Proportional Search Space Reduction: A Novel Metric for Cross-View Image Geo-Location

Leon Debnath[1][0009-0001-0395-434X], Alexia Briassouli[2][0000-0002-0545-3215], and Mirela Popa[2]

[1] Department of Advanced Computing Sciences, Maastricht University, Paul-Henri Spaaklaan 1, 6229 EN, Maastricht, The Netherlands
l.debnath@student.maastrichtuniversity.nl

[2] Department of Advanced Computing Sciences, Maastricht University, Paul-Henri Spaaklaan 1, 6229 EN, Maastricht, The Netherlands
{alexia.briassouli,mirela.popa}@maastrichtuniversity.nl

Abstract. Identifying where a photo was taken can be achieved by matching the query ground view image to a satellite image of known location. This has been done in the past using Siamese Neural Networks by training a model that embeds images into feature vectors and comparing the distance between resultant vectors within the space to find a close match. Historically the number of correct recalls within top k% of matches was used as a metric when testing these models, named Recall at k (R@k). This paper highlights an issue with prior implementations of the canonical R@k metric related to boundary cases, leading to the miscounting of recalled images. As a result, models that provide state-of-the-art performance when measured using R@k may yield poor qualitative results in practice. Therefore, this paper proposes a novel metric, Proportional Search Space Reduction (PSSR), which measures how much the search space is reduced by a model under assessment, and has the potential to include boundary cases that R@k may miss. Three models were trained and evaluated to show that models with high Recall at 1% do not perform as well in real world applications as the metric may suggest, and proposes the use of PSSR for future research into the problem.

Keywords: Cross-View Image Geo-location · Siamese Neural Networks · Recall at K · Proportional Search Space Reduction.

1 Introduction

Image geo-location aims to identify the location of an image from its contents, without using embedded metadata (such as EXIF location). Many social media sites remove metadata from images before they are made public to protect users; this can make corroboration of location of a photograph intractable for journalists or governments. Image geo-location has been conducted manually as an investigation technique in many high profile cases, such as the downing of Malaysia Airlines MH17 over Ukraine in 2014, where images of Russian launchers were located by matching them to road signs in street view images, satellite photos, and social media posts.

Image geo-location aims to identify the location of an image from its contents, without using embedded metadata. Image geo-location is an image retrieval problem; the query is posed as a ground level image without a known location, and its matching aerial image (with a known coordinate position) provides the solution. The most common metric for performance found in the literature was introduced to this domain by Lin*et al.* [6]; Recall at k (R@k). This is defined as the accuracy of the model across the test set to recall, within the top-k results (or top-k% of results), the correct aerial image for each query image within the set. Recall is formally defined as [12]:

$$\text{recall} = \frac{TP}{TP + FN} \tag{1}$$

In the context of image geo-location R@k is a binary metric that is true when the correct image is recalled within the top-k results that are retrieved, a count of the top-k is made across the entire test set for a percentage average. The R@k metric is known to have the issue of being less descriptive of success for smaller test sets, as Ghanem *et al.* [2] noted: ”one of the shortcomings of the R@k metric is that it depends on the size of the validation dataset”. R@k% (**n.b.** the %) is considered more balanced as, when the validation set grows, the allowable error for an image to remain in the top k% grows with it proportionally.

Incredibly impressive R@1 and R@1% accuracy has been achieved in the past 82.53% and 99.67% respectively [4], however, this was using 360° panoramic images from Google Street View and unfortunately the same model only scored a meagre 4% R@1% on ordinary photographs with the field of view (FoV) limited to 70°.

1.1 Contributions

This paper introduces a novel metric called Proportional Search Space Reduction (PSSR) for cross-view image geo-location focused on the proportional reduction of the search space. A model was trained that learnt to take advantage of R@k to perform well beyond the state-of-the-art, despite poor qualitative results. The PSSR metric was applied to the proposed model to contrast the performance against the R@k metric, and to highlight the limitations of the R@k metric's implementations.

2 Related Work

Lin *et al.* [7] introduced the use of Siamese Neural Networks to the field of image geo-location, inspired by their use in DeepFace[13], and created a dataset of images from street view panoramas and 45° aerial images, coining the term Cross-View Image Geo-location (CVIG). Workman *et al.* [15] collected the canonical

CVUSA dataset with 1,036,804 Street View panoramas and 551,851 images from the Flickr photo sharing website and assessed several networks' performance with pre-trained weights from Places[16] and ImageNet[1]. They acknowledged that the dataset contained many images that did not provide enough context to draw useful feature vectors from.

Hu *et al*. [5] proposed the CVM-Net architecture, utilising NetVLAD to form global image descriptors that were invariant to large viewpoint changes. Shi *et al*. [10] investigated the use of polar transforming the aerial image in order to identify an orientation using a sliding window method. The polar transformed aerial image was cropped and shifted to best align the ground features to the aerial image. This method produced results of 98.54% recall within top 10 and 91.96% within the top 1 images for images with 360° FoV. More recently Hogan *et al*. [4] produced the Where In The World (WITW) dataset, partially to address the difference in parallax of aerial images commonly used in mapping applications, compared to the more expensive high resolution satellite imagery. Rao *et al*. [9] proposed a cross convolutional model based on a Resnet50[3] architecture showing over 90% R@1.

2.1 Problem Statement

R@k has been used in the domain of Recommender Systems, where only a small proportion of total results are shown to the user, with many possible positive class results within the data. However, when applied to the retrieval problem of CVIG. where only a single positive result exists, this metric experiences some shortcomings.

Firstly, the R@k metric provides very little insight into the performance of images that are not recalled in the top-k closest matches, given that the key issue noted in many of the cited papers is the difference in performance of models when provided panoramic (360° FoV) images, and the more limited (70 - 90°) FoV images that are commonly taken and distributed.

Secondly, implementations of the metric within the literature of CVIG commonly use the same method (shown at Algorithm 1) a flaw in which allows networks to learn ways of cheating the metric, as is explained in Section 6, achieving high performances that are not reflective of the actual performance of the model.

3 Methods

In Siamese Neural Networks two embedding networks are trained in parallel (see Figure 1), one for the aerial view and another for the ground level view images. In order to improve efficiency during the testing phase, all of the test images are embedded into a vector form using the trained embedding networks and a matrix of distances $D = (m \times n)$ is computed between each of the n ground images (queries) and m aerial images. The main diagonal of the matrix D shows the distance between image pairs of the same location, an ascending sort of this matrix's column ranks the similarity of each aerial image to the query.

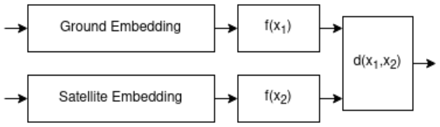


Fig. 1: The outline of Siamese Network architecture, where the embedding networks used were either VGG16 or ResNet50, and $f(x_i)$ being either a fully connected feature layer or a NetVLAD layer.

3.1 Proportional Search Space Reduction

Given the case where, in a dataset of 5,000 images, the correct image is recalled at index 102 of the ranked distances. This image would not contribute to the R@k accuracy metric (see Fig. 6). A hypothetical twofold improvement in performance sees that same image indexed at position 51, yet would still not register on the R@1% metric where the threshold for inclusion is $k \leq 50$.

PSSR addresses this by observing the number of values that fall above and below the positive image and is formally defined as:

$$PSSR = \frac{1}{n}\left(\sum_{i=1}^{n}\frac{n-k_i}{n}\right) \tag{2}$$

where k_i is the number of image embeddings that are closer to or equidistant from the correct image and the anchor, and n is the size of the dataset (see Fig. 7). PSSR provides two benefits over R@k. The granularity of averaged results means that improvements of a small reduction (e.g. 1%) averaged per image across all images will be reflected in the metric in a way that is unlikely for R@k (unless this improvement were to fall across a specifically measured boundary; such as the boundary between indexes 50 and 51 for R@1% on a dataset of 5000). Additionally PSSR results can be measured element-by-element within the dataset, demonstrating the distribution of results more clearly. The distribution of a well performing model is hypothesised to be a distribution with a long left tail, where the reduction of the search space is close to 100% for the majority of the images (although a model that performed this well was not trained in this study).

3.2 Dataset

The dataset used throughout this study was the CVUSA dataset, to which access was granted by Workman *et al*. [15]. The dataset comprised two parts: the first part containing approximately 550,000 images scraped from the website www.flickr.com with corresponding geolocated satellite images scraped from Bing maps and the second part consisting around 1.2 million panoramic images from Google Street-view with corresponding satellite photos. The Flickr images were all scraped from the website from different locations in the United States, a detailed breakdown of which can be found within the accompanying paper [15].

Many of the images were not suited to cross-view geo-location due to the lack of context available within them (such as macro photography or pictures of building interiors). To remove these images, a subset of 10,000 images were hand classified as viable or non-viable for geo-location with best effort made to achieve a 50% split of both classes. A VGG16 CNN with weights pre-trained on Places 365 was used as a feature extraction network, with the feature vectors classified by a Random Forrest classifier. This model produced an 87% accuracy which reduced the size of the dataset from 552,817 to 201,051 ground samples. A further 41,980 images were generated by cropping a subset of street-view panoramas to 90° field of view (4 per panorama) bringing the total to 243,031 image pairs.

An additional set of images were scraped from www.pic2map.com and matching satellite images were downloaded from Bing maps as a test set. This test set was hand validated, with any non-viable images discarded, leaving 6149 images in total. The test set was selected given the worldwide distribution of the images on the site, correcting for any chance of a performance boost due to having seen the location before, as many of the Flickr and street-view images were taken in the same towns in the US. It is acknowledged that the test set is small in comparison to the size of the training data (2.5% of the total data), but given the need to reduce the chance of the same location being used in both test and training set, more images could not be drawn from the test set. As Shi *et al.* [10] used a similar size test set in their paper, this was deemed an acceptable compromise.

3.3 Network Architectures

Three architectures were used, each using the same Siamese CNN design (see figure 1) without weights shared between the twins, and constructed from three layer blocks: an embedding layer of a CNN pre-trained on ImageNet[1], a feature extraction layer, and a differencing layer. The differencing layer was only used for network training and calculated the L2 distance between the two embeddings as per equation 3.

$$d(f(x_1), f(x_2)) = ||f(x_1) - f(x_2)||^2 \tag{3}$$

In the search for finding the best feature extraction technique, several options were considered, while keeping the complexity of the network quite low. The first network, based on the work of Lin *et al.* [7], used a VGG16[11] CNN as the embedding network, and a three layer feature network made up of three fully connected layers of 512, 256, and 256 neurons, each interspersed with a batch normalisation layer. All layers used the ReLU activation function. The second used ResNet50[3] as the embedding layer, with a similar feature network as the first. Finally CVM-Net-I[5] was implemented using a VGG16 embedding and the NetVLAD layer for feature extraction.

Method	Recall at top:					
	1	5	10	1%	5%	10%
VGG16	0.016	0.097	0.194	1.049	5.376	11.027
Resnet50	0.533	0.662	0.613	34.614	98.757	98.757
CVM-Net-I	0.048	0.113	0.194	1.017	5.004	10.010

Table 1: Recall at K results using the cropped test set

4 Experiments

Given that the satellite images were all centered on the point where the image was taken, in order to avoid the network generalising to the centered location, the satellite image data was randomly cropped down to no less than 70% of the original size in both height and width. To avoid a disparity between the random crops of the test set, the test set was randomly cropped once and saved to file. Tests were run both for the cropped and uncropped test sets. Each of the three architectures were trained initially on the entire training dataset using the max-margin loss function. Subsequently, once the improvement of the loss function ceased, the soft-margin loss function[14] was used. The models were optimised using the Adam optimiser with a learning rate $\alpha = 1 \times 10^{-5}$, a margin of $m = 0.5$ was used for the max-margin loss function when used; a weight of $\lambda = 10$ for soft-margin loss. A mini-batch size of 16 was used for each network while training for circa 10 epochs over the dataset taking around 25 hours to complete. [3]

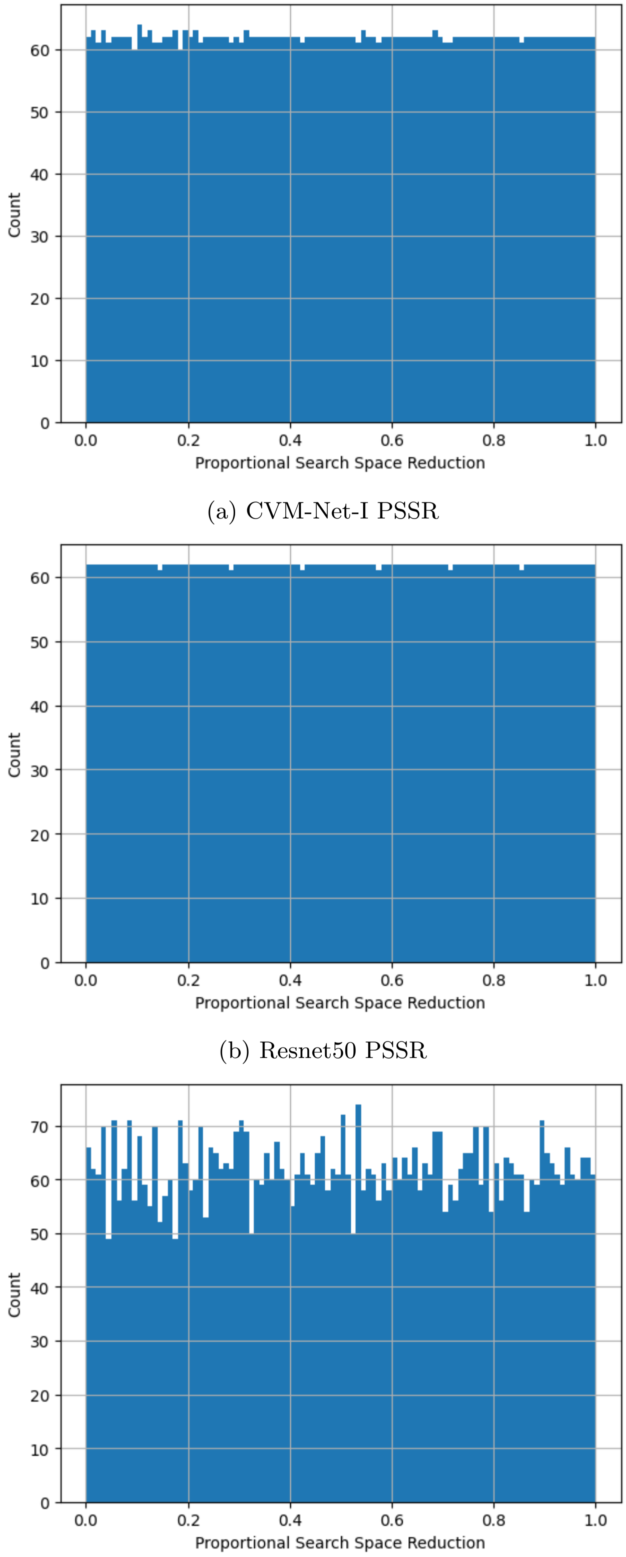
Both the PSSR and R@k metrics were recorded for each test using the cropped and uncropped test sets. It was hypothesised that CVM-Net-I, as the most recently created of the three architectures, would prove to be the most performant, however given the advances in the field since CVM-Net-I it was not expected to outperform the current state-of-the-art.

5 Results

The VGG16 and CVM-Net-I models performed reasonably similarly, with the CVM-Net-I proving to be slightly more invariant to the cropping of the test set satellite images (see Tables 1 and 2). Resnet50 performed exceptionally well, out performing the state of the art by a significant margin (circa 34% R@1%). The previous study conducted by Hogan *et al.* (WITW)[4], where a test set of non-panoramic images were used to test a model, stated that: ”Performance by one measure drops from 99% to circa 3% when switching from aligned panoramas to an equal number of ordinary photos. No single factor is responsible for that – it’s the collective result of many small, quantifiable effects”.

It can be observed in Figure 4 that, for selected examples of the network retrieving images, the correct images are being retrieved. However, this qualitative

[3] Code available at: https://github.com/S010MON/image-locate



(a) CVM-Net-I PSSR

(b) Resnet50 PSSR

(c) VGG16 PSSR

Fig. 2: PSSR distribution histograms for Resnet50, VGG16, and CVM-Net-I distribution histogram showing the number of images within each range of proportional reduction.

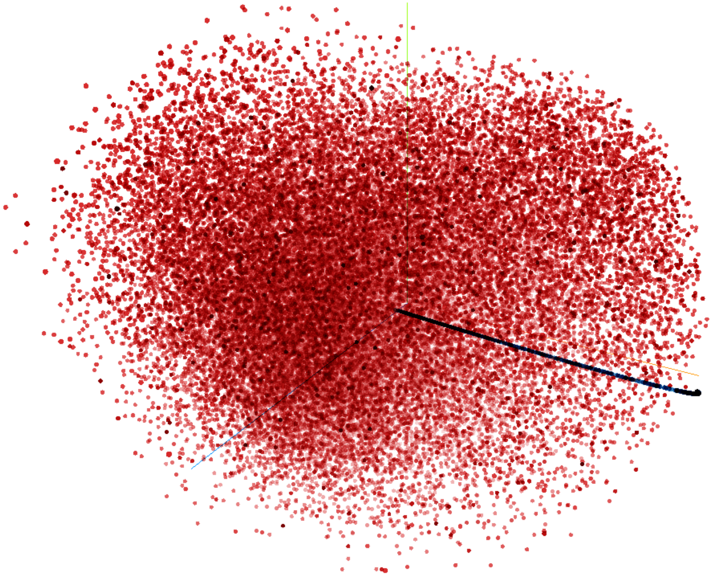


Fig. 3: A plot of 10,000 image embeddings for both Resnet (in blue) and VGG16 (red)

analysis also highlights that the same images occur in each of the 13 images in the R@5 category. When the PSSR metric for each model is plotted as a histogram (see Figs. 2a - 2b) the results are far less impressive. A well performing model would be expected to have a high count at the 1.0 mark (reducing the space almost completely) with a long tail on the left as the harder to classify images cause more of the search space to be included. Very slight perturbations in the CVM-Net-I and VGG16 left sides show this very slightly (although not to any level of statistical significance), however Resnet50 shows an almost perfect uniform distribution.

The high performance of the Resnet50 network becomes apparent when its embeddings are directly contrasted to VGG16 (see Fig. 3). As both use the same feature extraction layer, both embed to a 256-dimensional space and can be plotted together after principal component analysis is applied. The embeddings for VGG16 fill the space evenly, while Resnet50 takes on a dense linear structure that embeds the 10,000 images very closely together.

6 Discussion

The key to understanding the disparity between the very high performance of the Resnet50 network on the R@1% metric (see Table 1), while performing no better than random with PSSR is due to the implementation of R@K. Examining the implementations of some of the latest CVIG papers' validation steps provides

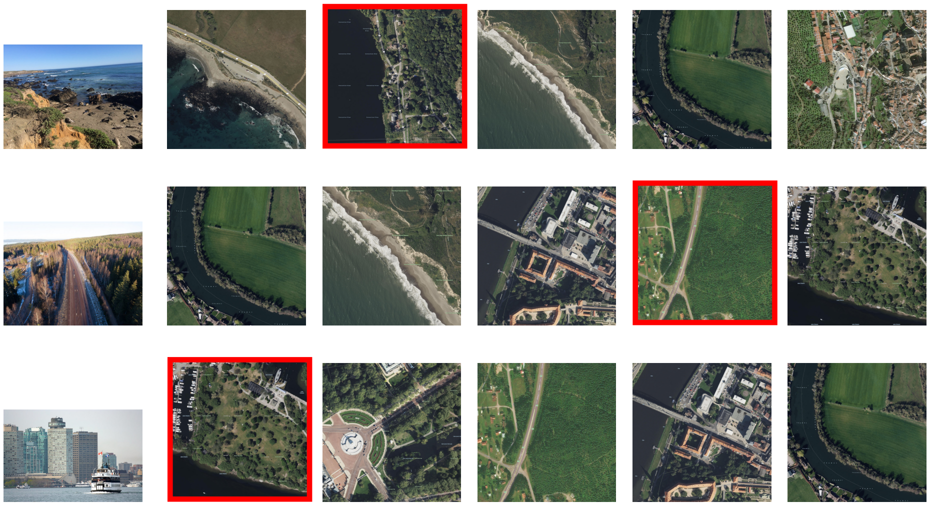


Fig. 4: Three examples of the Resnet50 network performing at its ”best”, The query image on the left, with the top 5 images shown in order of similarity, with the correct image shown in red

some insight; Hu *et al.* [5] published their Tensorflow 1 code from CVM-Net-I which was also used by Lui *et al.* [8] and Shi *et al.* [10] without major modification:

```
for i in range(dist_array.shape[0]):
  gt_dist = dist_array[i, i]
  prediction = np.sum(dist_array[:, i] < gt_dist)
  if prediction < top1_percent:
    accuracy += 1.0
  data_amount += 1.0
accuracy /= data_amount
```

While Hogan *et al.* used a PyTorch implementation with a vectorised solution:

```
for idx in tqdm.tqdm(range(count)):
  ...
  ranks[idx] = torch.sum(torch.le(
                        distances,
                        distance)).item()
...
top_percent = np.sum(ranks * 100 <= count)
                            / count * 100
```

And this paper implemented a Tensorflow 2 version that similarly took advantage of vector operations:

```
correct_dists = distances.diagonal()
sorted_dists = np.sort(distances, axis=1)
one_percent_idx = int(float(count) * 0.01)
top_one_percent = np.sum(correct_dists <=
         sorted_dists[:, one_percent_idx])
         / count * 100
```

It should be noted that all of these implementations of R@1% all follow the same algorithm:

```
Algorithm 1: Recall at K
  Input  : Matrix of distances $D$
  Output: Count of distances satisfying the condition
  Step 1: Identify the distance of each image pair ($\hat{d}$);
  Step 2: Calculate the index for the 1% element ($d_{1\%}$) and retrieve its value;
  Step 3: Count for all $\hat{d} < d_{1\%}$ or $\hat{d} \leq d_{1\%}$ (depending on implementation);
```

In the case where $d_{1\%}$ is equal to many values, retrieving the top 1% of closest elements to a query within the dataset will not necessarily select the element corresponding to $\hat{d}$. I.e. if 1% of the dataset is 500 images, however 1,000 images map to the same location in the vector space, then all images will be included regardless of proximity. In the case of the Resnet50 network, where a very dense mapping into the vector space occurs, there inevitably are far more of the dataset than the top 1% of elements. When not using a vectorised implementation, one could argue for the count of only the top-k by proximity, ceasing to count once k has been reached. This would raise an alternative issue, where the sorting of distances affects the order in which the images are validated, and different sorting algorithms would affect the results differently. PSSR avoids this problem by measuring how well each query performs in relation to the entire test set, providing a metric that can be averaged, or plotted per image. It also does not suffer from boundary conditions.

7 Conclusion

This paper introduced a new metric for evaluating the performance of Cross-View Image Geo-Location models and introduces a model that achieves state-of-the-art results. This performance is marred by the fact that the canonical Recall at k metric was proved to be a poor metric to show performance in real world applications in the way it has been implemented within the literature and recommends the use of PSSR to assess the performance of future CVIG models.

References

1. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (2009)
2. Ghanem, A., Abdelhay, A., Salah, N., Eldeen, A., Elhenawy, M., Masoud, M., Hassan, A., Hassan, A.: Leveraging cross-view geo-localization with ensemble learning and temporal awareness (3 2023). https://doi.org/10.13140/RG.2.2.35293.08169
3. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015)
4. Hogan, D., Tindall, L., Ashley, R., Gogia, M., Etten, A.V.: Where in the world: A new data-set for cross-view image geolocalization (2023)
5. Hu, S., Feng, M., Nguyen, R., Lee, G.: Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization (06 2018). https://doi.org/10.1109/CVPR.2018.00758
6. Lin, T.Y., Belongie, S., Hays, J.: Cross-view image geolocalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2013)
7. Lin, T.Y., Cui, Y., Belongie, S., Hays, J.: Learning deep representations for ground-to-aerial geolocalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2015)
8. Liu, L., Li, H.: Lending orientation to neural networks for cross-view geo-localization. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
9. Rao, Z., Lu, J., Li, C., Guo, H.: A cross-view image matching method with feature enhancement. Remote Sensing **15**(8) (2023). https://doi.org/10.3390/rs15082083, https://www.mdpi.com/2072-4292/15/8/2083
10. Shi, Y., Yu, X., Campbell, D., Li, H.: Where am i looking at? joint location and orientation estimation by cross-view matching. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2020)
11. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). https://doi.org/10.48550/ARXIV.1409.1556, https://arxiv.org/abs/1409.1556
12. Somogyi, Z.: Performance Evaluation of Machine Learning Models, pp. 87–112. Springer International Publishing, Cham (2021). https://doi.org/10.1007/978-3-030-60032-$7_3, https : //doi.org/10.1007/978 - 3 - 030 - 60032 - 7_3$
13. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: Deepface: Closing the gap to human-level performance in face verification. 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 1701–1708 (2014)
14. Vo, N., Hays, J.: Localizing and orienting street views using overhead imagery (2017)
15. Workman, S., Souvenir, R., Jacobs, N.: Wide-area image geolocalization with aerial reference imagery. CoRR **abs/1510.03743** (2015), http://arxiv.org/abs/1510.03743
16. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: A 10 million image database for scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(6), 1452–1464 (2018). https://doi.org/10.1109/TPAMI.2017.2723009

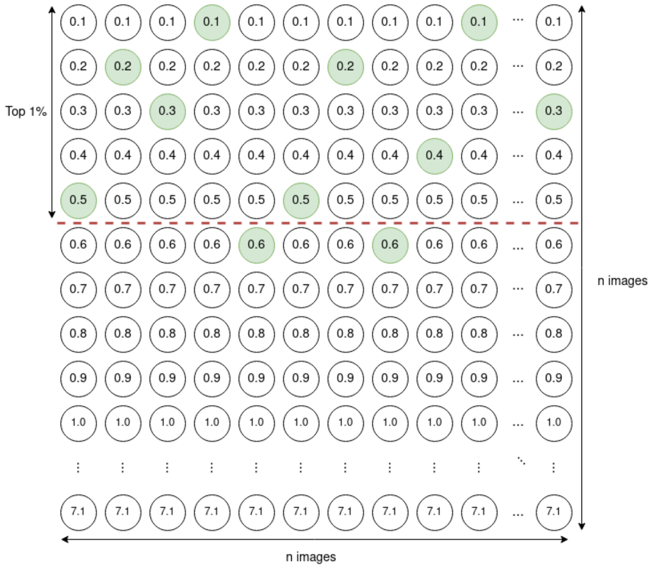


Fig. 5: Recall at 1% on a well-defined dataset, illustrative distances shown within each circle (green highlights the satellite image that matches the query image). Green images above the 1% mark are counted as correctly recalled.

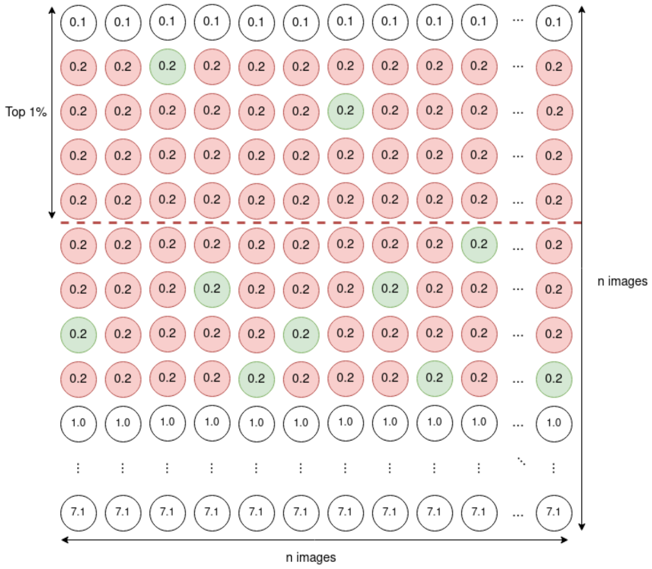


Fig. 6: Recall at 1% where images are mapped to vectors with almost no distance apart. Green images among the Red images are mapped to the same space, thus (due to the $\leq$ 1% implementation issue) being counted regardless of whether within the 1% recall space.

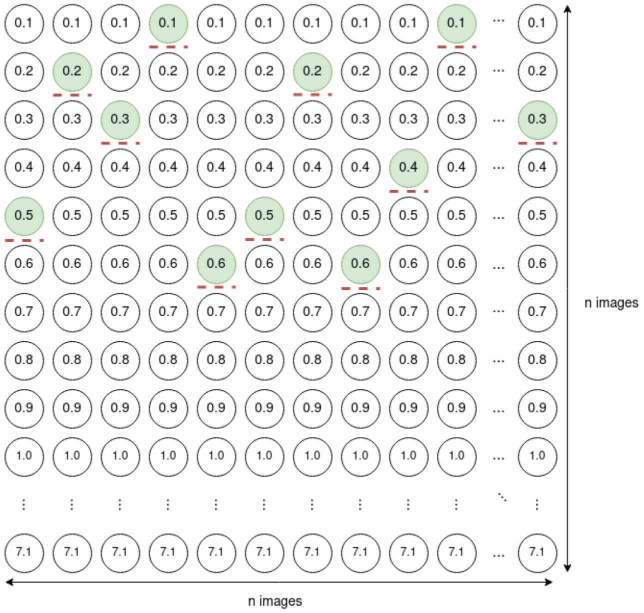


Fig. 7: Proportional Search Space Reduction sets a boundary per query image and measures how much of the search space is retained above the boundary

Method	Recall at top:					
	1	5	10	1%	5%	10%
VGG16	0.032	0.097	0.177	1.017	5.021	10.123
Resnet50	0.371	0.484	0.613	54.552	98.741	98.741
CVM-Net-I	0.0323	0.113	0.194	1.017	5.037	10.058

Table 2: Recall at k results using the un-cropped test set