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

Leon Debnath
Department of Advanced Computing Sciences
Maastricht University
l.debnath@student.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 is done by transforming the images into feature representations and comparing the distance between vectors within the space to find a close match. Historically the number of correct recalls within top 1% of matches was used as a metric. This paper proposes a novel metric, Proportional Search Space Reduction (PSSR), which measures the reduction in the search space of images without the loss of the correct image. 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.

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 is an image retrieval problem; the query is posed as a ground level image with location unknown, and its matching aerial image (with coordinate position known) provides the solution. The most common metric for performance found in the literature was introduced to this domain by Lin *et al*. [9] as 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 a each query image within the set. The R@K metric is known to have the issue of being less descriptive of success for smaller test sets, as Ghanem *et al*. [4] noted: "one of the shortcomings of the R@K metric is that it depends on the size of the validation dataset". R@$K\%$ is considered more balanced as, when the validation set grows, the allowable error for an image to remain in the top 1% grows with it proportionally.

Incredibly impressive R@1 and R@1% accuracy has been achieved in the past 82.53% and 99.67% respectively [7], however this was using 360° panoramic images commonly found on 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°. This could indicate a misalignment between the common metric used within the domain of research and the human-centric issue that image geo-location is attempting to solve, as the performance on panoramic images struggles to transfer to limited FoVs.

If the assumption is made that image geo-location will, in the close term future at least, not surpass the abilities of trained humans, then there will be the requirement for human intervention within systems utilising these models. Redefining the metrics for success as the ability of the model to reduce cognitive load from human operators, and provide results as fast as possible will encourage the development of models that are most fit for purpose for use by humans.

1.1. Contributions

This paper introduces a novel metric called Proportional Search Space Reduction for cross view image geo-location focused on the proportional reduction of the search space. The metric is applied to a state of the art model to contrast the performance against the canonical R@K metric, and shows the limitations of the R@K metric's implementations. Additionally this paper proposes a method of hierarchical batching to create "super batches" to improve the training speed of networks while online triplet mining.

1.2. Related Work

In 2008 Hays and Efros [5] proposed an algorithm to predict image locations through similarity of image features (forests, bodies of water, etc) to those held on database in order to geo-localise regions of likelihood. Subsequently Arandjelovìc *et al*. [1] investigated the use of Convolutional Neural Networks (CNNs) as a method of comparison invariant of clutter and viewpoint with their NetVLAD paper. Although this proved a highly successful technique for locations for which there exist many photographs to match (such as common tourist destinations like the Eiffel Tower), locations that are not frequently captured were not easily identifiable. Weyand *et al*. in PlaNet [18] used CNNs and LSTMs to estimate the location of images within a 200km location without querying the entire database, this was done by treating the problem as a classification problem through dividing the globe into cells and predicting the likelihood of an image belonging to a cell, localising images to larger regions.

A change in perspective. Lin *et al*. [9] introduced the use of cross-view aerial images which, taking advantage of the almost total imagery coverage of the earth's land surface area, solved the issue of the lack of sufficient data to match the query image. Lin *et al*. [10] subsequently introduced the use of Siamese Neural Networks, inspired by their use in DeepFace [16], and created a dataset of images from street view panoramas and 45° aerial images. Workman *et al*. [19] collected the canonical CVUSA dataset with 1,036,804 Street View panoramas and 551,851 images from the Flickr photo sharing website. Assessing several networks' performance with pretrained weights from Places [20] and ImageNet [3]. They acknowleged that the dataset contained many images that did not provide enough context to draw useful feature vectors from.

Loss functions. Vo and Hays [17] experimented with different network architectures, loss functions, and data augmentation methods and found that a network with triplet loss performed better than the siamese network with contrastive loss by a large margin. Triplet loss forces the network to reduce the distance between the positive example and the anchor image, $d(a,p)$ i.e d_{pos}, and increase the distance between the negative and the anchor, $d(a,n)$ i.e. d_{neg}. The loss function is defined as:

$$\mathcal{L}_{max} = \frac{1}{N}\sum_{i}^{N} max\Big(d_{pos} - d_{neg} + \alpha, 0\Big) \qquad (1)$$

where:

$$d_{pos} = d(f(x_1), f(x_2)) = ||f(x_1) - f(x_2)||^2 \qquad (2)$$

and α is some margin greater than zero, typically chosen between $0 \leq \alpha \leq 1$. Vo and Hays also noted, as did Schroff *et al*. in their FaceNet [13] paper, that triplet sampling can significantly improve learning performance by selecting triplets that are hard to distinguish from each other. Randomly selected triplets are routinely trivially easy due to the fact that the positive image is most likely closer to the anchor image than the size of the margin ($d(a,p) + \alpha < d(a,n)$) and thus returns a loss of 0. Semi-hard triplets (see figure 1) occur when the distance of the negative to the anchor is not less than the positive to the anchor, however the difference is still less than the margin, so the loss is not driven to zero i.e. $d(a,n) < d(a,p) < d(a,p) + \alpha$.

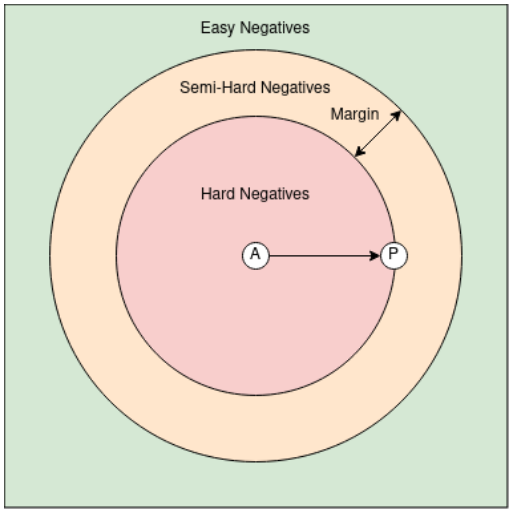

Figure 1. Triplet Mining. Semi-hard negatives are further from the anchor (A) than the positive (P) yet within the margin and provide the fastest learning for the network

Vo and Hays [17] proposed the use of a novel soft-margin triplet loss to overcome the issue of selecting a margin used in max-margin triplet loss. This improvement proved to be effective and was built on by Hu *et al*. [8] with the weighted soft-margin ranking loss used in CVM-Net-I and CVM-Net-II.

$$\mathcal{L}_{soft} = ln(1 + e^{\lambda d}) \qquad (3)$$

where: $d = d_{pos} - d_{neg}$ and λ is a weight value that, when increased, improves the speed of convergence. A weight value of $\lambda = 10$ was experimentally found to perform well by Hu *et al*.

Triplet Mining. Hu *et al*. [8] proposed their CVM-Net architectures, incorporating the NetVLAD module to pool the local feature extractions from the siamese CNNs. This achieved significant improvements over the state of the art. Their methodology included initial training on randomly assigned triplets before mining hard triplets for further training. Cai *et al*. [2] proposed a novel loss method that adaptively focused on semi-hard triplets to improve network training and learn more discriminative features. Zhu

et al. [21] furthered this approach with online global mining using updates during the training step to avoid the computational cost of offline global mining.

Image alignment. Hu *et al*. [8] proposed the CVM-Net architecture, utilising NetVLAD to form global image descriptors that were invariant to large viewpoint changes. Shi *et al*. [14] 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. Zhu *et al*. [21] further highlighted the importance of image alignment in their revisitiaion paper. Zhu *et al*. [22] approached the subject of top-down alignment of aerial images with the VIGOR benchmark set introducing aerial imagery with coordinates identifying the location of the ground view image, rather than images being centred. More recently Hogan *et al*. [7] 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. Several of these models were combined in an ensemble model by Ghanem *et al*. [4] noting extremely promising results, yet limited deployablility for running several such models concurrently. In the same year Rao *et al*. [12] proposed a cross convolutional model based on a Resnet50 [6] architecture showing over 90% R@1.

1.3. Problem Statement

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. The R@K metric provides very little insight into the performance of images that are not in the K closest matches. The additional context is important for applications of CVIG, where end users are looking to identify the matching satellite image. In the manual form of CVIG, operators will look to reduce the search space by identifying markers within the image to narrow the search by country (e.g. using road markings, language on signs, or known information about the image), region (floral makeup or man-made structures), and location (comparison matching). At the current performance of state-of-the-art models, automating this process will not remove humans from the loop. Augmenting their ability by identifying the highest likelihood locations and reducing the search space of possibilities is well within the current capability. Crucially, more needs to be known about how the models perform on reducing the search space to practically apply the models without high failure rates.

1.4. Research Questions

The following research questions seek answers to key issues in creating and assessing CVIG models in closer alignment to real world application needs.

1. By how much do different cross-view image geo-location model architectures reduce the search space with 100% recall?

2. How can datasets be qualitatively improved or enlarged?

3. How are triplets best mined for improved training speed?

2. Methods

2.1. Dataset

The dataset used throughout this study was the CVUSA dataset, to which access was granted by Workman *et al*. [19]. The dataset consists of 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, the second part consisting around 1.2 million panoramic images from Google Street-view with corresponding satellite photos. This paper focuses mainly on the Flickr dataset, with some minor augmentation using a subset of streetview images. 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 [19]. Many of the images were not suited to cross-view geo-location due to the lack of context. Macro flora and fauna; sport and scuba-diving; and close-up vehicle and building photography are very common within the dataset (see fig. 2).

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).

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

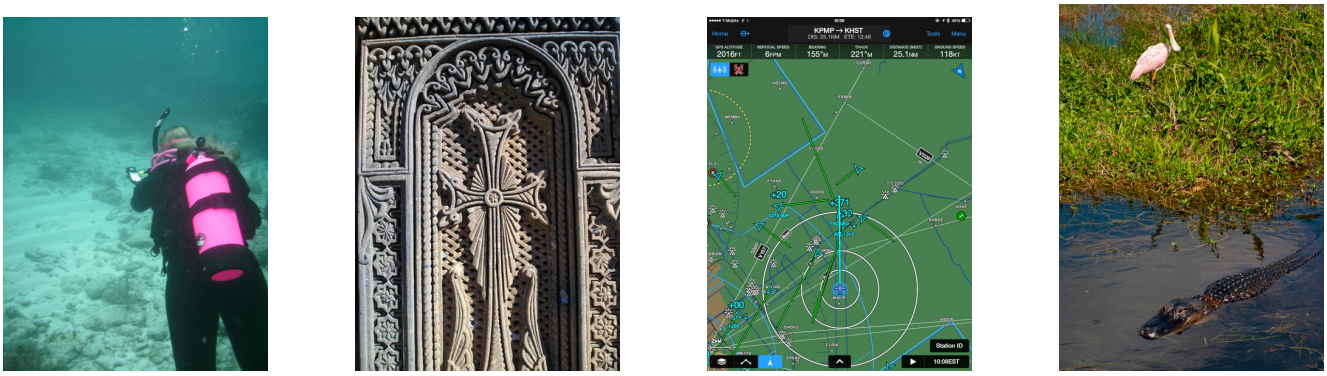

Figure 2. Images from the CVUSA dataset that are unsuitable for cross-view matching due to lack of context within the image

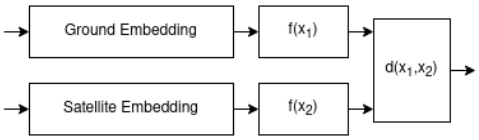

Figure 3. The outline siamese network architecture, where the embedding networks used were either VGG16 or ResNet50, and $f(x_i)$ being either a fully connected feature layer or NetVLAD layer.

the location before as many of the Flickr and street-view images were taken in the same locations in the US.

2.2. Network Architectures

Three architectures were used, each using the same siamese CNN design (see figure 3) without weights shared between the twins, and constructed from three layer blocks; an embedding layer of a CNN pre-trained on ImageNet [3], a feature extraction layer, and a differencing layer. The differencing layer was only used for network training and calculated the L2 distance as per equation 2.

The first network, based on the work of Lin *et al*. [10], used a VGG16 [15] 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 [6] as the embedding layer, with a similar feature network as the first. Finally CVM-Net [8] was implemented using a VGG16 embedding and the NetVLAD layer for feature extraction.

2.3. Proportional Search Space Reduction

Proportional Search Space Reduction (PSSR) aims to improve on the lack of context provided by the R@K metric at low levels of recall. While R@K provides a quantitative metric of the success rate of an algorithm, in the case where, in a dataset of 5,000 images, an image that lies at index 102 of the ranked distances will not contribute to the accuracy metric. However, a twofold improvement in recall sees that same image indexed at position 51, yet still does not register on the R@1% metric. PSSR takes the correct image as the boundary and observes the number of values that fall above and below the boundary. Formally defined as:

$$PSSR = \frac{1}{n}\left(\sum_{i=1}^{n}\frac{n-k_i}{n}\right) \tag{4}$$

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. 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.

2.4. Hierarchical Super Batching

The use of online semi-hard triplet mining increased the speed of learning, however the batches still inherently relied on stocasticity within the shuffling of the batches to achieve semi-hard triplets that were non-trivial for the network to learn. As a batch size of 16 was used, the probability of having similar images could be improved by creating "super batches" of similar satellite images and selecting mini-batches from them for training.

An initial approach of treating this as a classification task was considered, however as the images were encoded into an n-dimensional euclidean space, k-means clustering presented an unsupervised option to categorise the similarity of the images. A hypothesis was formulated that using a value

Figure 4. k-means clustered dataset with $k = 3$, embedded by CVM net with 1 epoch of training

of $k = 3$ would output three clusters aligned to the features of urban, rural, and littoral satellite images.

The size of the dataset did not allow for k-means to be run on it in its entirety, however a subset of 50,000 embeddings (circa 20% of the data) was used to train a model that predicted the classes of the remaining 80%. Reducing the dimensionality of the embeddings with PCA and applying the predictions as coloured overlay shows the mapping of three distinct image super-batches. Figure 4 shows the three classes in red, purple, and blue while figure 5 demonstrates a random sub-sample of each class confirming that the hypothesised split roughly holds across rural, urban, and littoral terrain images.

2.5. Reducing Computational Overhead

Lin *et al*. in their 2015 paper, explored the use of separating the networks after training and pre-computing the satellite images offline to speed up the comparison matching process. They used Locality Sensitive Hashing to speed up the comparison of k-nearest neighbours to query matches. An alternative method of reducing the time taken to retrieve from the search space is the use of k-means to bucket elements within the dataset. When run recursively, with an arbitrarily selected k, the search space can be divided into k^d buckets, where d is the number of times the dataset is recursively clustered. This method generates a search tree where the mean value of each class' centroid is used as the marker for each node. When retrieval is conducted, a single forward pass of the query embedding is required. Subsequently the query embedding distance is calculated to the mean of the k uppermost nodes. The closest is selected and the next k nodes's mean values are again retrieved until the final bucket is reached. This algorithm allows for $O(log_k(n))$ time complexity.

Component	Detail
OS	Pop!_OS 22.04 LTS x86_64
CPU	AMD Ryzen 5 5600X (12) @ 3.700GHz
GPU	NVIDIA GeForce RTX 3060
RAM	64221MiB

Table 1. Hardware specifications

3. 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 to between 100% and 70% of the original image size. To avoid a disparity between the random crops of testing between test sets that were cropped in a more or less difficult manner, the test set was randomly cropped once and saved to file. This set was used for every test. Each of the architectures were trained on the CVUSA dataset. Each model was trained initially on the entire test dataset using the max-margin loss function. Subsequently, once the loss function ceased improving, the super-batches were created by classifying the embeddings that the partially trained network had learned and were used to increase the likelihood of hard and semi-hard triplets within each batch. Additionally the loss function was changed to the weighted soft-margin triplet loss for the second bout of training. Both the PSSR and R@K metrics were recorded for each test. The test set was made up of 6149 images that were selected worldwide and did not share any locations with the training set, and were validated manually to ensure that the images were of high quality and provided enough context to be classified. It was hypothesised that CVM-Net, as the most recently created of the three architectures, would prove to be the most performant.

3.1. Implementation Details

The models were all implemented in the Tensorflow 2 framework and containerised within docker to both take advantage of hardware acceleration and to improve reproduceability. All training and testing was run on a machine with specifications shown in table 1.

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.

4. Results

The VGG16 and CVM-Net-I models performed reasonably similarly, with the CVM-Net-I proving to be slightly

(a) Class: 0 (b) Class: 0 (c) Class: 0 (d) Class: 0 (e) Class: 0

(f) Class: 1 (g) Class: 1 (h) Class: 1 (i) Class: 1 (j) Class: 1

(k) Class: 2 (l) Class: 2 (m) Class: 2 (n) Class: 2 (o) Class: 2

Figure 5. A random sample from each class describes how the classes 0, 1, and 2 roughly align with the littoral, urban, and rural categories respectively

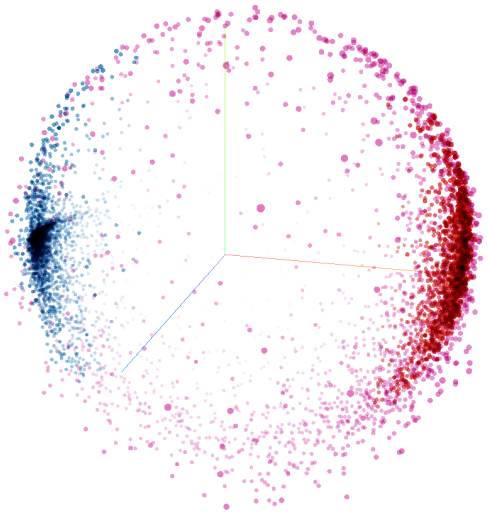

Figure 6. k-means clustered dataset with $k = 3$, embedded by CVM net with 7 epochs of training

more invariant to the cropping of the test set satellite images. 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), where non-panoramic images were tested against, stated "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."

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

It can be observed in figure 11 that, for selected examples of the network retrieving the correct image, the correct images are being retrieved. However this quantitative analysis also highlights that the same images occur in each of

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 3. Recall at K results using the cropped test set

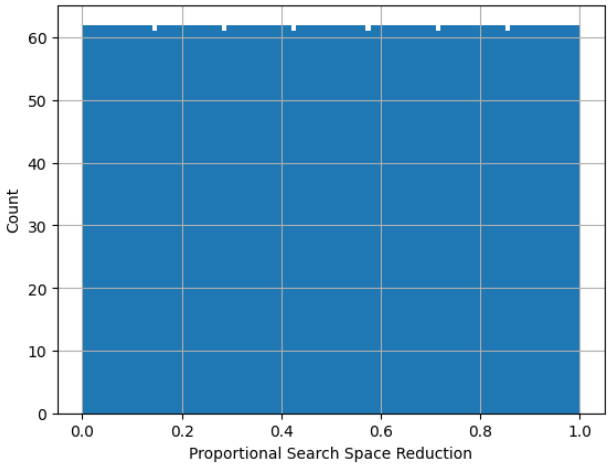

Figure 7. Resnet50 PSSR distribution histogram showing the number of images within each range of proportional reduction

the 13 images in the R@5 category. When the PSSR metric for each model is plotted as a histogram (see figs 7 - 9) 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, 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. As both use the same feature extraction layer, both embed to a 256 dimensional space and are plotted together in fig 10. The embeddings for VGG16 fill the space, while Resnet50 takes on a dense linear structure that embeds the 10,000 images very close together.

5. Discussion

The key to understanding the disparity between the very high performance of the Resnet50 network on the R@1% metric, 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 some insight; Hu *et al*. [8] published their Tensorflow 1 code from CVM-Net which was also used by

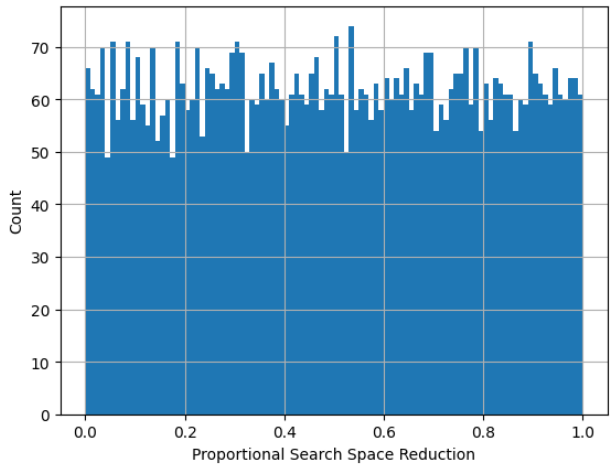

Figure 8. VGG16 PSSR distribution histogram showing the number of images within each range of proportional reduction

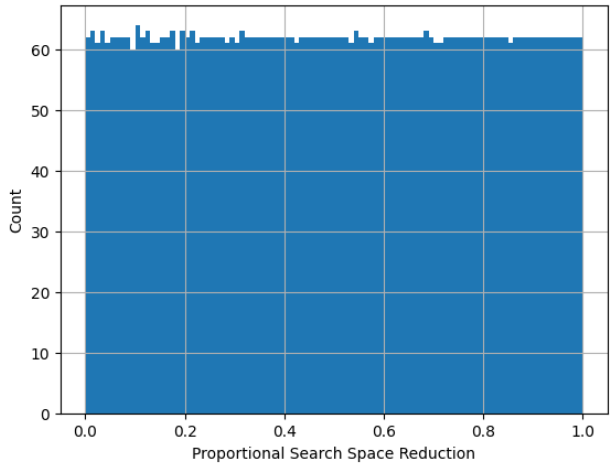

Figure 9. CVM-Net-I PSSR distribution histogram showing the number of images within each range of proportional reduction

Lui *et al*. [11] and Shi *et al*. [14] 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
```

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

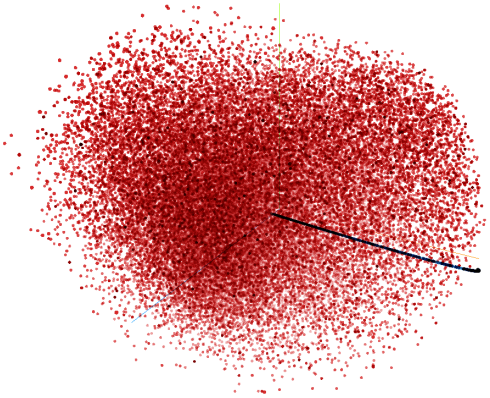

Figure 10. A plot of 10,000 image embeddings for both Resnet (in blue) and VGG16 (red)

```
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 study 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% follow the same algorithm:

Algorithm 1: Recall at K

Input : Set of distances D
Output: Count of distances satisfying the condition

Step 1: Identify the correct distance $(\hat{d})$;
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 $\hat{d} \leq d_{1\%}$;

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 $\hat{d}$, depending on the sorting algorithm used. I.e. if 1% of the dataset is 600 images, however 800 images correspond to the same location in the vector space, including the correct one, whether or not the correct image is included will depend on the order of the sorted elements that are equal to each other. 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.

6. Conclusion

This paper introduced a new metric for evaluating the performance of Cross-View Image Geo-Location models, a method of improving online triplet mining through hierarchical batching and introduces an architecture 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 asses the performance of future CVIG models.

References

[1] Relja Arandjelović, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition, 2016. 2

[2] Sudong Cai, Yulan Guo, Salman Hameed Khan, Jiwei Hu, and Gongjian Wen. Ground-to-aerial image geo-localization with a hard exemplar reweighting triplet loss. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8390–8399, 2019. 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 2, 4

[4] Abdulrahman Ghanem, Ahmed Abdelhay, Noor Salah, Ahmed Eldeen, Mohammed Elhenawy, Mahmoud Masoud, Ammar Hassan, and Abdallah Hassan. Leveraging cross-view geo-localization with ensemble learning and temporal awareness. 3 2023. 1, 3

[5] James Hays and Alexei A. Efros. im2gps: estimating geographic information from a single image. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 3, 4

[7] Daniel Hogan, Lucas Tindall, Ryan Ashley, Mona Gogia, and Adam Van Etten. Where in the world: A new data-set for cross-view image geolocalization. 2023. 1, 3

[8] Sixing Hu, Mengdan Feng, Rang Nguyen, and Gim Lee. Cvm-net: Cross-view matching network for image-based ground-to-aerial geo-localization. 06 2018. 2, 3, 4, 7

[9] Tsung-Yi Lin, Serge Belongie, and James Hays. Cross-view image geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013. 1, 2

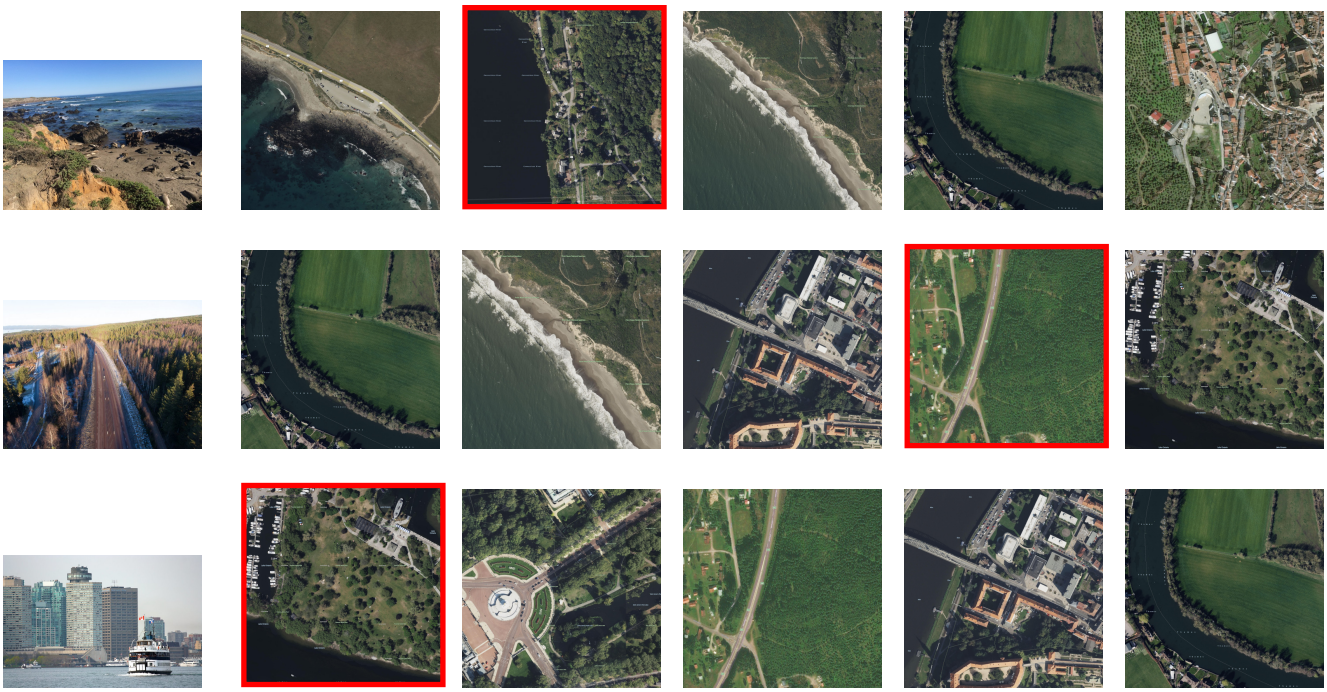

Figure 11. Three examples of the network performing at its best, The query image on the left, with the top 5 images shown in order of similarity, with the correct image shown in red

[10] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2, 4

[11] Liu Liu and Hongdong Li. Lending orientation to neural networks for cross-view geo-localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7

[12] Ziyu Rao, Jun Lu, Chuan Li, and Haitao Guo. A cross-view image matching method with feature enhancement. *Remote Sensing*, 15(8), 2023. 3

[13] Florian Schroff, Dmitry Kalenichenko, and James Philbin. FaceNet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, jun 2015. 2

[14] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. 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. 3, 7

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2014. 4

[16] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014. 2

[17] Nam Vo and James Hays. Localizing and orienting street views using overhead imagery, 2017. 2

[18] Tobias Weyand, Ilya Kostrikov, and James Philbin. PlaNet - photo geolocation with convolutional neural networks. In *Computer Vision – ECCV 2016*, pages 37–55. Springer International Publishing, 2016. 2

[19] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. *CoRR*, abs/1510.03743, 2015. 2, 3

[20] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1452–1464, 2018. 2

[21] Sijie Zhu, Taojiannan Yang, and Chen Chen. Revisiting street-to-aerial view image geo-localization and orientation estimation. In *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 756–765, 2021. 3

[22] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2021. 3