Urban Image Geo-Localization Using Open Data on Public Spaces

Glistrup, M.; Rudinac, S.; Jónsson, B.Þ.

DOI
10.1145/3549555.3549589

Publication date
2022

Document Version
Final published version

Published in
Proceedings of 19th International Conference on Content-based Multimedia Indexing

Urban Image Geo-Localization Using Open Data on Public Spaces

Mathias Glistrup
mpglistrup@gmail.com
IT University of Copenhagen
Copenhagen, Denmark

Stevan Rudinac
s.rudinac@uva.nl
University of Amsterdam
Amsterdam, The Netherlands

Björn Þór Jónsson*
bjorn@ru.is
Reykjavik University
Reykjavík, Iceland

ABSTRACT

In this paper, we study the problem of urban image geo-localization, where the aim is to estimate the real-world location in which an image was taken. Among the previous approaches to this task, we note three distinct categories: one only analyzes metadata; the other only analyzes the image content; and the third combines the two. However, most previous approaches require large annotated collections of images or their metadata. Instead of relying on large collections of images, we propose to use publicly available geographical (GIS) data, which contains information about urban objects in public spaces, as a backbone database to query images against. We argue that images can be effectively represented by the objects they contain, and that the spatial geometry of a scene—i.e., the positioning of these objects relative to each other—can function as a unique identifier for a particular physical location. Our experiments demonstrate the potential of using open GIS data for precise image geolocation estimation and serve as a baseline for future research in multimedia geo-localization.

CCS CONCEPTS

• **Information systems** → **Multimedia and multimodal retrieval**; **Geographic information systems**.

KEYWORDS

image geo-localization, urban multimedia data, multimedia retrieval

ACM Reference Format:
Mathias Glistrup, Stevan Rudinac, and Björn Þór Jónsson. 2022. Urban Image Geo-Localization Using Open Data on Public Spaces. In *International Conference on Content-based Multimedia Indexing (CBMI 2022), September 14–16, 2022, Graz, Austria.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3549555.3549589

1 INTRODUCTION

Image geo-localization, sometimes also referred to as geo-tagging, geolocating, geospatial localization, location inference, or location estimation, is the task of estimating or inferring the real-world location in which an image was taken. It typically involves extracting and analyzing image content or the accompanying metadata to find geographical clues. While there are many approaches to image geo-localization, it is by and large still an unsolved task.

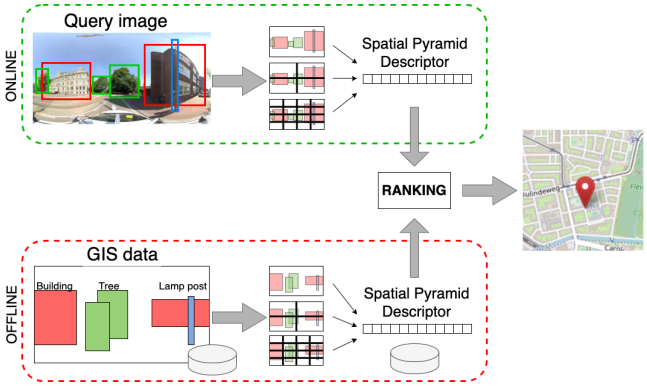


Figure 1: Overview of our architecture using spatial pyramid descriptors to geolocate images. The lower section represents an offline process that extracts open GIS data to build a database of scene descriptors. At query time, corresponding scene descriptors are built from the query image, and compared to the GIS-based database to rank potential locations.

Automatic geo-localization holds great potential in many different contexts. Online image and video sharing services, such as Flickr and YouTube, use a variety of techniques to automatically generate tags, from object-level information (e.g., *dog*, *man*, or *surfboard*) to information about scene and setting (e.g., *indoor* or *outdoor*, *restaurant* or *office*) or—most importantly in this context—the real-world geographical location of the scene. Such tags can make it easier for users to find the content they are looking for [7].

Another major application area for geo-localization algorithms is the work of digital investigators. Law enforcement agents often confiscate electronic equipment or intercept electronic messages containing images depicting criminal offenses. In some cases, accurately geo-localizing these images can help solve or prevent violent crimes, drug related activities, or even human trafficking [19]. For investigative journalists and researchers who work with image and video files, the task of geo-localization has become a standard part of the toolkit for verifying the authenticity of the material [32].

Image geo-localization has been a topic of intensive research, and the field has significantly widened within the last decade following the rapid evolution of modern machine learning capabilities. Overall, most of the related work can be divided into three categories, based on the data used: approaches utilizing visual content [2, 11, 16, 22, 25–27, 29, 30, 35, 37, 38]; approaches relying on metadata, such as image title, tags and description [34]; and multimodal approaches combining the content and metadata [5, 14].

Generally speaking, systems that consider image metadata tend to fare much better than those that do not [14]. However, different

*Research conducted while the author was with the IT University of Copenhagen.

CBMI 2022, September 14–16, 2022, Graz, Austria

ACM ISBN 978-1-4503-9720-9/22/09…$15.00
https://doi.org/10.1145/3549555.3549589

approaches often have widely different scopes: while some systems cover the entire world [16], most work on a much smaller scale, e.g., within a single city [3]. It is thus hard to compare results directly between different systems, since the setups are rarely comparable.

One major drawback to all of the strategies discussed above is that they require large collections of geo-referenced images or image tags as a backbone for their systems. Such collections are not always readily available, and those that exist are typically tailored for a very specific task in terms of structure, geographical coverage, and the type and representation of annotations.

In this paper, we take a different approach to image geo-localization. Rather than relying on large image collections, we exploit the fact that cities in many countries are legally obliged to keep accurate GIS data about the state of assets and other objects in public spaces. Such data is often openly available, including the size and location of many urban objects, such as *buildings*, *trees*, and *lamp posts*. We propose using this data as a backbone for our system, thus reducing the need for large annotated collections by instead comparing automatically detected objects in an input image directly with the indexed GIS data. Figure 1 illustrates the approach.

While the general idea of utilizing GIS data for urban image geo-localization is not entirely new [3], we propose an approach to geo-localization that is far more intuitive, computationally inexpensive and in line with how humans understand and analyze images and scenes [1, 15, 36]. Both journalists and law enforcement agents are used to working with eye witnesses who describe a scene of a crime or a location of interest. Having instruments that can assist in determining the real-world location based on sparse, high-level descriptions of a scene could potentially be of great importance. For example, one could imagine that even though a witness may not know the location of the scene, they may be able to describe the high-level layout of objects and their spatial geometry. Being able to convert sparse descriptions into a set of probable locations could be very useful.

We investigate the potential of geo-localizing the images solely based on how well the automatically detected objects match openly available GIS data. Reducing the need for large collections of annotated images by using GIS data as a backbone instead could prove valuable for future research and real-world applications.

With this paper, we make the following contributions:

- We present a novel approach for the task of geo-localizing images that neither relies on potentially misleading metadata nor large image collections.
- We demonstrate that pre-processed, publicly available GIS data can be used as a lookup database for matching detected objects in an image.
- We show that performance can be improved drastically by incorporating information about scene geometry.

The remainder of this paper is structured as follows: In Section 2 we discuss related work in geo-localization. Section 3 then outlines the proposed architecture and its implementation. Section 4 details the experimental setup, while Section 5 presents and discusses the results. Finally, Section 6 concludes the paper.

2 BACKGROUND AND RELATED WORK

Several approaches to geo-localization have been suggested throughout the last two decades. Many have surfaced from evaluation initiatives in computer vision and multimedia communities, such as the Places challenge [39] and the MediaEval benchmark [17]. Efforts have also been made in geo-localizing media other than surface images, e.g., social media text [24], or satellite images [12, 13]. These are outside the scope of this paper.

With regard to the type of data they rely on, most existing image geo-localization approaches fall under one of the three general categories elaborated in the following subsections. It is worth noting that due to a lack of standard for evaluating geo-localization algorithms and systems, it is still hard to directly compare the different approaches. Not only do they cover very different scopes in terms of geographical reach, but they also work on widely different datasets and are evaluated on a variety of different task-specific metrics.

Using Only Metadata. Van Laere et al. [34] used a language model classifier based on the tags from more than 8.6 million Flickr images to narrow down the search space to a limited geographical area. Then they performed similarity search within that area and used a weighted average of the nearest neighbors as a final location estimate. This approach, however, is not suitable for our case since metadata will not always be available.

Using Only Image Content. Several researchers have used SIFT features to geo-localize images, either with simple voting [29] and dynamic pruning [37], or with Generalized Minimum Clique Graphs [38]. The IM2GPS system combined a collection of image features to estimate geolocation as a probability distribution over the Earth's surface [11]. Penatti et al. [27] proposed the *bag-of-scenes* and represented each scene by low-level features in a dictionary.

More recently, many have turned to convolutional neural networks (CNN) for geo-localization. Arandjelovic et al. [2] proposed a CNN architecture specifically designed with geo-localization in mind, with the main component being the task-specific NetVLAD layer. Medina et al. [22] combined NetVLAD features with clustering and density based voting to geo-localize videos. Peddada and Hong [26] used two-stage CNN classification to first determine a city and then to determine an exact location within that city. Weyand et al. [35] trained a CNN on 126 million noisy images and presented two models: A single-image model and a model that takes a photo album as input. Using a sequence of images rather than a single image improved geo-localization accuracy almost 10-fold. Seo et al. [30] used the same system with overlapping partitions of the Earth, greatly improving accuracy. Müller-Budack et al. [25] included scene classification in the learning process of a CNN for state-of-the-art performance. Kordopatis-Zilos et al. [16] combined classification and retrieval practices to achieve state-of-the-art on a global dataset. However, none of these approaches are suitable for our case either, since they rely on large collections of images.

Combining Image Content and Metadata. With both visual keypoints and image tags, Crandall et al. [5] determined geolocations using mean shift clustering on over 30 million images and substantially improved geo-localization accuracy. Kelm et al. [14] combined visual features with toponym and gazetteer look-ups of words in the image tags, accurately geo-localizing 40% within a 100 meter radius. While this kind of multimodal approach generally fares significantly better than those that use only either metadata or image

content, it is still not suitable for our case for the same reasons mentioned in relation to the previous two categories.

Using GIS Data for Geo-Localization. Perhaps closest to our approach is that of Ardeshir et al. [3], who first suggested to use publicly available GIS data for geo-localization. They perform similarity search between GIS objects and detect objects in the query image over a dense grid of locations that are 20 meters apart. At each location the search is performed in a 360° fashion at 20° intervals for 18 different angles in total. The similarity search is treated as a graph matching problem where the two sets of objects can be considered opposite sides of a bipartite graph, making the assumption that the geometric model between the two can be approximated with an affine transformation. A Random Sample Consensus, or RANSAC, algorithm is used to assign edges between objects of the same class in each set such that the correspondence best fit a global affine model, resulting in a single similarity score for each location. While the authors demonstrate the potential in using GIS data for geo-localization, graph matching is both computationally expensive and not very intuitive in cases where the query is a rough scene description provided by e.g. an eyewitness recalling a situation.

Datasets for Geo-Localization. As with other types of computer vision and multimedia tasks, many researchers collect and annotate their own datasets tailored to specific purposes. However, some location annotated datasets have been frequently used for the task of geo-localization, including *IM2GPS* [11], *Mmsys-14* [31], and *Placing-16* [33]. Again, because we aim to avoid using large collections of images, these datasets are not suitable for our work.

3 APPROACH AND ARCHITECTURE

Our goal is to create a system that takes an image as input and outputs a proposed physical location. We follow a strict rule of not trusting any kind of metadata from the input image, instead making the assumption that the pixels in the image are trustworthy, and everything else is not. Our approach relies solely on visible objects in the image that are queried against a database of preprocessed GIS data containing information about the physical placement of different urban objects. A location is estimated by matching detected objects and their spatial distribution with that of the GIS objects. Contrary to previous approaches, we compare two different types of information on the query side and the database side: Our input is an image, and our backbone data is GIS data from open sources.

Figure 1 shows our proposed architecture. As a preprocessing step we build a database of scene descriptors for every scene in our dataset. For new query images, we detect objects and build a similar scene descriptor from the output of the object detector. The input descriptor is then matched with the representations of all scenes stored in the database. The top ranked scene determines the estimated location for the query image. The following sections will describe each component of our approach in more detail.

3.1 Building a Database of Scene Descriptors

Our pipeline works with any collection of GIS object data that has been mapped to 2-dimensional panoramic scenes and annotated with location coordinates.

Using abstract features, such as SIFT or SURF, is the traditional approach to represent the shapes in the image, and there exist tried-and-tested methods of matching sets of SIFT features. The same applies to other low-level image features such as texture and color. However, these approaches all require collections of images, which we have argued against using. Instead, we want to build scene descriptors that are purely based on object metadata and include information about the types of objects present in the scene, as well as their (rough) spatial geometry.

One way of accomplishing this is to rank scenes using graph matching in the same fashion as Ardeshir et al. [3], as described in Section 2, rather than to pre-compute descriptors for the dataset. However, as detailed in Section 2, a major caveat of such approach is a high computational cost associated with graph matching, which makes it inapplicable in e.g. interactive geo-localization scenarios.

By instead choosing descriptors that we can build in advance, we move a large part of our computational cost offline. This allows for easier matching with query descriptors. The scenes in our particular dataset assume a panoramic format, which requires that the query image is oriented the same way as the scenes. To do this, we only need the *heading* of the image, meaning the direction the center of the image is facing, which is included in our dataset. However, we conjecture that our architecture should work on any kind of image, requiring only minor adjustments to incorporate separate representations for e.g. each viewing direction at a given location.

Global Object Histogram. Perhaps the simplest representation satisfying our requirements would be a count of objects present in the scene. Similar representations have been widely used in different multimedia applications, such as video search and event detection [21, 28]. The underlying intuition is that scene compositions are somewhat unique as long as the number of objects is relatively high. Depending on the types of objects present in the GIS data, it is not uncommon that scenes contain several dozens of objects. The combination alone of that many objects in a scene could in theory be sufficient to identify potential candidates for a location.

The global object histogram consists of a K-vector where K is the number of object classes in the dataset. Values in the vector represent the number of instances of each object class that are present in the scene. This type of descriptor does not relay information about how the objects are organized, but only whether objects are present or not.

Spatial Pyramid Descriptors. To incorporate information about the geometry of the scene, we use the spatial pyramid descriptor defined by Grauman and Darrell [9] for its ability to accurately capture the spatial distribution of features, or in our case, objects. This approach has the advantage of corresponding fairly well with how humans analyze and describe images. As described in the introduction of this paper, we imagine a system that can be used with eyewitness accounts of an event that needs geo-localization. Even though a person might not remember many details of the place in question, it is not unlikely that they remember bits of information regarding the objects present. For example, they might be able to say, "There was a large building on the left hand side and a couple of trees to the right of it. There were also two lamp posts on the street in front of the building, and a smaller building all the way to the right." This minimal amount of information can easily be translated to a spatial

pyramid descriptor, and may even result in some partial matches in the dataset. While this is not something we examine further in this paper, it is something we keep in mind and suggest for future research. For all the reasons mentioned here, we believe the spatial pyramid descriptors are far more intuitive than the alternatives.

To build the descriptors, we follow the same procedure as described by Lazebnik et al. [18] and Bosch et al. [4]. A scene is partitioned into increasingly finer grids of even-sized cells and objects are counted in a histogram for each cell. Every level l in $0, \ldots, L$ has 2^l cells along each dimension, and every cell is represented by a K-vector histogram, with K being the number of classes in the dataset. Thus, every level can be represented by a $C \cdot K$-vector where C is the number of cells on that level. The final descriptor is a weighted concatenation of all levels with a total dimensionality of $K \sum_{l \in L} 4^l$. Weights w for each level l are defined by Lazebnik et al. [18] as $w^l = \frac{1}{2^{L-l}}$ which puts more emphasis on higher levels with finer grids capturing the spatial layout in more detail. Finally, we normalize the descriptor to prevent scenes with many objects from being weighted more strongly than those with fewer objects.

We represent the location of each object by the center point of its bounding box. While using two coordinates instead of one—e.g., opposite corners of the bounding box—would preserve information about the size of the objects, this would also result in descriptors being twice as large. A spatial pyramid descriptor with $L = 5$ is already a 34,125-dimensional vector, associated with a substantial computational time. Since doubling of the descriptor size could unnecessarily put additional burden to interactivity, for the purpose of this paper we have not explored that further and choose to work with the center point.

3.2 Image Processing: Detecting Urban Objects

Given a query image, the first (online) step in determining its geolocation is detecting the visible objects and their approximate location within the image. From that we build the scene descriptor following the exact same procedure as detailed in Section 3.1.

For detecting objects we use a Cascade R-CNN model that has been pretrained on the weak and noisy PanorAMS dataset [10]. The model has a ResNet-50 backbone pretrained on the ImageNet dataset [6] with a 6 level FPN [20]. For details please refer to [10].

The detections are *weak*, meaning they contain inaccurate bounding boxes, and *noisy*, meaning they contain false positives. This is to be expected, since our scenes are typically complex and our objects nontrivial. The accuracy of the system thus relies on the quality and precision of the projected GIS objects on one hand, and the quality of the object detections on the other hand.

We empirically found that a threshold for the resulting bounding boxes of 0.3 gave the best results, meaning that all detections with a confidence level below 0.3 are ignored. To demonstrate the proof of concept, input images are here expected to be in panorama format since that is the format of the used dataset, although there is nothing standing in the way of deploying the system on the other formats.

3.3 Scene Ranking: Matching Descriptors

The second online stage is comparing the query descriptor with those in the database. Scoring the entire backbone dataset in this fashion will result in a ranked list of scenes, sorted by their relevance to the query image. We use the χ^2 distance, which expresses the relative distance between two numbers rather than the absolute distance. Finally, we return the location of the single highest ranked scene as our estimated location for the query image.

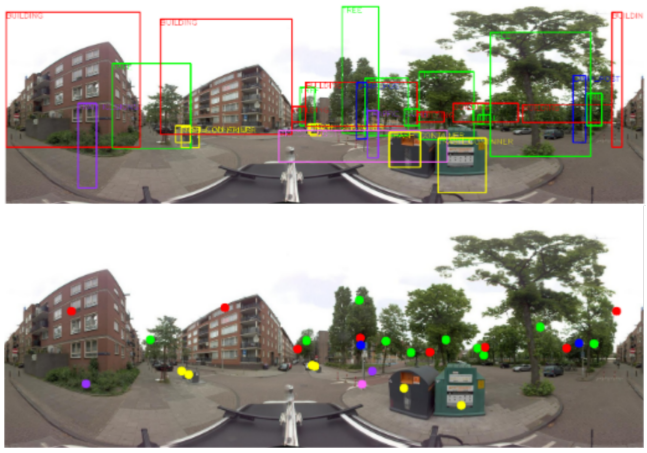


Figure 2: (top) Example image from the PanorAMS dataset with bounding boxes. (bottom) Same image with dots representing the center point of the boxes.

4 EXPERIMENTAL SETUP

Overall, our approach to evaluation is that we run the entire collection of images from our dataset through the geo-localization pipeline and evaluate performance on each image, comparing an image on the query side with processed GIS data on the database side. This section provides the details about dataset, queries and metrics used to evaluate performance of our approach.

Dataset. In this paper we use the PanorAMS dataset by Groenen et al. [10]. The dataset covers the city of Amsterdam in the Netherlands and is built from publicly available GIS data provided by the municipality of Amsterdam. It contains a total of 779,360 panoramic images annotated with more than 14 million weak and noisy bounding boxes of 24 urban object classes. It was built in an automated way using simple camera models and occlusion handling, thus eliminating the need for creating manual annotations. It is worth noting that although the dataset contains panorama images, we only rely on the bounding boxes resulting from the processing of GIS data and the spatial layout of the 2D mapping of those objects. The actual images in the dataset have only been used to train the object detector as described in Section 3.2 and to evaluate the system.

We work with a subset of the PanorAMS dataset consisting of roughly 1% of the full dataset, provided to us by the authors. The subset consists of 7,436 locations from 10 neighbourhoods and contains 19,178 unique objects and 140,119 bounding box projections. Every scene in the dataset is annotated with latitude-longitude coordinates. The subset is heavily imbalanced in terms of objects, where the classes *tree* and *building* represent 75% of all bounding boxes, and in terms of neighbourhoods, where one neighbourhood represents roughly 22% of all scenes and another only 3%.

Evaluation Protocol. All evaluations of our system follow the same protocol: A query image is run through the object detector and the resulting bounding boxes are converted to a scene descriptor. The image query descriptor is then compared to all scene descriptors in the backbone dataset, which are further ranked by increasing distance. The geolocation of the highest ranked scene is then selected as an estimated location for the query image.

Since the previous approaches to geo-localization are hard to compare with directly due to a difference in setting and task, we do not use previous work as baselines for our experiments. Instead we use random guessing as a baseline and primarily compare the global object histogram with spatial pyramids.

Evaluation Metrics. To evaluate our system, we measure the real-world distance between the estimated location and the ground truth location for every evaluated scene. Since geographical coordinates can be considered as a continuous space it does not make sense to use traditional measurements like precision and recall. Instead, we evaluate our system with the *Precision@k* formula defined by Medina et al. [22], which measures the percentage of evaluated scenes that are predicted within k meters of the correct location. We evaluate for $k \in [10, 100, 500, 1000]$.

To further evaluate our descriptors, inspired by the Hit Rate at top k (HR@k) measure widely adopted in evaluation of recommendation systems, we use a top-k score that expresses the percentage of query images where the ground truth scene is included in the top k ranked scenes. We report the results for $k \in [1, 10]$. Similar to the Precision@k metric, we define top-k as:

$$Top(k) = \frac{\sum_{n=1}^{|N|} 1\{gt\_rank_n <= k\}}{|N|} \quad (1)$$

where $|N|$ is the total number of scenes that are evaluated, and $\{gt\_rank_n <= k\}$ is the number of evaluated images where the ground truth rank is lower than or equal to k.

Furthermore, we perform an ablation study in which single classes of objects are removed from the dataset to see how it affects performance. If performance increases when removing a certain class of object, we can deduce that that specific class contributes negatively to the results, and vice-versa. For computational reasons, and because there are 24 classes in our dataset, we limited the scope of these tests to only include randomly chosen 500 query images. We evaluate on spatial pyramid descriptors of depth 4.

5 EXPERIMENTAL RESULTS

In this paper we investigate the feasibility of an object-based approach to geo-localization that handles two different types of information, i.e., images on the query side and GIS data on the database side, respectively. In particular, through the experiments we aim to answer the following questions:

- What is the usefulness of publicly available GIS data on urban spaces for geo-localization purposes?
- How useful are individual object categories for image geo-localization?
- How does the geo-localization performance vary across the urban neighbourhoods?

The following sections address each of these questions in turn.

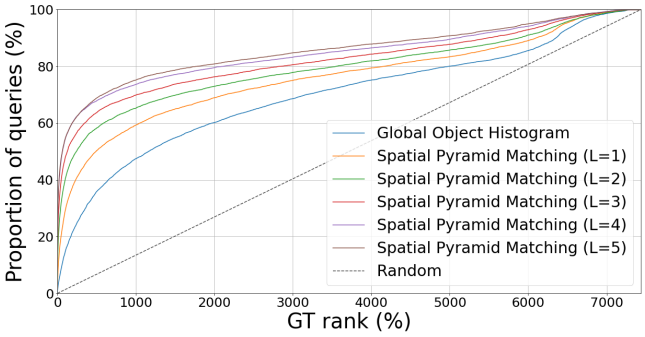


Figure 3: Evaluation of ranking using different descriptors. The plot depicts the cumulative GT ranks for the entire collection of 7,436 scenes in the dataset.

5.1 Geo-Localization Performance

We evaluate the quality of scene ranking our pipeline produces following the procedure laid out in Section 4. We study the effect of only including a global object count versus the effect of including spatial information at different granularity as well.

Ranking. Figure 3 shows the ranking results, where the x-axis represents the rank of the ground truth (GT) location, and the y-axis represents the percentage of queries where the GT location falls within a given rank. The main observation here is that including spatial information improves the ranking performance of our architecture. Furthermore, the performance increases on all parameters as we move towards a higher L and thereby a finer partitioning of the scene. However, as L grows higher, the relative increase in performance becomes smaller and smaller, indicating that there is a limit to the efficacy of partitioning further.

Estimating Geolocation. Table 1 shows the results for estimating geolocation coordinates, using all images from the dataset as queries, as well as the top-1 and top-10 ranking.

We observe how our first baseline—randomly selecting a location from the backbone dataset—is a very poor location predictor. We also observe a great improvement when we represent the image with a global object histogram. Note that this is equal to a spatial pyramid descriptor of depth 0, meaning that no partitioning of the scene has been done and no spatial information is preserved.

As Table 1 shows, the deeper we make the pyramid descriptors — i.e., the higher values of L we choose — the better the accuracy becomes. However, it is also clear that the effect slows down as L grows higher, just as we see in the ranking results. There is little difference in performance between $L = 4$ and $L = 5$, and while the P@10 score increases slightly when moving one level up, the P@100 score actually decreases. We conjecture that with $L > 5$, further gains are unlikely and performance may even degrade.

5.2 Importance of Different Classes

Figure 4a shows the results of our ablation experiment, where we evaluate the importance of individual classes for the overall performance. The x-axis represents the normalized distance between estimated and ground truth locations, and the y-axis represents

Table 1: Geo-localization results for different depths of spatial pyramid descriptors. GOH = Global Object Histogram.

Method	P@10	P@100	P@500	P@1000	Top1	Top10
Random	0.05	1.83	11.47	16.12	-	-
GOH	1.47	10.45	25.79	30.12	0.28	2.78
L = 1	6.23	20.28	36.14	40.77	1.79	8.57
L = 2	9.49	25.24	39.30	43.75	4.71	16.98
L = 3	14.21	30.82	43.91	47.88	8.20	24.58
L = 4	19.92	**36.71**	48.94	53.00	12.82	32.61
L = 5	**20.19**	36.22	**49.93**	**53.13**	**13.13**	**32.72**

the proportion of the evaluated subset, consisting of 500 randomly selected scenes. We have evaluated on all 24 classes, but for visualization purposes the figure only includes results for the 5 classes that affect results the most. When we exclude the *buildings* class from our descriptors, the performance degrades significantly, indicating that this class contributes a lot to the overall accuracy of our system. For all other classes, these effects are minor.

Furthermore, we empirically found a correlation between the number of GIS objects in a scene and the ranking accuracy. This correlation is particularly clear for buildings, trees, and lamp posts, where the ranking of the GT scene improves as the number of those objects in the scene increases. For classes with fewer total objects present in the dataset, this correlation becomes less clear.

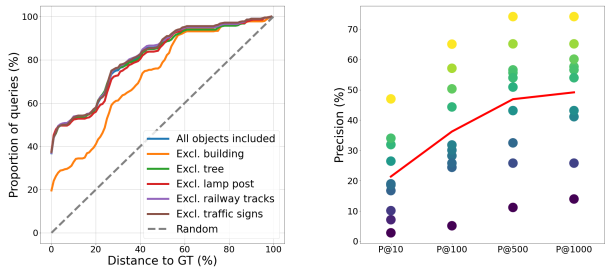
5.3 Variance Across Neighbourhoods

Figure 4b shows the variance in geo-localization accuracy across neighbourhoods. The y-axis represents precision, meaning the number of scenes in our dataset that are geo-localized correctly within a certain threshold, and along the x-axis we see results for four such thresholds. Colored dots represent neighbourhoods. We observe how results vary widely across neighbourhoods. In the lowest scoring neighbourhood we only manage to correctly geolocate less than 3% of images within 10 meters of their ground truth location. For the highest scoring neighbourhood this number is almost 47%, more than twice as many as the whole dataset on average.

Much of this can be attributed to inaccurate detections of the classes *building* and *tree*. We see some correlation between the number of detected objects of these classes and the ranking accuracy across all neighbourhoods.

5.4 Discussion of Results

While the results of the ranking experiments clearly show the promise of the approach, our analysis also illustrates the issues stemming from using GIS objects as references for ranking. In many cases, ground truth locations are ranked very poorly, and some arbitrary locations are ranked high instead. One potential reason for this could be that many scenes are generic and contain the same combination of objects, which leads to very similar descriptors. However, we do not see this reflected in the similarity scores of the ranked dataset. Therefore, we believe that it is primarily due to the large difference between the objects suggested by the object detector and the noisy GIS objects in the PanorAMS dataset.



(a) Classes **(b) Neighbourhoods**

Figure 4: (a) Ablation study on the class importance on a subset of 500 randomly selected scenes. Only the 5 classes with the biggest impact are visualized. (b) Variance in precision across all neighbourhoods, evaluated on the full dataset.

Nearest neighbour retrieval, without advanced indexing structures, has a linear complexity. We have not explored the myriad available optimizations to the ranking process. We expect that some of the many different feature reduction and selection or, perhaps, search re-ranking approaches from the literature could be successfully applied to this data [23, 28]. Furthermore, the pyramid descriptors are extremely sparse, so using sparse representations and corresponding high-dimensional indexing approaches should significantly decrease computational complexity and running time. In a full implementation, queries will thus run in sub-second time. In contrast, the approaches based on graph matching, such as [3], are typically associated with an exponential complexity [8].

6 CONCLUSION

This paper presents a novel approach to the task of image geo-localization. It explores the feasibility of using openly available GIS data as a backbone dataset and shows that physical locations can be represented by the urban objects they contain. By focusing on GIS information rather than low-level image features, we greatly reduce the need for large, annotated collections of images or image metadata. We show that even just a simple count of objects proposed by an object detector significantly improves performance over random guessing. Additionally, we show that including information about the scene's spatial geometry drastically improves the performance further. By building spatial pyramid descriptors from nothing but the scene's objects, our system correctly geo-localizes 20.19% of all query images within just 10 meters of their correct location. For one neighbourhood in our dataset this number was even 46.99%. Our work demonstrates the usefulness and potential in using publicly available GIS data for the task of geo-localization. The intuitiveness of our object-based spatial pyramid descriptors makes it easy to imagine such a system being used in the fields of law enforcement, journalism, and other areas relying on digital investigation.

ACKNOWLEDGMENTS

This work was supported in part by the European Regional Development Fund (ERDF) Interreg VB North Sea Region Programme project Smart Cities + Open Data Re-use (SCORE).

REFERENCES

[1] Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA.

[2] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. 2015. NetVLAD: CNN Architecture for Weakly Supervised Place Recognition. *Computing Research Repository (CoRR)* abs/1511.07247 (2015).

[3] Shervin Ardeshir, Amir Roshan Zamir, Alejandro Torroella, and Mubarak Shah. 2014. GIS-Assisted Object Detection and Geospatial Localization. In *Proc. European Conference on Computer Vision (ECCV)*. Zürich, Switzerland, 602–617.

[4] Anna Bosch, Andrew Zisserman, and Xavier Muñoz. 2007. Representing Shape With a Spatial Pyramid Kernel. In *Proc. International Conference on Image and Video Retrieval (CIVR)*. Amsterdam, The Netherlands, 401–408.

[5] David J. Crandall, Lars Backstrom, Daniel P. Huttenlocher, and Jon M. Kleinberg. 2009. Mapping the World's Photos. In *Proc. International Conference on World Wide Web (WWW)*. Madrid, Spain, 761–770.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Miami, Florida, 248–255.

[7] Jianlong Fu and Yong Rui. 2017. Advances in Deep Learning Approaches for Image Tagging. *APSIPA Transactions on Signal and Information Processing* 6 (2017).

[8] M. R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman.

[9] K Grauman and T. Darrell. 2005. Pyramid Match Kernels: Discriminative Classification With Sets of Image Features. *Computer Science and Artificial Intelligence Laboratory Technical Report* (2005).

[10] Inske Groenen, Stevan Rudinac, and Marcel Worring. 2022. PanorAMS: Automatic Annotation for Detecting Objects in Urban Context. (2022). Under Review.

[11] James Hays and Alexei A. Efros. 2008. IM2GPS: Estimating Geographic Information From a Single Image. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, Anchorage, AK, USA.

[12] Sixing Hu, Mengdan Feng, Rang M. H. Nguyen, and Gim Hee Lee. 2018. CVM-Net: Cross-View Matching Network for Image-Based Ground-to-Aerial Geo-Localization. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, UT, USA.

[13] Sixing Hu and Gim Hee Lee. 2020. Image-Based Geo-Localization Using Satellite Imagery. *International Journal of Computer Vision* 128, 5 (2020), 1205–1219.

[14] Pascal Kelm, Sebastian Schmiedeke, Jaeyoung Choi, Gerald Friedland, Venkatesan Nallampatti Ekambaram, Kannan Ramchandran, and Thomas Sikora. 2013. A Novel Fusion Method for Integrating Multiple Modalities and Knowledge for Multimodal Location Estimation. In *Proc. International Workshop on Geotagging and its Applications in Multimedia (GeoMM@ACM)*. Barcelona, Spain, 7–12.

[15] Alex Kendall, Yarin Gal, and Roberto Cipolla. 2018. Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics. In *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*. Salt Lake City, Utah, USA.

[16] Giorgos Kordopatis-Zilos, Panagiotis Galopoulos, Symeon Papadopoulos, and Ioannis Kompatsiaris. 2021. Leveraging EfficientNet and Contrastive Learning for Accurate Global-scale Location Estimation. In *Proc. ACM International Conference on Multimedia Retrieval (ICMR)*. Taipei, Taiwan, 155–163.

[17] Martha Larson, Mohammad Soleymani, Pavel Serdyukov, Stevan Rudinac, Christian Wartena, Vanessa Murdock, Gerald Friedland, Roeland Ordelman, and Gareth J. F. Jones. 2011. Automatic Tagging and Geotagging in Video Collections and Communities. In *Proc. ACM International Conference on Multimedia Retrieval (ICMR)*. Trento, Italy, Article 51, 8 pages.

[18] Svetlana Lazebnik, Cordelia Schmid, and Jean Ponce. 2006. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. New York, NY, USA.

[19] Gido Limperg. 2020. *Fusing Object Representations for Hotel Instance Retrieval.* Master's thesis. University of Amsterdam.

[20] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. 2016. Feature Pyramid Networks for Object Detection. *Computing Research Repository (CoRR)* abs/1612.03144 (2016).

[21] Masoud Mazloom, Efstratios Gavves, and Cees G. M. Snoek. 2014. Conceptlets: Selective Semantics for Classifying Video Events. *IEEE Transactions on Multimedia* 16, 8 (2014).

[22] Salvador Medina, Zhuyun Dai, and Yingkai Gao. 2018. Where is This? Video Geolocation Based on Neural Network Features. *Computer Research Repository (CoRR)* abs/1810.09068 (2018).

[23] Tao Mei, Yong Rui, Shipeng Li, and Qi Tian. 2014. Multimedia Search Reranking: A Literature Survey. *ACM Comput. Surv.* 46, 3, Article 38 (2014), 38 pages.

[24] Stuart E. Middleton, Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Yiannis Kompatsiaris. 2018. Location Extraction from Social Media: Geoparsing, Location Disambiguation, and Geotagging. *ACM Trans. Inf. Syst.* 36, 4 (2018), 40:1–40:27.

[25] Eric Müller-Budack, Kader Pustu-Iren, and Ralph Ewerth. 2018. Geolocation Estimation of Photos Using a Hierarchical Model and Scene Classification. In *Proc. 15th European Conference on Computer Vision (ECCV)*. Munich, Germany, 575–592.

[26] Amani V. Peddada and James Hong. 2016. Geo-Location Estimation With Convolutional Neural Networks. http://cs231n.stanford.edu/reports/2015/pdfs/CS231N_Final_Report_amanivp_jamesh93.pdf

[27] Otávio Augusto Bizetto Penatti, Lin Tzy Li, Jurandy Almeida, and Ricardo da Silva Torres. 2012. A Visual Approach for Video Geocoding Using Bag-of-Scenes. In *Proc. International Conference on Multimedia Retrieval (ICMR)*. Hong Kong, China, 53.

[28] Stevan Rudinac, Martha Larson, and Alan Hanjalic. 2012. Leveraging Visual Concepts and Query Performance Prediction for Semantic-Theme-Based Video Retrieval. *International Journal of Multimedia Information Retrieval* 1 (2012), 263–280.

[29] Grant Schindler, Matthew A. Brown, and Richard Szeliski. 2007. City-Scale Location Recognition. In *Proc. Conference on Computer Vision and Pattern Recognition CVPR*. Minneapolis, MN, USA.

[30] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. 2018. CPlaNet: Enhancing Image Geolocalization by Combinatorial Partitioning of Maps. *Computing Research Repository (CoRR)* abs/1808.02130 (2018).

[31] Hatem Mousselly Sergieh, Daniel Watzinger, Bastian Huber, Mario Döller, Elöd Egyed-Zsigmond, and Harald Kosch. 2014. World-Wide Scale Geotagged Image Dataset for Automatic Image Annotation and Reverse Geotagging. In *Proc. Multimedia Systems Conference (MMSys)*. Singapore, Singapore, 47–52.

[32] Craig Silverman (Ed.). 2021. *Verification Handbook: For Disinformation And Media Manipulation.* European Journalism Centre, Maastricht, The Netherlands.

[33] Bart Thomee, Olivier Van Laere, Claudia Hauff, and Jaeyoung Choi. 2017. A Mediaeval Benchmark on Multimedia Location Prediction! https://multimediacommons.wordpress.com/placing-task/

[34] Olivier Van Laere, Steven Schockaert, and Bart Dhoedt. 2011. Finding Locations of Flickr Resources Using Language Models and Similarity Search. In *Proc. ACM International Conference on Multimedia Retrieval (ICMR)*. Trento, Italy, 48.

[35] Tobias Weyand, Ilya Kostrikov, and James Philbin. 2016. PlaNet - Photo Geolocation With Convolutional Neural Networks. *Computing Research Repository (CoRR)* abs/1602.05314 (2016).

[36] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. 2016. Image Captioning With Semantic Attention. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, Nevada.

[37] Amir Roshan Zamir and Mubarak Shah. 2010. Accurate Image Localization Based on Google Maps Street View. In *Proc. European Conference on Computer Vision (ECCV*. Heraklion, Greece, 255–268.

[38] Amir Roshan Zamir and Mubarak Shah. 2014. Image Geo-Localization Based on Multiple Nearest Neighbor Feature Matching Using Generalized Graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36, 8 (2014), 1546–1558.

[39] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2018. Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 6 (2018), 1452–1464.