Looking around in the neighbourhood: Location estimation of outdoor urban images

Jie Huang1  |  Haozhi Huang2,3

1 Faculty of Information Technology, Macau University of Science and Technology, Avenida Wai Long, Macau, China
2 School of Computer Science, Fudan University, Shanghai, China
3 Zhuhai Fudan Innovation Institute, Zhuhai, Guangdong Province, China

Abstract
Visual geolocalisation has remained as a challenge in the research community: Given a query image, and a geo-tagged reference database, the goal is to derive a location estimate for the query image. We propose an approach to tackling the geolocalisation problem in a four-step manner. Essentially, our approach focuses on re-ranking the candidate images after image retrieval, by considering the visual similarity of the candidate and its neighbouring images, to the query image. By introducing the neighbouring images, the visual information of a candidate location has been enriched. The evaluation has been conducted on three street view datasets, where our approach outperforms three baseline approaches, in terms of location estimation accuracy on two datasets. We provide discussions related to, firstly, whether using deep features for image retrieval helps improve location estimation accuracy, and the effectiveness of geographical neighbourhoods; secondly, using different deep architectures for feature extraction, and its impact on estimation accuracy; thirdly, investigating if our approach consistently outperforms the classic 1-NN approach, on two datasets with significant difference in visual elements.

1 INTRODUCTION
Humans are able to recognize places and infer locations from images: Provided with a photo of the Ruins of St. Paul’s, some may instantly know it is a landmark in Macau, China. A more challenging example would be an image of coconut trees by the sea. In this case, we may have trouble deriving an accurate location estimate from this generic photo. However, we know this photo depicts “somewhere in the tropical areas”. And this knowledge helps narrow down the list of candidate locations. Data association [1] and semantic reasoning both play an essential role in the human ability of visual place recognition [2]. Hays and Efros pointed out that semantic reasoning is a huge challenge computationally in 2008 [2]. Luckily, large image collections available on the Internet have paved the way for data-driven approaches to visual geolocalisation.

Many research efforts have been devoted to solving this problem, for prospective applications in robot vision [3], landmark identification [4], as well as city reconstruction [5]. In this article, we confine ourselves to a more defined problem: Given a query image, can we derive a location estimate of the query, based on the visual content of the query and reference images in a geo-tagged database?

But what makes visual geolocalisation difficult? First of all, the diversity of image content. A generic image of plants, animals, or the sky does not contain much geographical information, thus making it hard for an accurate location estimate. However, a landmark photo of the Eiffel Tower would make it easier for location estimate. Research work has contributed to the progress in geolocalisation of social network images [2, 6, 7], and street views [4, 8, 9] in particular. Second, the geographical distribution of reference images. If there are sufficient reference images near or at the query location, it is more likely to geolocalise the query based on the visual information of reference images. Finally, the overlapping between the visual content of a query and reference images. At first sight, the problem of visual geolocalisation can be formulated as an “object recognition” problem. However, this formulation makes more sense when there are reference images that contain objects shown in the query image, or more ideally, duplicates or near-duplicates. In street view databases such as [4, 9], it is not guaranteed that there are overlapping objects between the visual content of a...
query and reference images. Moreover, we note that it would not be appropriate to model the problem as a “scene classification” problem either. Conventionally, this involves defining a number of scene categories and classifying a query image into one of the categories based on hand-crafted features. A major problem is the unscientific choice of categories as pointed out by Hays and Efros [2]. Visual geolocalisation to street views is particularly challenging, if modelled as a classification problem. Because a city has far too many visual features and attributes, and different cities may have visual elements in common. Hence, a human or a powerful classifier would have trouble recognizing a query scene from a random part of the city, unless there are landmarks or distinctive visual elements in the query image. The recent years have seen research efforts in formulating geolocalisation as a classification problem, with the advancement in deep networks: Weyand et al. proposed PlaNet [10], where the earth is divided into multiple geographical cells, and GPS locations are mapped onto these cells. Finally, a CNN is trained to predict a GPS location from the visual content of images. By enhancing the PlaNet [10] setup, Muller-Budack et al. [11] introduced earth partitions with different levels of granularity, and scene classification into the model.

The aim of writing this article is proposing an approach to location estimation of outdoor urban images, when equipped with a reference database of location-aware street view images. It is a challenging task due to the fact that the query and reference images are captured in different lighting conditions, at different times of the day, and from different viewing perspectives. The contribution of our approach is, a new way in performing post-retrieval location estimation, by re-ranking the retrieved candidates based on the visual content of each candidate, as well as its neighbouring images. In this way, the visual information of a candidate location is enriched, which is previously represented only by the candidate image itself.

To evaluate the location estimation performance of our approach, we adopt three publicly available datasets, the first one is the UCF 2017 Street View Dataset of over 150,000 reference images, contributed by Zemene et al. [9]. The second one is a large-scale dataset of over 1.06 million reference street views from San Francisco shared by Chen et al. in 2011 [4]. Finally, the third Dataset is DeepGeo Dataset (2018), which consist of 0.5 million reference images, as well as 100,000 queries, from 50 states in the United States.

We evaluate our approach against three baseline approaches on UCF 2017 and San Francisco Dataset, and our approach achieves the best accuracy on both datasets. In addition, we also investigate if the use of geographical neighbourhoods contributes to yielding better location accuracy on UCF 2017 Dataset. Moreover, the experiments are also carried out to evaluate the estimation performance, when using different deep architectures in feature extraction, on both UCF 2017 and DeepGeo Dataset. Finally, we compare our approach to the classic 1-NN by Hays and Efros [2] in terms of estimation performance, again on UCF 2017 and DeepGeo Dataset. Our approach outperforms 1-NN on UCF 2017 Dataset, while 1-NN performs better on DeepGeo Dataset. We look into a query image from each dataset, and visualize its visual candidates returned by image retrieval, along with one of the geo-neighbourhoods. It is observed that, when faced with datasets of generic images without distinct visual elements (such as buildings), the geo-neighbourhoods generated by our approach, are likely to struggle in providing useful information for a GPS location represented by a candidate of the query image.

The remainder of the article is structured as follows: Section 2 breaks down the geolocalisation problem into two sub-problems, namely, image retrieval and post-retrieval location estimation. In addition, related approaches in the existing literature have been reviewed. Section 3 provides a step-by-step introduction to the proposed approach. Section 4 introduces the datasets, evaluation metric, and baseline approaches used in our experiments. Section 5 demonstrates the experimental results we have obtained on three datasets, and includes three discussions, including (1) whether using deep features helps improve location estimation accuracy, as well as the effectiveness of geographical neighbourhoods; (2) the difference in estimation accuracy when using different deep architectures for feature extraction; (3) investigating if our approach outperforms 1-NN on DeepGeo Dataset, along with the visualization of difference in visual elements of images, between UCF 2017 and DeepGeo Dataset. Eventually, we conclude the article in Section 6.

2 GEOLOCALISATION BY IMAGE RETRIEVAL

Geographical location and visual content are two different types of information about images. The visual geolocalisation problem, in essence, is to estimate the geographical location of a query image, given the visual content of a geo-tagged database. Among approaches with an image retrieval component, such as[2, 6, 7], an underlying assumption is, when two images are visually similar, their geographical locations are likely to be in the proximity of each other. In order to solve the geolocalisation problem by image retrieval, it is necessary to tackle two sub-problems as follows:

- **Image retrieval**: When it comes to image retrieval, two things must be taken into consideration: the first one is, which descriptors to use, in order to describe the visual content of images. While the second one is, which distance metric to adopt, when computing the distance between the descriptors of two images. Additionally, the visual dissimilarity between images is usually represented by the distance between image descriptors. In Section 2.1, we review retrieval using two different image features: hand-crafted features (including global and local descriptors) and deep features.

- **Post-retrieval location estimation**: After image retrieval, one remaining problem is how to derive a location estimate for the query, based on geographical location of the retrieved image(s). Existing location estimation approaches can be assigned into one of the two categories, i.e., direct location estimation and re-ranking.
2.1 Image retrieval

2.1.1 Retrieval using hand-crafted features

Hand-crafted features can be broadly divided into two categories: global and local features. Examples of location estimation approaches that rely on global features (such as GIST descriptors) for retrieval can be found in [2, 7, 12]. While those dependent on local features (e.g., SURF) are [6, 7, 9, 13].

The pioneering approach is “im2gps” by Hays and Efros [2], where they considered two situations in location estimation. One is when there is only one retrieved image (or known as the “first-nearest neighbour”) returned by image retrieval. Hence, they propagated the GPS coordinates of the first-nearest neighbour directly to the query image, as its location estimate; The other, however, is a more challenging situation, when there are multiple nearest neighbours returned by image retrieval. Hays and Efros therefore proposed using mean-shift clustering to tackle this situation [2]. The nearest neighbours are assigned into clusters based on their GPS coordinates. Finally, the centroid location of the largest cluster will be adopted as the location estimate of the query image. It is noted that the mean-shift clustering approach requires tuning the bandwidth parameter for clustering when using a different dataset.

Li et al. proposed and matured the “geo-visual ranking” (GVR) approach in [6, 7]. They used SURF features in [6], while GIST, SURF features individually for image retrieval in the succeeding work [7]. Here we review [7] in more detail, since it is an improved version of its precedent [6]. The GVR approach builds on the idea of “im2gps” [2], while adding an additional ranking step to select the best match among the nearest neighbours. The ranking approach first generates clusters, by performing mean-shift clustering on nearest neighbours, based on their GPS coordinates. It then proceeds to compute aggregate similarity of each cluster to the query image, where the aggregate similarity is represented as 2-norm distance between GIST descriptors of two images. Eventually, the cluster of the highest aggregate similarity will be selected, and the centroid location of the cluster will be adopted as the location estimate. One weakness of clustering-dependent approaches such as [2, 7] is that the size of the cluster plays an important role, where in some cases, a smaller cluster is likely to preserve a better location estimate for the query image.

Huang and Lo proposed a neighbourhood-based approach in [12]. Their approach does not rely on clustering, which avoids the process of tuning the bandwidth parameter for a specific dataset. Huang’s approach examines visual candidates returned by image retrieval, based on the average similarity of the candidate and its nearby images in the “geo-neighbourhood”, to the query image.

2.1.2 Retrieval using deep features

NetVLAD is presented in [14] as an approach to visual place recognition, which serves as the main component of a convolutional neural network (CNN) architecture. Essentially, NetVLAD is a generalized layer, originally inspired by “Vector of Locally Aggregated Descriptors” [22], which are adopted widely in image representation. In addition, the NetVLAD layer can be plugged into any CNN architecture.

Vo et al. [15] performed two tasks related to location estimation, one is estimation by classification, where a city label was predicted for each query image, using features extracted by a CNN and Support Vector Machine (SVM). The location estimate of the query was thus the location of the predicted city; The other task is estimation by retrieval, where CNN features were adopted for image representation in retrieval, and the location estimate was further made based on the GPS coordinates of the retrieval results. It should be noted the neural network model requires tuning for a specific location estimation task.

2.2 Post-retrieval location estimation

After image retrieval, a set of retrieved images and their corresponding geographical locations are returned. It is necessary to decide, how to derive a location estimation, given such information.

2.2.1 Direct location estimation

As the name suggests, direct location estimation is to directly derive a location estimate, based on the location of retrieved images. It only deals with the geographical information (such as latitude and longitude) in the process of location estimation. One of the straightforward direct estimation approaches, is explicitly adopting the location of the best match from image retrieval, which is considered as the most similar image to the query image. A more complex approach was first proposed in [2], where mean-shift clustering was used to cluster a set of geographical locations. The centroid location of the largest cluster, will be adopted as the query location estimate.

2.2.2 Re-ranking

While re-ranking approaches further rank the list of image candidates, based on the visual content of the candidates, but in a different way from what has been used in image retrieval. For example, the approach [7] re-ranks the candidates, by grouping them into different clusters at first, and evaluating the visual similarity of a cluster to the query image. The location estimate of the query is eventually derived from the location of images belonging to the cluster, which is of the highest aggregate similarity to the query image.

2.3 Motivation of our approach

Image retrieval does well in retrieving duplicates or near-duplicates of the query images (such as, the retrieval of landmark images), but has difficulty in retrieving reasonable
candidates, when the query images do not have corresponding (near-)duplicates in the database. In our evaluation databases, the query images and reference database are from different sources, and the query images are neither landmarks, or guaranteed to have (near-)duplicates in the reference database.

The idea behind our approach is that, using image retrieval to first narrow down the list of candidate locations, by retrieving candidate images. The proposed approach then re-ranks the candidate images on the list, based on the visual content of a candidate’s geographical neighbourhood. A geographical neighbourhood consists of, the candidate itself and a few other images from the reference database, whose location are in the proximity of the candidate location. Since each candidate image represents a geolocation, its neighbouring images are likely to provide additional visual information about the candidate location. The street view databases adopted in our evaluation, provide at least two images at a location, the neighbouring images in fact come from the same location as that of the candidate image. We call this set of images, which contains a candidate image and its neighbouring images (or “neighbours”), the candidate’s neighbourhood. In Section 3, the framework of our approach will be introduced in details.

3 | LOCATION ESTIMATION OF OUTDOOR URBAN IMAGES

We approach the problem in a four-step manner. The four steps are visual candidate retrieval, geo-neighbourhood construction, scene matching, and query location estimation. The framework of our approach is illustrated in Figure 1.

- **Step 1: Visual candidate retrieval** (Section 3.1): We retrieve from the reference database, \( k \) most similar images to the query \( q \), where we refer to these retrieved images as “visual candidates” of the query. The purpose of this step is narrowing down the search list of the most likely GPS locations for the query.
• Step 2: Geo-neighbourhood construction (Section 3.2): Each visual candidate forms its own geographical neighbourhood, which consists of all the reference images, whose locations are the closest to that of a visual candidate in terms of Haversine distance. The rationale is that, since each visual candidate has a geographical location, it would be helpful if we collect more visual information from, or near the candidate location.

• Step 3: Scene matching (Section 3.3): We perform scene matching via Deformable Spatial Pyramid (DSP) \cite{19} on all the candidate neighbourhoods. The purpose of this step is to compute the average match distance for each neighbourhood, so that we can further determine which candidate location is the best match for the query location.

• Step 4: Query location estimation (Section 3.4): Finally, we propagate the GPS coordinates of the visual candidate, which has the smallest average match distance obtained in Step 3, to the query image as its location estimate.

### 3.1 Visual candidate retrieval

When given a query image \( q \), the purpose of this step is to retrieve from the reference database \( R \), candidate images (or “visual candidates”) that are most visually similar to the query. An underlying assumption is that visually similar images are more likely to be in the proximity of one another geographically, due to having visual elements in common. Hence, by retrieving visual candidates for the query, we narrow down the list of candidate locations for the query.

We compute the visual distance, denoted \( G(q, r) \), between a reference image \( r \) and the query image \( q \), using 2-norm distance (L2 distance) as defined follows. \( k \) reference images of the shortest visual distance will be included in the final visual candidate set \( C(q) \).

\[
G(q, r) = \| F(r) - F(q) \|_2^2
\]

where \( F(\cdot) \) refer to the feature vector of an image. We adopt GIST descriptors \cite{16, 17} for the purpose of image representation in image retrieval, since GIST descriptors are easy to compute and serve as a compact representation of an image.

### 3.2 Geo-neighbourhood construction

After retrieving \( k \) visual candidates of the query image, the aim of the second step is to collect more visual evidence for each location represented by a candidate image \( \alpha \). However, an image often does not provide sufficient information about a specific location. Hence, we propose to include geographical neighbours of a candidate in order to visually represent a location more vividly. We accomplish this by creating a geographical neighbourhood \( N_\alpha \) for each candidate image \( \alpha \), where we retrieve from the rest of the database, the images whose locations are the closest to the candidate location in terms of Haversine distance between two locations. It is noted that the number of images in the neighbourhood, denoted \( \phi_\alpha \), varies among candidates.

The Haversine distance \( H(a, b) \) between two location pairs, for instance, \( a = (\text{lat}_a, \text{lon}_a) \) and \( b = (\text{lat}_b, \text{lon}_b) \), can be computed using the Haversine formula \cite{18} as shown in Algorithm 1.

### 3.3 Scene matching

We perform scene matching via DSP matching \cite{19} on the query and each image in the neighbourhood, and compute the average match distance for each neighbourhood. The purpose of this step is to evaluate the visual candidates obtained in Step 1 (“visual candidate retrieval”), and pave the way for selecting the best match among the visual candidates.

The DSP matching method is an efficient yet accurate dense matching algorithm which has its advantage in matching between images of different point of views. By using the DSP algorithm, we can investigate if the retrieved images obtained based on GIST descriptors happen to have the scene depicted in the query image, possibly from a different point of view.

The DSP approach first builds a spatial pyramid, by dividing the entire image into four rectangular grid cells, and continue dividing until reaching the desirable number of pyramid levels. Then the spatial pyramid is represented with a graph, where each grid cell and pixel is a node. The matching cost is computed for each node by using local descriptors extracted from the images. The goal is to find the best translation for each node from the first to the second image, by minimizing the energy function as defined in Equation (2) \cite{19}.

\[
E(t) = \sum_i D_i(t_i) + \epsilon \sum_{i,j \in \mathcal{E}} V_{ij}(t_i, t_j),
\]
The data term $D_t$ and the smoothness term $V_{ij}$ are defined in Equations (3) and (4), respectively.

$$D_t(t_i) = \sum_q \min \left( \left\| F_t(q) - F_t(q + t_i) \right\|_1, \lambda \right), \quad (3)$$

$$V_{ij}(t_i, t_j) = \min \left( \left\| t_i - t_j \right\|_1, \gamma \right), \quad (4)$$

where $t_i = (u_i, v_i)$ is the translation of node $i$ from the first to the second image, $\omega$ denotes pairs of nodes linked by graph edges, $q$ denotes pixel coordinates within a node $j$ where the local descriptors were extracted, $F_1$ and $F_2$ are the descriptors extracted at location $q$ and $q + t_i$, $\lambda$ and $\gamma$ are the thresholds of the truncated L1 norm for robustness to outliers.

We define the matching cost between an image $m$ in the geo-neighbourhood $N_\alpha$ ($m \in N_\alpha$) and the query $q$ in Equation (5).

$$\text{Match}(m, q) = \min_t E(t; m, q), \quad (5)$$

where $E(t; m, q)$ denotes the energy function by finding the optimal translation of each node in the $m$ to the query image $q$. A smaller match cost Match$(m, q)$ indicates two images are a better match. Instead of using GIST descriptors for image retrieval, we adopt the SIFT descriptors in DSP matching, in order to perform scene matching between images in the neighbourhood $N_\alpha$ and the query $q$. In our experiments, we set $\epsilon = 0.005$ in Equation (2), $\lambda = 500$ in Equation (3), and $\gamma = 0.25$ in Equation (4), following the parameter settings in [19].

### 3.4 Query location estimation

For each neighbourhood $N_\alpha$, we compute the average DSP matching cost between all neighbourhood images and the query, to evaluate the visual similarity of the neighbourhood $N_\alpha$ to the query $q$, as defined in Equation (6).

$$\text{AvgMatch}(N_\alpha) = \frac{\sum_{m \in N_\alpha} \text{Match}(m, q)}{\Phi_\alpha}, \quad (6)$$

where $m_i$ is an image in the neighbourhood $N_\alpha$. The candidate $\alpha$ with the smallest average DSP matching cost will be considered the best match for the query image, as expressed in Equation (7).

$$\alpha^* = \arg\min_{\alpha \in R} \text{AvgMatch}(N_\alpha). \quad (7)$$

The location of $\alpha^*$, denoted $(\text{lat}_{\alpha^*}, \text{lon}_{\alpha^*})$, will be adopted as the query location estimate, denoted $\text{loc}_{\alpha^*}$. The framework of our proposed approach is further described using the pseudocode in Algorithm 2.

#### ALGORITHM 1 Our approach to location estimation

**Input:** a query image $q$, a geo-tagged reference database $R$, the number of best candidates $k$  

**Output:** the estimated location of the query, $\text{loc}_{\alpha^*}$  

1: **Step 1: Visual candidate retrieval**  
2: for each image $r$ in the reference database $R$ do  
3: \hspace{1em} $G(q, r) \leftarrow$ GIST visual distance computed using Equation (1)  
4: \hspace{1em} $C(q) \leftarrow$ visual candidates in $R$ of the top-$k$ shortest GIST visual distance $G(q, r)$  
5: **Step 2: Geo-neighbourhood construction**  
6: for each candidate $\alpha$ in $C(q)$ do  
7: \hspace{1em} $N_\alpha \leftarrow$ the set of candidate $\alpha$ and the closest images in $R$ based on the Haversine distance $H$ computed using Algorithm 1  
8: \hspace{1em} $\Phi_\alpha \leftarrow$ the number of images in the neighbourhood $N_\alpha$  
9: **Step 3: Scene matching**  
10: for each image $m_i$ in the neighbourhood $N_\alpha$ do  
11: \hspace{1em} $\text{Match}(m_i, q) \leftarrow$ Matching cost via DSP matching  
12: **Step 4: Query location estimation**  
13: \hspace{1em} $\text{AvgMatch}(N_\alpha) \leftarrow \frac{\sum_{m \in N_\alpha} \text{Match}(m, q)}{\Phi_\alpha}$  
14: \hspace{1em} $\alpha^* \leftarrow \arg\min_{\alpha \in C(q)} \text{AvgMatch}(N_\alpha)$ (the candidate image $\alpha$ with the smallest average matching cost)  
15: \hspace{1em} $\text{loc}_{\alpha^*} \leftarrow (\text{lat}_{\alpha^*}, \text{lon}_{\alpha^*})$  
16: return $\text{loc}_{\alpha^*}$

### 4 EXPERIMENTAL SETUP

#### 4.1 Datasets and evaluation metric

In order to evaluate the performance of our proposed approach, we use two street view datasets in our experiments. The first dataset, UCF 2017 Street View Dataset, released by Zemene et al. [9], comprises 152,536 reference street views of 14 cities from Europe, U.S.A., and Australia. The 259 queries, on the other hand, come from 13 of the reference cities. Figures 2 and 3 demonstrate a number of reference images and queries from the UCF 2017 Dataset, respectively. In addition, Table 1 shows the distribution of the images in 13 reference cities. The second dataset, San Francisco Landmark Dataset, was made public by Chen et al. in 2011 [4]. This dataset is made up of 1,062,468 street view images, along with 803 queries from downtown San Francisco. One-hundred reference images, as well as 49 query samples are shown in Figures 4 and 5, respectively.

For evaluation purpose, we define a distance threshold, denoted $\Theta$, which controls the evaluation accuracy and tolerance to noise in ground truth GPS coordinates as adopted in [7]. A query image is considered to be geolocalised correctly, if its estimated coordinates, denoted $\text{loc}_{\alpha^*}$, fall within the radius $\Theta$, of the ground truth location $\text{loc}_q$. As formally expressed in Equation (8), the correctness of geolocalising an image can be obtained using eval$_H$($\text{loc}_{\alpha^*}$, $\text{loc}_q$), with respect to the distance threshold $\Theta$, where $H$($\text{loc}_{\alpha^*}$, $\text{loc}_q$) refers to the Haversine
distance between the estimated location $\text{loc}_{q^*,a}$, and the ground truth location $\text{loc}_q$ of the query image $q$, which can be computed using Algorithm 1.

$$\text{eval}_s(\text{loc}_{q^*,a}, \text{loc}_q) = \begin{cases} \text{correct}, & H(\text{loc}_{q^*,a}, \text{loc}_q) \leq \theta \\ \text{incorrect}, & \text{otherwise} \end{cases}.$$  \hspace{1cm} (8)

**TABLE 1** Distribution of reference and query images from the UCF 2017 dataset [9]

| Continent/Country | City       | # of Reference Images | # of Query Images |
|-------------------|------------|-----------------------|-------------------|
| Europe (5 cities) | Amsterdam  | 15,608                | 17                |
|                   | Frankfurt  | 8,788                 | 0                 |
|                   | Rome       | 15,729                | 13                |
|                   | Milan      | 15,986                | 2                 |
|                   | Paris      | 16,588                | 28                |
| Australia (2 cities) | Sydney    | 15,986                | 16                |
|                   | Melbourne  | 14,190                | 10                |
| USA (7 cities)    | Vegas      | 5,950                 | 54                |
|                   | Los Angeles| 12,610                | 13                |
|                   | Phoenix    | 4,233                 | 13                |
|                   | Houston    | 10,361                | 11                |
|                   | San Diego  | 3,079                 | 15                |
|                   | Dallas     | 4,740                 | 18                |
|                   | Chicago    | 8,688                 | 49                |
| Total number      |            | 152,536               | 259               |

**FIGURE 4** One-hundred reference images from the San Francisco Landmark Dataset [4]

### 4.2 Baseline approaches

The location estimation performance of our approach is compared against three baseline approaches, which are all retrieval-based approaches that rely on a preceding procedure of image retrieval in order to extract candidate images for a query, then make the location estimate of the query image based on the location of retrieved images.
4.2.1 First-nearest neighbour (1-NN)

As proposed in [2], the first-nearest neighbour of the query image is retrieved from the reference database, and the GPS coordinates of the 1-NN are adopted as the query location estimate. The first-nearest neighbour is defined as the reference image with the shortest visual distance (or highest visual similarity) to the query image. In our experiments, we adopt GIST descriptors in image representation and 2-norm distance in measuring the visual distance between two feature descriptors.

4.2.2 Clustering

Since the 1-NN approach is often not robust, Hays and Efros proposed using a larger set of nearest-neighbours in [2]. \( k \)-nearest neighbours (\( k > 1 \)) of a query are therefore retrieved, and grouped into different clusters using mean-shift clustering based on their GPS coordinates. Eventually, the mode location of the largest cluster is adopted as the location estimate for the query. In the experiments, we retrieve 20 nearest neighbours for each query following [7]. In terms of mean-shift clustering, we use the bandwidth of 0.2 for the UCF 2017 Dataset, and 0.02 for the San Francisco Landmark Dataset, although other bandwidths work similarly. The mode location is defined as the average location of all images in the largest cluster.

4.2.3 Ranking

The “geo-visual ranking” approach [7] used as a baseline in our experiments, is an extended version of the one proposed in [6]. Similar to the clustering approach [2], \( k \)-nearest neighbours (\( k > 1 \)) are first retrieved for the query, then assigned into different clusters on the basis of their GPS coordinates. However, only the top \( m \) neighbours and the corresponding clusters (\( m < k \)), will enter the location estimation process. Each cluster will be given a ranking score, which is the aggregate visual similarity of images in the cluster to the query. The cluster with the highest ranking score will be selected, and the GPS coordinates of the cluster centroid will be the query location estimate. Regarding the ranking approach [7], we set the number of nearest neighbours \( k \) to 20 during image retrieval, and the number of clusters \( m \) eligible for the location estimation process to 10, following the optimal parameter combination in [7].

5 RESULTS AND DISCUSSION

5.1 Location estimation results

Table 2 shows the location estimation accuracy obtained on the UCF 2017 Dataset by our approach and three baselines. Our approach outperforms three baselines across all distance thresholds in terms of estimation accuracy. Among three baselines, the ranking approach achieves the best accuracy in three distance thresholds. We follow [2] and adopt 25, 200, 750, and 2500 kilometres as distance thresholds since the UCF 2017 Dataset is a global-scale dataset.

Table 3 demonstrates the location estimation accuracy on the San Francisco Landmark Dataset, where our approach again obtains the best estimation accuracy across all distance thresholds. The 1-NN and ranking approach are neck-to-neck, obtaining the best accuracy in three distance thresholds. We follow [20] and choose smaller distance thresholds (50, 100, 200, and 300 m) for the San Francisco Dataset since it is city-scale dataset.

| TABLE 2 | Location estimation accuracy on the UCF 2017 Dataset [9] |
| --- | --- | --- | --- | --- | --- |
| Distance threshold (km) | 25 | 200 | 750 | 2500 |
| 1-NN [2] | 8.49% | 9.27% | 15.06% | 27.03% |
| Clustering [2] | 6.18% | 6.95% | 15.44% | 27.41% |
| Ranking [7] | 8.88% | 8.88% | 20.08% | 35.91% |
| Ours | 10.42% | 10.81% | 20.85% | 38.22% |

| TABLE 3 | Location estimation accuracy on the San Francisco Landmark Dataset [4] |
| --- | --- | --- | --- |
| Distance threshold (m) | 50 | 100 | 200 |
| 1-NN [2] | 5.85% | 8.22% | 11.28% |
| Clustering [2] | 0.62% | 2.12% | 7.47% |
| Ranking [7] | 5.85% | 8.22% | 11.33% |
| Ours | 6.10% | 9.22% | 12.45% | 16.81% |
5.2 Discussion 1: Does using deep features help improve estimation accuracy

The location estimation results presented in Section 5 were obtained using GIST descriptors for image retrieval. The recent years have seen a wide adoption of deep features in image recognition. Deep features, are features learned from images, using deep neural networks. We have evaluated our approach, using features from two different convolutional networks, namely, VGG-19 [21] and VGG-16 NetVLAD features [14], which have been trained for image recognition tasks. In order to evaluate the effectiveness of geographical neighbourhoods, while using each type of deep feature, two groups of experiments have been conducted:

- **Without geographical neighbourhoods.**
  After image retrieval, the candidates are re-ranked by their visual similarity to the query. Since the approach adopts local descriptors in this stage, the dissimilarity is therefore defined as, the DSP matching cost (discussed in Section 3.3) between the query and a candidate image: The higher the matching cost, the less similar two images are.

- **Using geographical neighbourhoods.**
  When using geographical neighbourhoods, after image retrieval, each candidate forms its own geographical neighbourhood. The candidates are now represented by their geographical neighbourhoods. The average DSP matching cost is computed for each geographical neighbourhood. The candidate, whose neighbourhood has the smallest average DSP matching cost, will be considered as the best match.

### 5.2.1 Using VGG-19 features

Regarding VGG-19 features, two situations have been considered:

- **Considering only global information.**
  For each image, the outputs of the last fully connected layer of the VGG-19 Net, are adopted as image features for the purpose of image retrieval. These features are high-level features, and also known as “global features”.

- **Considering both the global and local information.**
  Since the outputs of the last fully connected layer of VGG-19 Net, are global features, which are usually considered as “coarse features”. Hence, it makes sense to evaluate if the inclusion of local features would improve the retrieval results.

Regarding VGG-19 Net, outputs of the last fully connected layer is a 1000-dimensional vector. We find that at each convolutional layer (here we have 16 convolutional layers), there are depths (number of channels of feature maps), together with the convolutional kernel, for example, 3 x 3, with a stride. Thus, for example, in the first convolutional layer, we have a convolutional kernel of 3 x 3, and a depth of 64, which means that there are 64 feature maps related to the first convolutional layer. The question is: how do we obtain a local feature? One straightforward way, is to condense each feature map to only 1 value using what is known as global average pooling mechanism. In other words, in the first layer, the 64 feature maps are condensed to a 64-dimensional vector. Using this mechanism, we have, for the VGG-19 Net, the following local feature vectors:

\[
2 \times 64 + 2 \times 128 + 4 \times 256 + 4 \times 512 + 4 \times 512 \\
= 128 + 256 + 1024 + 4096 = 5504. \quad (9)
\]

In essence, we simply concatenate the local features together to form this long vector. So, the feature vector length will be a local + global = 5504 + 1000 = 6504-dimensional vector.

Table 4 shows that both types of VGG-19 features achieve better performance, when geographical neighbourhoods are taken into consideration, especially in terms of accurate geolocalisation (evaluating using smaller distance thresholds). Considering thresholds of 750 and 2,500 km, VGG-19 features with geographical neighbourhoods, also yield promising accuracy, especially with the 2,500 km distance threshold.

### 5.2.2 Using NetVLAD features for image retrieval

NetVLAD [14] is inspired by VLAD (Vector of Locally Aggregated Descriptors) [22], which is a way of image representation adopted in image retrieval. In [14], NetVLAD is essentially a main component in their CNN architecture, and trained for the “visual place recognition” task, which is to recognize the location of a given query photo. It makes sense to adopt NetVLAD image presentation in image retrieval, since it is one of the state-of-the-arts. Hence, an evaluation has been conducted, to find out if the geographical neighbourhoods help improve the location estimation accuracy, after adopting VGG-16 NetVLAD features in image retrieval.

As shown above in Table 4, in general, using CNN features for image retrieval, helps yield better location estimation accuracy, especially with geographical neighbourhoods.
accuracy. While in the case of using VGG-16 NetVLAD features for image retrieval, the geographical neighbourhoods do not help improve the estimation accuracy.

5.2.3 Discussion summary

Coming back to the question itself: does using deep features help improve location estimation accuracy? From Table 4 and Figure 6, the following observations can be made:

- Compared to using GIST descriptors (a type of non-deep features), the location accuracy improves when VGG-19 features are adopted. However, a slight decrease is also witnessed while using VGG-16 NetVLAD features;
- The effectiveness of geographical neighbourhoods. Regarding VGG-19 Net features in both settings, a mild increase in accuracy can be observed, with smaller distance thresholds (25 km and 200 km). The most significant increase, however, can be yielded with the largest distance threshold (2,500 km). In addition, when using VGG-16 NetVLAD features, a decrease is shown across three out of four distance thresholds.

5.3 Discussion 2: Location estimation accuracy when using different deep architectures

When it comes to feature extraction using deep networks, two names cannot go unmentioned: VGGNet [21, 23] and ResNet [24]. In [21], it is shown that classification error decreases with the increased ConvNet depth, that is, VGG-19 yields better classification accuracy than VGG-16, due to having more weighted layers. Hence, in the experiments shown in Section 5.2, we adopt VGG-19 and VGG-16 NetVLAD as two representative architectures, and demonstrate how the choice of deep architecture in feature extraction affects the estimation accuracy.

In this section, our approach is evaluated on VGGNet and ResNet architecture. For VGGNet, we consider VGG-S/M/F [23] and VGG-16 [21]; in terms of ResNet, three variants of ResNet are taken into consideration, namely, ResNet-50, ResNet-101, as well as ResNet-152. We also cover two situations where using VGG-16 architecture could result in feature vectors of different size: (1) a 5224-dimensional feature vector for each image, including both local and global information; (2) a 1000-dimensional feature vector for an individual image, depicting only global information.

Table 5 shows the location estimation accuracy obtained on the UCF 2017 Dataset [9], using different deep architectures.

| Distance threshold (km) | VGG-S | VGG-M | VGG-F | VGG-16 (5224-d, local+global) | VGG-16 (1000-d, global) | VGG-19 (1000-d, global) | ResNet-50 | ResNet-101 | ResNet-152 |
|------------------------|-------|-------|-------|-----------------------------|------------------------|------------------------|-----------|------------|-----------|
| 25                     | 18.53%| 15.06%| 18.15%| 7.34%                       | 19.69%                 | 13.13%                 | 6.95%     | 7.72%      | 11.20%    |
| 200                    | 18.53%| 15.06%| 18.15%| 7.34%                       | 19.69%                 | 13.13%                 | 6.95%     | 7.72%      | 11.20%    |
| 750                    | 28.57%| 25.10%| 32.05%| 23.17%                      | 32.82%                 | 29.73%                 | 17.37%    | 17.37%     | 22.01%    |
| 2,500                  | 54.83%| 52.51%| 58.30%| 25.91%                      | 57.14%                 | 55.21%                 | 39.38%    | 40.54%     | 44.02%    |

A more recent, country-scale Google Street View (GSV) Dataset is presented in [25] for image classification purpose, as “the first large-scale, open source GSV dataset of the United States”. Here, we address the dataset [25], as the DeepGeo Dataset.

The DeepGeo Dataset consists of two parts: (1) The 50States10K Dataset, which is the training set, includes 10,000 unique images from each of 50 States, providing a total of 0.5 million images. Additionally, in each state, there are 2,500 sample locations, and each sample location has 4 Street View images.
images. (2) The 50States2K, used as the test set, covers 2,000 unique images from each state. In every state, there are 500 sample locations, with four images taken from each location. The test locations are selected carefully, which are different from one another, as well as the locations in 50States10K [25].

Five hundred images are randomly selected from 50States2K, as the query images in our experiments. Table 6 presents the estimation results obtained on the DeepGeo Dataset, when different architectures are used in feature extraction. Additionally, GIST features, a type of hand-crafted features, are also adopted in comparison with the features extracted using deep networks.

For the DeepGeo Dataset, VGG-16 (1000-d), ResNet-50 and ResNet-101 yield better accuracy than other architectures, in the corresponding distance threshold(s).

### Discussion 3: 1-NN versus our approach on DeepGeo Dataset (2018)

#### The location estimation results

Our approach is retrieval-based, and aims at re-ranking multiple image candidates returned by image retrieval, based on the visual information of each candidate’s neighbouring images from the reference database. The highest ranked candidate image will be selected as the best match and its location will be assigned to the query, as its location estimate. 1-NN for geolocalisation, on the other hand, first appeared in Hays and Efro’s work, IM2GPS [2]. 1-NN is a straightforward approach, that simply retrieves the best match, that is, the most visually similar image to the query.

In experimental results shown in Section 5.1, our approach yields better accuracy than 1-NN, on UCF 2017 Dataset [9] and San Francisco Landmark Dataset [4], respectively. In this section, we again evaluate the results of the classic 1-NN and our approach, obtained on the DeepGeo Dataset.

Three variants of ResNet-50 are adopted and tested in [25], which are early, medium, and late integration, for the purpose of image classification. While in our experiments, the three variants of ResNet-50 are adopted in feature extraction. We also look into two scenarios discussed in DeepGeo [25]. One is single-state accuracy, where we choose one of the top performing states, California. There are 2,000 queries along with 10,000 reference images taken in California. Additionally, each image has a distinct location (latitude, longitude) from one another. The other is country-level accuracy, where we randomly sample 500 queries, from the original query set of 100,000 images. In addition, these 500 queries come from different states.

Table 7 lists the estimation results of 2,000 California query images. By the integration method, early integration [25] achieves the best estimation accuracy among three methods; Between 1-NN and our approach, 1-NN yields better performance in all four distance thresholds.

In the original DeepGeo paper [25], the top-5 prediction accuracy (medium integration) for CA is about 48%. The prediction is considered to be correct if 1 out of 5 guesses is correct. Our approach, on the other hand, generates a location estimate, instead of only predicting a state, when given a query image. Our approach yields an accuracy of 42.95%, in the 200 km threshold, as well as an accuracy of 75.65% in the 750 km distance threshold.

When it comes to the estimation results of 500 random queries, by integration method, medium integration is the winner in all four distance thresholds, which corresponds to the observation in [25], that medium integration is the best performing method, for the purpose of state prediction. In terms of comparison between 1-NN and our approach, 1-NN yields better accuracy in three distance thresholds, while our approach outperforms 1-NN in the 750 km distance threshold. Additionally, when compared to the results previously shown in Section 6, the estimation accuracy has an insignificant increase. This is due to the fact that three integration methods are tailored for the DeepGeo Dataset [25].

#### Table 6: Location estimation results of 500 random images from the DeepGeo Dataset [25], using different architectures

| Distance threshold (km) | 25   | 200  | 750  | 2,500 |
|-------------------------|------|------|------|-------|
| GIST                    | 0.20%| 1.80%| 19.60%| 85%   |
| VGG-S                   | 0    | -4%  | 18.60%| 83.60%|
| VGG-M                   | 0    | 3.20%| 21.40%| 84.40%|
| VGG-F                   | 0    | 3.40%| 20.40%| 85.80%|
| VGG-16 (1000-d, global) | 0.40%| 3.80%| 19.60%| 89%   |
| VGG-19(1000-d, global)  | 0    | 2%   | 16.60%| 83%   |
| ResNet-50               | 0.20%| 1.40%| 22.60%| 85%   |
| ResNet-101              | 0.40%| 4.20%| 21%   | 85%   |
| ResNet-152              | 0    | 2.20%| 21.60%| 84%   |

#### Table 7: 1-NN versus our approach: Location estimation accuracy of 2,000 California query images from the DeepGeo Dataset [25]

| Distance threshold (km) | 25   | 200  | 750  | 2,500 |
|-------------------------|------|------|------|-------|
| 1-NN (late integration [25]) | 0.60%| 47.60%| 84% | 98.60%|
| Ours (late integration [25]) | 0    | 4%   | 85.40%| 99.80%|
| 1-NN (medium integration [25]) | 0.80%| 48.80%| 90% | 100% |
| Ours (medium integration [25]) | 0    | 49.60%| 88.80%| 99.60%|
| 1-NN (early integration [25]) | 0.60%| 48%   | 85%  | 99.60%|
| Ours (early integration [25]) | 0    | 48.20%| 84%  | 99.60%|

#### Table 8: 1-NN versus our approach: Location estimation accuracy of 500 random queries from the DeepGeo Dataset [25]

| Distance threshold (km) | 25   | 200  | 750  | 2,500 |
|-------------------------|------|------|------|-------|
| 1-NN (late integration [25]) | 0.60%| 47.60%| 84% | 98.60%|
| Ours (late integration [25]) | 0    | 4%   | 85.40%| 99.80%|
| 1-NN (medium integration [25]) | 0.80%| 48.80%| 90% | 100% |
| Ours (medium integration [25]) | 0    | 49.60%| 88.80%| 99.60%|
| 1-NN (early integration [25]) | 0.60%| 48%   | 85%  | 99.60%|
| Ours (early integration [25]) | 0    | 48.20%| 84%  | 99.60%|
5.4.2 Visualizing visual candidates and a candidate’s neighbourhood

It defies the expectation, that our approach does not yield better accuracy than 1-NN, in terms of estimation accuracy shown in Tables 7 and 8. And it sparks our interest in investigating why our approach does not meet the expectation.

We start with visualizing the candidates of a query image, along with one of the neighbourhoods (i.e. 1-NN’s neighbourhood), coming from UCF 2017 Dataset [9], and DeepGeo Dataset [25], respectively. UCF 2017 Dataset is made up of images depicting street scenes or building façades. When given a query image, the 1-NN approach [2] returns a candidate, that is considered as the most visually similar to the query. What our approach does is, essentially, to extract multiple candidate images (instead of only one candidate, like 1-NN does). And for each candidate, it creates a neighbourhood made up of nearby images. These neighbourhoods introduce additional visual information about the GPS location represented by a candidate image.

Figure 7 illustrates, for a given query, multiple image candidates and one of the neighbourhoods. The neighbourhood provides additional visual information regarding the GPS location represented by the candidate image. However, in the DeepGeo example (as seen in Figure 8), it becomes more challenging to geolocalise a query image. On one hand, the images are generic: the visual elements include roads, skies, trees etc. They do not provide much useful information about a specific location in the first place; On the other hand, the introduced geo-neighbours do not include distinct visual elements either. Hence, our approach is likely to struggle in coming up with an accurate location estimate, when given a generic query image.

During post-retrieval location estimation, instead of having only one retrieved image represent a geographical location, we allow its neighbouring images to provide more visual information for a candidate location. A retrieved image and its neighbouring images form a so-called “geographical neighbourhood”. Afterwards, we re-rank the list of candidate images returned by image retrieval, based on the visual information of all the images in a neighbourhood.

In terms of evaluation, we adopt two public datasets: one is a global-scale dataset, UCF 2017 Dataset [9], while the other is a city-scale one, San Francisco Landmark Dataset [4]. In addition, we also use the UCF 2017 Dataset, to investigate if using deep features (such as VGG-19 Net and VGG-16 NetVLAD) helps improve location estimation accuracy; as well as whether geographical neighbourhoods contribute to yielding better estimation accuracy.

To further evaluate the performance of our approach, we adopt a more recent Google Street View dataset, DeepGeo Dataset [25], where we compare the accuracy when using different deep architectures (including VGGNet and ResNet) in feature extraction. Moreover, we evaluate the estimation performance between the classic 1-NN and our approach, in terms of single-state and country-level accuracy, using different subsets of the original DeepGeo query set. Finally, we investigate why our approach outperforms 1-NN on UCF 2017 Dataset, while the results show the opposite on DeepGeo Dataset. Through visualizing the visual candidates of a query image, and one of the geo-neighbourhoods, it is observed that DeepGeo Dataset consists of more generic images, where the geographical neighbourhoods generated by our approach, struggle in providing useful information for the GPS location represented by a candidate image.

One possible direction for the future work is developing a one-versus-many matching approach to computing the similarity between the query and multiple images at a candidate location. In our approach, the similarity is computed between two images at a time. Hence, it is worth investigating if a change in the way of computing the similarity between the query and multiple reference images would help improve the location estimation accuracy.

6 CONCLUSION

We approach the problem of visual geolocalisation, in a four-step manner. Essentially, we break down the problem into two sub-problems: image retrieval and post-retrieval estimation.
ACKNOWLEDGEMENTS
The authors would like to thank Dr. Sio-Long Lo and Professor Ah Chung Tsoi for their comments on earlier versions of the manuscript.

REFERENCES
1. Bar, M.: The proactive brain: using analogies and associations to generate predictions. Trends in cognitive sciences 11(7), 280–289 (2007)
2. Hays, J., Efros, A.A.: Im2gps: estimating geographic information from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition, 2008, CVPR 2008, pp. 1–8. IEEE, Piscataway (2008)
3. Meier, L., et al.: Pixhawk: A system for autonomous flight using onboard computer vision. In: IEEE International Conference on Robotics and Automation, 2011, ICRA 2011, pp. 2992–2997. IEEE, Piscataway (2011)
4. Chen, D.M., et al.: Pixhawk: A system for autonomous flight using onboard computer vision. In: IEEE Conference on Computer Vision and Pattern Recognition, 2011, CVPR 2011, pp. 737–744. IEEE, Piscataway (2011)
5. Tuine, K., et al.: Photocity: training experts at large-scale image acquisition through a competitive game. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1383–1392. ACM, New York (2011)
6. Li, X., Larson, M., Hanjalic, A.: Global-scale location prediction for social images using geo-visual ranking. IEEE Trans. Multimed. 17(5), 674–686 (2015)
7. Zamir, A.R., Shah, M.: Image geo-localization based on multiple nearest neighbor feature matching using generalized graphs. IEEE Trans. Pattern Anal. Mach. Intell. 36(8), 1546–1558 (2014)
8. Zemene, E., et al.: Large-scale image geo-localization using dominant sets. arXiv preprint, arXiv:170201238, 2017
9. Weisend, T., et al.: Planet-photo geolocation with convolutional neural networks. In: European Conference on Computer Vision, pp. 37–55. Springer, Cham (2016)
10. Muller-Budack, E., et al.: Geolocation estimation of photos using a hierarchical model and scene classification. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 563–579. Springer, Cham (2018)
11. Huang, J., Lo, S.L.: Location estimation of urban images based on geographical neighborhoods. J. Phys.: Conf. Ser. 1004, 012019 (2018)
12. Zamir, A.R., Shah, M.: Accurate image localization based on google maps street view. In: European Conference on Computer Vision, pp. 255–268. Springer, New York (2010)
13. Arandjelovic, R., et al.: Netvlad: CNN architecture for weakly supervised place recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5297–5307. IEEE, Piscataway (2016)
14. Vo, N., et al.: Revisiting im2gps in the deep learning era. In: IEEE International Conference on Computer Vision, 2017, ICCV 2017, pp. 2640–2649. IEEE, Piscataway (2017)
15. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vis. 42(3), 145–175 (2001)
16. Oliva, A., Torralba, A.: Building the gist of a scene: The role of global image features in recognition. Prog. Brain Res. 155, 23–36 (2006)
17. Robusto, C.C.: The cosine-haversine formula. Am. Math. Monthly 64(1), 38–40 (1957)
18. Kim, J., et al.: Deformable spatial pyramid matching for fast dense correspondences. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2307–2314. IEEE, Piscataway (2013)
19. Yagcioglu, S., et al.: City scale image geolocalization via dense scene alignment. In: 2015 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 726–732. IEEE, Piscataway (2015)
20. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint, arXiv:14091556, 2014
21. Jegou, H., et al.: Aggregating local image descriptors into compact codes. IEEE Trans. Pattern Anal. Mach. Intell. 34(9), 1704–1716 (2011)
22. Chatfield, K., et al.: Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint, arXiv:14053531, 2014
23. He, K., et al.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778. IEEE, Piscataway (2016)
24. Suresh, S., et al.: Deepgeo: Photo localization with deep neural network. arXiv preprint, arXiv:181003077, 2018

How to cite this article: Huang J, Huang H. Looking around in the neighbourhood: Location estimation of outdoor urban images. IET Image Process. 2021;1–13. https://doi.org/10.1049/ipr2.12190