paper, we propose the GeoCapsNet based on the capsule network for ground-to-aerial image geo-localization. The network first extracts features from both ground-view and aerial images via standard convolution layers and the capsule layers further encode the features to model the spatial feature hierarchies and enhance the representation power. Moreover, we introduce a simple and effective weighted soft-margin triplet loss with online batch hard sample mining, which can greatly improve the image retrieval accuracy. Experimental results show that our GeoCapsNet significantly outperforms the state-of-the-art approaches on two benchmark datasets. *The source code will be released soon.*

Index Terms— Image geo-localization, Cross-view image matching, Capsule network, Batch hard-mining

1. INTRODUCTION

Image geo-localization refers to the problem of determining where (i.e. GPS coordinates) an image is taken from based on the visual information only. This research has attracted widespread attention in recent years, due to its potential applications in autonomous driving, augmented reality, to name a few. Traditional geo-localization approaches requires accurate telemetry and sensor model e.g. digital ortho-quad (DOQ) [1], and Digital Elevation Map (DEM) to perform geo-registration with the reference data. However, these accurate models are difficult to obtain. Recently, geo-localization based on image matching has attracted growing interest since it is free from the constraint of requiring the meta data [2, 3]. A typical solution to this problem relies on matching "ground-to-ground" using a reference database of geo-tagged photographs [2, 4, 5, 6, 7]. These methods are relatively easy because both query and reference images are ground-level and they are in the same domain. One main drawback of such
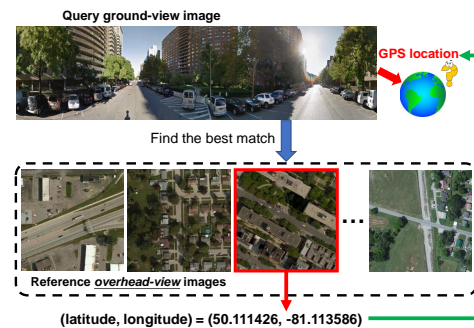
Fig. 1. Cross-view image geo-localization. Given a street-view image as a query, the goal of geo-localization is to determine its GPS location by matching it with a reference database of overhead satellite images with GPS coordinates. Due to viewpoint difference, the visual contents look very different in cross-view images.

approaches is that the reference dataset contains geo-tagged images which are concentrated in cities and tourist attractions. However, ground-level images of some geo-graphical locations may not have geo-location information. Therefore, ground-level image matching based methods cannot scale to global scale due to lack of reference data.

On the other hand, thanks to the advent of satellite and aerospace surveys, aerial photographs densely cover the entire planet. As a result, matching ground-level photos to aerial imagery (e.g. Google satellite imagery) has become an attractive alternative to the geo-localization problem [8, 9, 6, 10, 11, 12, 13, 14, 15, 16, 17]. As shown in Fig. 1, cross-view image matching is a very challenging task because of the drastic change in viewpoint between ground and aerial images. A key element of cross-view image matching is to learn the powerful feature presentation of the cross-view images, such that the distance of a matched pair of images is small whereas the distance of the unmatched pair is large in this feature space. However, learning feature embedding using pairs of images only considers the visual information similarity. The geometric discrepancy between two cross-view images is not properly addressed.

Motivation. Recently, capsule network has been proposed [18] to address some of the limitations of convolutional neural networks (CNN). Capsule network enables building parts-to-whole relationship between entities and allows capsules to learn viewpoint invariant representations. Inspired by these properties of the capsule network, we propose an aerial to ground view image geo-localization approach, namely GeoCapsNet, by leveraging the feature representation power of the capsule network. It tasks as input cross-view (ground view and overhead view) image pairs, matched or unmatched, and learns a feature embedding space such that features of the matching image pairs are close and unmatched image pairs are far apart. The contributions of the paper are:

- We propose an end-to-end network architecture GeoCapsNet for cross-view image-based geo-localization. Our work expands the use of capsule network to the task of image matching for the first time.

- We introduce a new weighted soft-margin triplet loss with online hard sample mining in each training batch. We show the batch hard-mining process is effective for improving the generalization ability of the network and therefore can boost the image retrieval performance.

- Extensive experiments on two datasets demonstrate that our GeoCapsNet significantly outperforms the state-of-the-art algorithms for cross-view image geo-localization.

2. RELATED WORK

In this section, we provide a review of the state-of-the-art solutions to the cross-view image geo-localization problem. Lin et al. [9] introduced the "discriminative translation" approach in which an aerial image classifier is trained based on ground-level scene matches for ground-to-overhead geo-localization. Bansal et al. [8] matched query street-level facades to airborne imagery under viewpoint and illumination variation by selecting the intrinsic facade motif scale and modeling facade structure through self-similarity. Shan et al. [6] proposed a fully automated ground-based multi-view stereo model for matching ground-level photos to aerial imagery, which is capable of handling drastic viewpoint variations by adopting a novel view-dependent feature matching approach. Workman et al. [12] used multi-scale overhead images for the same location in order to perform cross-view training by embedding the feature representations from both views in a joint semantic feature space. Vo et al. [13] introduced a distance-based logistic loss to improve the performance of Siamese network and Triplet network for cross-view geo-localization. They also showed that explicit orientation supervision can improve the localization accuracy. Zhai et al. [15] developed a new network architecture to predict semantic layout of ground-level images from the corresponding overhead images, which can also be used for several other tasks such as orientation estimation and geo-calibration. Hu et al. [17]

adopted the fully convolutional network to extract local image features, which are then encoded into global image descriptors using the NetVLAD [19]. Although a great deal of effort has been devoted to build discriminative feature representations for cross-view images, it still remains challenging for cross-view image matching due to the large differences in visual contents and scene structures.

3. PROPOSED GEOCAPSNET

3.1. Capsule Network

The capsule network [18] uses a group of neurons to represent an entity. The important information about the state of the features detected by all capsules in the capsule network is encapsulated in the form of a vector. Since the neurons in the traditional network layers are too simple to represent a concept, the capsule network uses vectors as feature representations in the capsule layers. The output vector of the capsule represents two parts: (1) its length represents the probability of occurrence of an instance (e.g. object, visual concept or part thereof), (2) its direction indicates graphical properties of the object (e.g. position, color, direction, shape, etc.)

In the cross-view image matching problem, aerial and ground images share some semantics, e.g. road, tree, building, etc. Moreover, the scene layout and geometric structure are also important cues for image matching. Inspired by the capsule networks' capability of modeling spatial relationships (i.e. orientation and position) of extracted features, we propose an end-to-end cross-view image matching network incorporating the capsule layers, dubbed as GeoCapsNet, to encode the relative spatial relationship between features to obtain a powerful image representation. In the following section, we present the details of GeoCapsNet.

3.2. GeoCapsNet Architecture

The overall architecture of the proposed GeoCapsNet is shown in Fig. 2. It follows the Siamese network [20] structure with two identical networks in parallel. The input to the two networks is the ground and satellite image, respectively. For higher-level capsules to obtain semantic representations, we begin with a residual network structure called ResNetX to extract the semantic features of images. Following the convention of ResNet [21], the details of the ResNetX structure are in Table 1. It consists of two convolutional layers and four residual blocks (i.e. Conv3_x - Conv6_x). ResNetX uses Batch Normalization at every layer. The max-pooling layer is not used to preserve the information about the input data.

The output of ResNetX is 2048 feature maps with spatial size 7×7, which are served as input to the capsule layer. We use two layers of capsules: PrimaryCaps and GeoCaps. The PrimaryCaps layer has 32 primary capsules whose job is to take basic features detected by the ResNetX and produce combinations of the features. The "primary capsules"
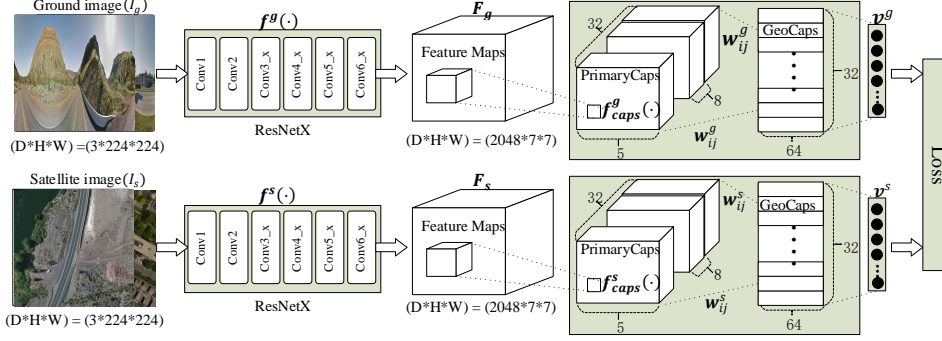
Fig. 2. The architecture of GeoCapsNet, which is a two-branch Siamese network takes as input a pair of cross-view images. Each network branch consists of two parts: ResNetX and Capsule layers (PrimaryCaps and GeoCaps layers).

Layer name	Output size	Layer	
Conv1	112×112	7×7, 64, stride 2	
Conv2	56×56	3×3, 64, stride 2	
Conv3_x	56×56	1×1, 64 3×3, 64 1×1, 256	$\times 3$
Conv4_x	28×28	1×1, 128 3×3, 128 1×1, 256	$\times 4$
Conv5_x	14×14	1×1, 256 3×3, 256 1×1, 1024	$\times 6$
Conv6_x	7×7	1×1, 512 3×3, 512 1×1, 2048	$\times 3$

Table 1. The structure of ResNetX.

are very similar to convolutional layer in their nature [18]. Each capsule applies eight $3\times3\times2048$ convolutional kernels (with stride 1) to the $7\times7\times2048$ input volume and therefore produces $5\times5\times8$ output tensor. Since there are 32 such capsules, the output volume has shape of $5\times5\times8\times32$. The Geo-Caps layer has 32 capsules, one for an entity in image. Each capsule takes as input a $5\times5\times8\times32$ tensor i.e. $5\times5\times32$ 8-dimensional vectors. As per the dynamic routing algorithm [18], each of these input vectors gets their own 8×64 weight matrix that maps 8-dimensional input space to the 64-dimensional capsule output space. The 32×64-dimensional vector representation of an image is obtained.

Concretely, let I_g and I_s denote the ground and satellite image, respectively. $f^g(\cdot)$ and $f^s(\cdot)$ indicate the corresponding ResNetX structure for I_g and I_s. In other words, the ResNetX models for I_g and I_s have separate model weights. The resulting features are $F_g = f^g(I_g)$ and $F_s = f^s(I_s)$ as shown in Fig. 2. F_g and F_s are passed to the PrimaryCaps layer of each branch, generating the output vectors of each capsule: $\boldsymbol{u}_i^s = f^s_{caps}(F_s)$, $\boldsymbol{u}_i^g = f^g_{caps}(F_g)$. Then $\boldsymbol{u}_i^s$ and $\boldsymbol{u}_i^g$ are fed into the corresponding GeoCaps layer through the dynamic routing algorithm, and the output of each capsule is

$\boldsymbol{v}_j^x$, where $x \in \{s,\ g\}$, and

$$\widehat{\boldsymbol{u}}_{j|i}^x = \boldsymbol{w}_{ij}^x \boldsymbol{u}_i^x, \quad \boldsymbol{s}_j^x = \sum_i c_{ij}^x \widehat{\boldsymbol{u}}_{j|i}^x, \quad \boldsymbol{v}_j^x = \frac{\|\boldsymbol{s}_j^x\|^2}{1 + \|\boldsymbol{s}_j^x\|^2} \frac{\boldsymbol{s}_j^x}{\|\boldsymbol{s}_j^x\|}$$

$\boldsymbol{w}_{ij}^x$ is a weight matrix that needs to be learned, c_{ij} are coupling coefficients that are determined by the iterative dynamic routing process. The representation of an image can be formulated as $\boldsymbol{v}^x = \{\boldsymbol{v}_1^x, \boldsymbol{v}_2^x, \boldsymbol{v}_3^x, \cdots \boldsymbol{v}_k^x\}$, where k represents the number of capsules in GeoCaps layer and $x \in \{s,\ g\}$ indicates the satellite or ground branch.

According to the weights learning strategy for the capsule layers in the two branches, we develop two variants of GeoCapsNet. Specifically, we denote two capsule branches with different model weights, i.e. $f^s_{caps}(\cdot) \neq f^g_{caps}(\cdot)$, as **GeoCapsNet-I**, and two capsule branches sharing the same model weights, i.e. $f^s_{caps}(\cdot) = f^g_{caps}(\cdot)$, $\boldsymbol{w}_{ij}^s = \boldsymbol{w}_{ij}^g$, $c_{ij}^s = c_{ij}^g$, as **GeoCapsNet-II**.

4. OBJECTIVE FUNCTION

In image retrieval tasks, Contrastive loss [22], Triplet loss [23, 24], and Quadruplet loss [25] are popular loss functions to train the deep neural networks. In image geo-localization, the goal of the loss function is to make the distance between the images of the same geo-location (positive pairs) as small as possible, and the distance between the images of different geo-locations (negative pairs) as large as possible. Take triplet loss as an example, a triplet is ensembled by randomly sampling three images from the training data, including an Anchor (a), a Positive sample (p) and a Negative sample (n). (a, p) forms a positive pair and (a, n) forms a negative pair. However, if the sample pairs are easy to distinguish, the network cannot learn a good feature representation, leading to poor generalization ability. To this end, we introduce a batch-wise hard sample mining method.

Batch construction and hard sample mining. We select M ground images in each training batch. For each ground image a in the batch, its matching satellite image p is used to construct the positive pair. Then a set of $M - 1$ satellite
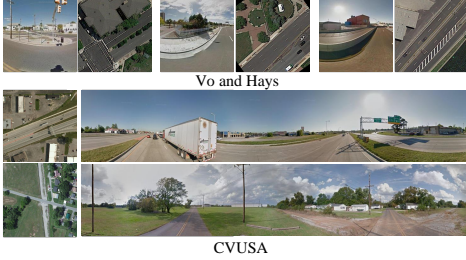
Fig. 3. Example cross-view images from two datasets.

Fig. 4. Top-K recall accuracy on CVUSA.

images in different geo-locations as a can be used to form negative pairs in the batch. This negative set of images is denoted as B. The triplet loss function is expressed as:

$$L_{tri} = \frac{1}{M} \sum_{a \in batch} (d_{a,p} - \min_{n \in B} d_{a,n} + \theta)_{+} \quad (1)$$

As shown in Eq. 1, $d_{a,p}$ is the distance between the capsule feature of a (i.e. $\mathbf{v}^g(a)$, see Fig. 2) and the capsule feature of p (i.e. $\mathbf{v}^s(p)$). $\min_{n \in B} d_{a,n}$ finds the negative sample which is closest to a, i.e. the hardest sample in the batch, to calculate the triplet loss. $\theta \geq 0$ is the margin, and $(\cdot)_{+} = max(0, \cdot)$.

To avoid manually setting the margin θ, we adopt the soft-margin triplet loss [17]: $L_{soft} = \ln\left(1 + e^d\right)$, where $d = d_{a,p} - d_{a,n}$. To improve the convergence rate, the weighted soft-margin ranking loss [17] scales d in L_{soft} by a coefficient α: $L_{weighted} = ln\left(1 + e^{\alpha d}\right)$. Therefore, our weighted soft-margin triplet loss with batch hard-mining (**Soft-TriHard Loss**) can be expressed as:

$$L_{sth} = \frac{1}{M} \sum_{a \in batch} \ln\left(1 + e^{\alpha\left(d_{a,p} - \min_{n \in B} d_{a,n}\right)}\right) \quad (2)$$

5. EXPERIMENTAL RESULTS

Datasets. We evaluate our GeoCapsNet on two cross-view datasets - CVUSA [15] and Vo and Hays [13]. The CVUSA consists of matching pairs of ground panoramas and satellite image. It contains 35532 image pairs for training and 8884 image pairs for testing. Vo and Hays dataset consists of street-view and overhead images from 11 different cities in the U.S. with more than 1 million pairs of images. Follow the same experimental setting in [17], we randomly select 9 cities, 8 of which are used to train our network, and the 9th – Denver city is for testing. Fig. 3 shows a few examples from the datasets.

Evaluation metric. The models are evaluated by the recall accuracy at top 1% for our networks, as is done in [17]. The recall at top 1% is percentage of cases in which the correct satellite match of the query ground view image is ranked within top 1 percentile.
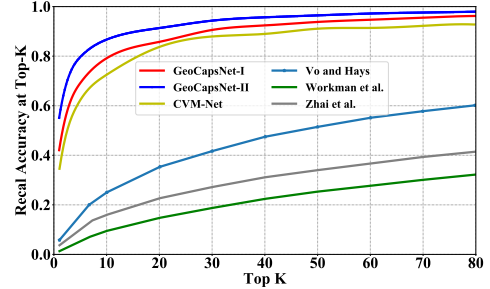
5.1. Implementation Details

The proposed model is trained for 50 epochs on the training set with a batch size of 32 ($M = 32$). We implement GeoCapsNet in Tensorflow. Adam optimizer [26] is used for training the model with an initial learning rate of 10^{-3}. RELU is the activation unit. Regularization is implemented in combination of L2-regularization and Batch Normalization. α is set to 15 in the Soft-TriHard loss. In GeoCaps layer, we set the number of capsule to 32, and the number of dynamic routing iterations to 4. L2 normalization is applied to the last layer feature of GeoCapsNet.

	Recall @top1%	
	Vo and Hays [13]	CVUSA [15]
Workman et.al [12]	15.40%	34.30%
Vo and Hays [13]	59.90%	63.70%
Zhai et al [15]	—	43.20%
CVM-Net [17]	67.90%	91.40%
GeoCapsNet-I	69.59%	96.52%
GeoCapsNet-II	**76.83%**	**98.07%**

Table 2. Performance comparison of our GeoCapsNet with the state-of-the-art cross-view geo-localization approaches.

5.2. Results and Ablation Study

Comparison to existing approaches. We compare our proposed GeoCapsNets to four state-of-the-art methods [12, 13, 15, 17] on two datasets. Table 2 shows the top 1% accuracies of our GeoCapsNet and other methods. Both GeoCapsNet-I and GeoCapsNet-II outperform all the other approaches by considerable margins on two datasets, leading to new state-of-the-art results. The results also reveal that GeoCapsNet-II achieves better performance than GeoCapsNet-I, suggesting that the weight sharing scheme of the two branch capsule layers forces the network to learn close and similar internal relationship representation between the cross-view images. It is also noted that all approaches have higher accuracy on CVUSA than Vo and Hays [13] dataset because the ground image in CVUSA is panoramic, which contains more information than the single-view image in Vo and Hays [13].
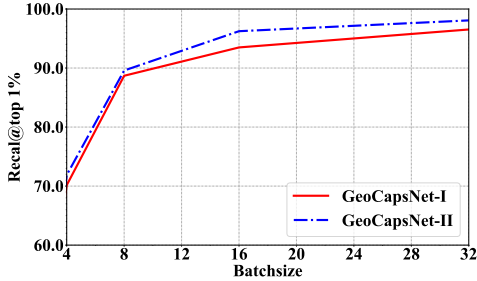
Fig. 6. Performance of our GeoCapsNets on CVUSA with different batch sizes.

Fig. 4 plots the Top-K (K=1–80) recall accuracy of our GeoCapsNets and other approaches on the CVUAS dataset. It is evident that our GeoCapsNets achieve much better performance than other approaches. The performance gain of GeoCapsNets over the state-of-the-art CVM-Net [17] is significant especially in the range of Top-1 to Top-20 recall, e.g. 20.53% improvement at Top-1 recall (GeoCapsNet-II 55.09% *vs* CVM-Net 34.56%).

To understand the effectiveness of our proposed GeoCapsNet, we conduct several ablation experiments to investigate the contribution of each important component.

Capsule layers. In this experiment, we replace the capsule layers (PrimaryCaps and Geo-Caps layer) in our Geo-CapsNets with a fully connected layer as the final feature representation to form the ResNetX-fc-I and ResNetX-fc-II networks. The proposed soft-TriHard loss is used. The results in Fig. 5 clearly demonstrate the advantage of using capsule layers for encoding more discriminative features.
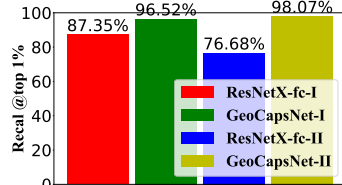
Fig. 5. Recall accuracy of Geo-CapsNet without capsule layers on CVUSA.

	Triplet	Soft-TriHard
GeoCapsNet-I	70.38%	96.52%
GeoCapsNet-II	77.46%	98.07%

Table 3. Recall@top1% of GeoCapsNet with different losses.

Batch hard-mining. To demonstrate the effectiveness of the batch hard sample mining procedure, we remove this process in our *Soft-TriHard* loss function. Therefore, the loss reduces to the weighted soft-margin triplet loss [17], represented by *Triplet* in Table 3. The results on the CVUSA datast suggest that batch hard-mining is very effective and is able to significantly boost the performance.

Batch size. As described in Section 4, we select M ground images in each training batch and construct the pos-

itive and negative pairs. We analyze the performance of our GeoCapsNets with different batch sizes. Specifically, we tune different values of batch size while keep other parameters the same. In Figure 6, as the batch size increases, the Top 1% recall accuracy of our GeoCapsNets becomes higher. This is because in our batch-hard mining method, the larger the batch size, the larger the search range of the samples, so that harder samples can be obtained. To balance the performance and memory requirement, we set $M = 32$.

Model comparison. Table 4 provides a model comparison between GeoCapsNet and CVM-Net in terms of the parameter size of the network, the storage size of the model, and the length of the feature encoding for image retrieval. The number of parameters in our GeoCapsNet is much smaller than that of the CVM-Net, leading to a more compact model. In addition, the code length (i.e. feature dimension) of Geo-CapsNet is only half the length of CVM-Net. In image retrieval, the shorter length of image feature coding means less computational complexity and faster retrieval speed. Finally, we show a few examples of geo-localizing query ground-view images using our GeoCapsNet in Fig. 7. **Please refer to the supplementary material for more examples and analysis.**

	# Parameters	Model size	Code length
CVM-Net [17]	160,311,424	1.8G	4096
GeoCapsNet-I	82,764,672	947.51M	2048
GeoCapsNet-II	**64,938,624**	**743.51M**	2048

**Table 4**. Comparison of GeoCapsNets and CVM-Net [17].

6. CONCLUSION

In this paper, we presented a cross-view image geo-localization method by matching query ground images with geo-tagged reference satellite images. We proposed the Geo-CapsNet architecture which captures high-level semantic features of images and their relationships due to the capsule layers. An effective batch hard sample mining is incorporated into the weighted soft-margin ranking loss, which greatly improves the retrieval accuracy of our network. Our approach significantly outperforms the state-of-the-art methods on two large-scale datasets.

7. REFERENCES

[1] Barbara Zitova and Jan Flusser, "Image registration methods: a survey," *Image and vision computing*, 2003.

[2] James Hays and Alexei A Efros, "Im2gps: estimating geographic information from a single image," in *CVPR*, 2008.

[3] Amir Roshan Zamir and Mubarak Shah, "Accurate image localization based on google maps street view," in *ECCV*, 2010.

[4] Akihiko Torii, Josef Sivic, and Tomas Pajdla, "Visual localization by linear combination of image descriptors," in *ICCV Workshops*, 2011.
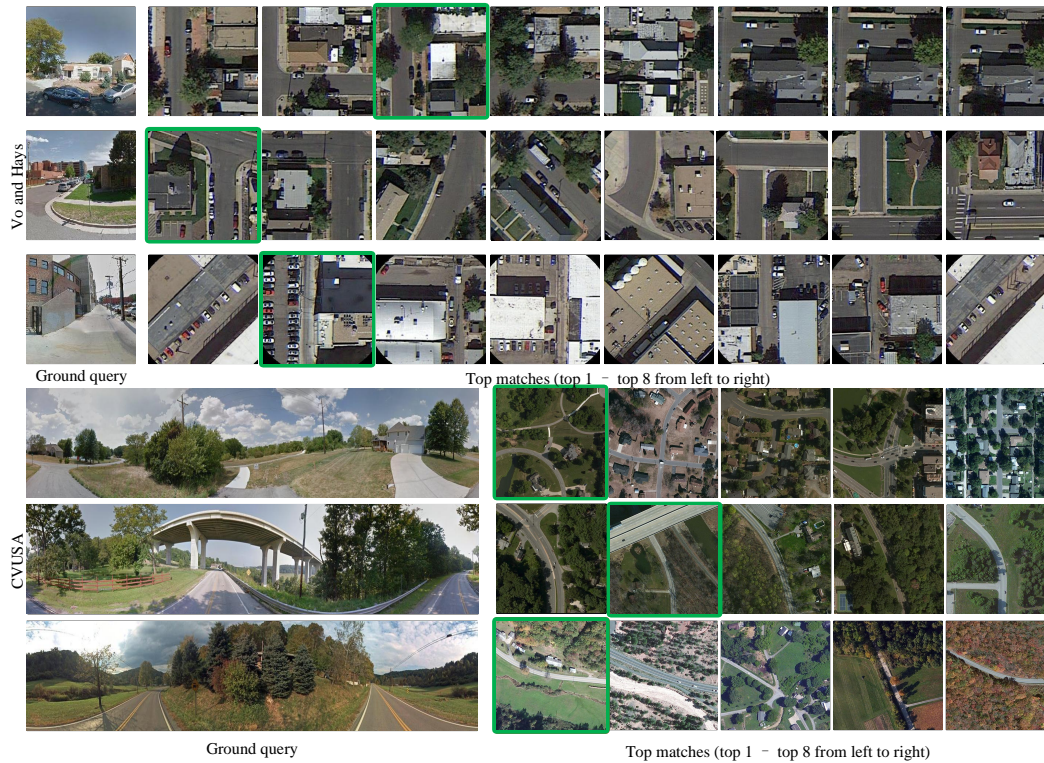
Fig. 7. Image retrieval examples of GeoCapsNet on two datasets. The image marked by the green box is the ground truth.

[5] Amir Roshan Zamir and Mubarak Shah, "Image geolocalization based on multiplenearest neighbor feature matching usinggeneralized graphs," *IEEE TPAMI*, 2014.

[6] Qi Shan, Changchang Wu, Brian Curless, Yasutaka Furukawa, Carlos Hernandez, and Steven M Seitz, "Accurate georegistration by ground-to-aerial image matching," in *3D Vision*, 2014.

[7] Grant Schindler, Matthew Brown, and Richard Szeliski, "City-scale location recognition," in *CVPR*, 2007.

[8] Mayank Bansal, Kostas Daniilidis, and Harpreet Sawhney, "Ultrawide baseline facade matching for geo-localization," in *Large-Scale Visual Geo-Localization*. 2016.

[9] Tsung-Yi Lin, Serge Belongie, and James Hays, "Cross-view image geolocalization," in *CVPR*, 2013.

[10] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays, "Learning deep representations for ground-to-aerial geolocalization," in *CVPR*, 2015.

[11] Scott Workman and Nathan Jacobs, "On the location dependence of convolutional neural network features," in *CVPR Workshops*, 2015.

[12] Scott Workman, Richard Souvenir, and Nathan Jacobs, "Wide-area image geolocalization with aerial reference imagery," in *ICCV*, 2015.

[13] Nam N Vo and James Hays, "Localizing and orienting street views using overhead imagery," in *ECCV*, 2016.

[14] Elena Stumm, Christopher Mei, Simon Lacroix, Juan Nieto, Marco Hutter, and Roland Siegwart, "Robust visual place recognition with graph kernels," in *CVPR*, 2016.

[15] Menghua Zhai, Zachary Bessinger, Scott Workman, and Nathan Jacobs, "Predicting ground-level scene layout from aerial imagery," in *CVPR*, 2017.

[16] Yicong Tian, Chen Chen, and Mubarak Shah, "Cross-view image matching for geo-localization in urban environments," in *CVPR*, 2017.

[17] Sixing Hu, Mengdan Feng, Rang M. H. Nguyen, and Gim Hee Lee, "Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization," in *CVPR*, 2018.

[18] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton, "Dynamic routing between capsules," in *NIPS*, 2017.

[19] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," in *CVPR*, 2016.

[20] Sumit Chopra, Raia Hadsell, and Yann LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *CVPR*, 2005.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *CVPR*, 2016.

[22] Rahul Rama Varior, Mrinal Haloi, and Gang Wang, "Gated siamese convolutional neural network architecture for human re-identification," in *ECCV*, 2016.

[23] Florian Schroff, Dmitry Kalenichenko, and James Philbin, "Facenet: A unified embedding for face recognition and clustering," in *CVPR*, 2015.

[24] De Cheng, Yihong Gong, Sanping Zhou, Jinjun Wang, and Nanning Zheng, "Person re-identification by multi-channel parts-based cnn with improved triplet loss function," in *CVPR*, 2016.

[25] Weihua Chen, Xiaotang Chen, Jianguo Zhang, and Kaiqi Huang, "Beyond triplet loss: a deep quadruplet network for person re-identification," in *CVPR*, 2017.

[26] D Kinga and J Ba Adam, "A method for stochastic optimization," in *ICLR*, 2015.