# Leveraging EfficientNet and Contrastive Learning for Accurate Global-scale Location Estimation

Giorgos Kordopatis-Zilos, Panagiotis Galopoulos, Symeon Papadopoulos, Ioannis Kompatsiaris
Information Technologies Institute, CERTH, Thessaloniki, Greece
{georgekordopatis,gpan,papadop,ikom}@iti.gr

## ABSTRACT

In this paper, we address the problem of global-scale image geolocation, proposing a mixed classification-retrieval scheme. Unlike other methods that strictly tackle the problem as a classification or retrieval task, we combine the two practices in a unified solution leveraging the advantages of each approach with two different modules. The first leverages the EfficientNet architecture to assign images to a specific geographic cell in a robust way. The second introduces a new residual architecture that is trained with contrastive learning to map input images to an embedding space that minimizes the pairwise geodesic distance of same-location images. For the final location estimation, the two modules are combined with a search-within-cell scheme, where the locations of most similar images from the predicted geographic cell are aggregated based on a spatial clustering scheme. Our approach demonstrates very competitive performance on four public datasets, achieving new state-of-the-art performance in fine granularity scales, i.e., 15.0% at 1km range on Im2GPS3k.

## CCS CONCEPTS

• **Computing methodologies** → **Computer vision problems**;
• **Information systems** → **Geographic information systems**.

## KEYWORDS

location estimation, global-scale location estimation, geolocation, contrastive learning, spacial clustering

## 1 INTRODUCTION

We define visual location estimation as the process of estimating the geographical coordinates of a scene based solely on the visual cues existing in the given image. One could think multiple variations of this task, depending on whether we restrict our input to a particular kind of scenes, e.g., landmarks [3, 5, 46], to be from a particular area [1, 17, 44], or on whether we are using different inputs, e.g., a sequence of images per scene [2, 31, 32], or aerial imagery [24, 34, 43]. In this study, we focus on global-scale location estimation from single images, which is the most challenging problem setting. Without restricting the type and location of the input images, some ambiguity is unavoidable, as not all images contain

enough visual cues to allow their precise localization. This ambiguity raises the difficulty of the task significantly and highlights the need to design a general model, resilient to over-fitting, able to extract the informative characteristics from the depicted scenes.

There are two prevalent formulations of the problem of global-scale location estimation and accordingly two solutions to tackle it: classification and retrieval. The former considers a classification task [15, 16, 29, 47], partitioning the earth's surface into a grid of cells, and then trains a classifier to assign input images to a grid cell. The latter considers location estimation as a retrieval task [12, 13, 20, 22, 30, 42], searching in a large-scale *background database* of geotagged images to retrieve similar ones based on a given query, and then aggregates them for estimating the location of the query image. Both formulations have limitations. The major drawback of classification approaches stems from the division of the earth into large geographic areas, which results in coarse estimations. A partial solution to that would be to consider a finer granularity grid, but this could potentially hurt the system's performance [16, 42, 47]. On the other hand, retrieval-based approaches perform worse than classification-based ones [16, 29, 47], and they need significantly more computational resources during inference. Also, our investigations indicated that they are prone to noise from the appearance of visually similar concepts within images that are not related to the particular location (e.g., humans, animals, vehicles).

Motivated by the above limitations, we propose a scheme that achieves high geolocation accuracy in all granularity scales. We build on the strong aspects of both classification and retrieval approaches. Their combination has already been employed in global-scale text-based geolocation solutions [19, 40]; yet, to the best of our knowledge, they have not been successfully employed in the visual domain. We leverage recent advances in the field of image classification, employing a state-of-the-art architecture, and contrastive learning [7, 14, 18], building a retrieval module that achieves better performance than using features directly extracted from pre-trained CNNs. In particular, we make the following contributions:

- We develop robust classification modules based on the state-of-the-art EfficientNet [37] architecture, which has not been employed before in the relevant literature, trained with three different training schemes from the literature.
- We build a retrieval module based on a residual architecture trained with contrastive learning. The network learns to capture location-relevant information and enriches the image features representations extracted from the CNN.
- We also propose the Search within Cell scheme that combines the two modules and estimates the final locations with an aggregation scheme based on spatial clustering.

- Our approach outperforms several state-of-the-art methods on four benchmark datasets, achieving up to 42% relative improvement at the 1km range on the Im2GPS3k. We also evaluate our method with various configurations to gain insight into its behaviour.

## 2 RELATED WORK

There are several works in the literature that tackle the problem of location estimation. These can be roughly classified into two categories according to [6, 28, 29]: (i) approaches restricted to specific environments or imagery, and (ii) planet-scale approaches without any restrictions. Our approach belongs to the second category.

The works in the first category focus on the localization on fine granularity scales, such as landmarks [3, 5, 46, 49] or at city-scale granularity [1, 17, 25, 39, 44]. In general, the solutions that are employed for such problems are based on retrieval systems that match the query images with ones from a background collection and then apply a post-processing scheme to estimate the final location. However, these methods use restricted data from popular scenes and urban environments and require many instance matches to perform robustly, which is infeasible at a global scale. Another instance that falls within this category are methods that estimate image locations from cross-view imagery, e.g., ground-to-aerial [23, 24, 34, 36, 43, 50]. Such methods are restricted on the type of imagery needed as input in order to perform location estimation.

The works in the second category tackle the location estimation problem under no constraints. Hays et al. [12] first introduced the problem with the composition of a dataset of about 6 million images collected from Flickr. They proposed an image retrieval method based on handcrafted features. A breakthrough was made by [47] when the authors formulated the problem as a classification one and trained a CNN, namely PlaNet, with the cross-entropy loss. The classes were defined using a heuristic process for the adaptive partitioning of the earth into geographic cells. Motivated by the PlaNet [47], a revision of the original Im2GPS paper was made in [42], where the authors proposed a retrieval approach for inference, extracting image features from a trained CNN. They also experimented with Deep Metric Learning for fine-tuning the network, yet without achieving significant performance gains. CPlaNet, a modification of the original PlaNet [47], was proposed by [15]. The authors used multiple coarse partitions of the earth and trained a different network for each of them. The final finer-granularity result was calculated by a *combinatorial partitioning* approach, which considered the intersections of the partitions. The authors of [29] experimented with different architectures, simultaneously using multiple cross-entropy loss functions corresponding to a coarse, middle, and fine-grained partition of the earth for the training of the CNN, and a fusion scheme of the probabilities in the three granularities for the inference of the estimated location. Additionally, they proposed as an initial step to split the images according to the scene they represent, such as indoor, natural or urban, and trained a different model per category. More recently, the authors of [16] proposed the use of the continuous von Mises-Fisher (vMF) distribution to model the geolocation problem as an alternative to the

simple classification approach. However, to the best of our knowledge, there has been no global-scale location estimation approach that successfully combines classification and retrieval. Also, the only recently proposed retrieval-based approach [42] did not manage to achieve better performance compared to using the features extracted from a pre-trained classification network.

## 3 METHODOLOGY

Figure 1 illustrates the proposed approach. Our method requires the partitioning of the earth's surface into cells (Section 3.1), which is the base of a classification and a retrieval module. We experiment with three practices to build the classification module (Section 3.2). For the retrieval module, we propose a residual architecture and train it with contrastive learning to map images to an embedding space, where the same location images have large similarity (Section 3.3). Finally, we combine the two modules with a search within cell scheme and a spatial clustering aggregation approach (Section 3.4).
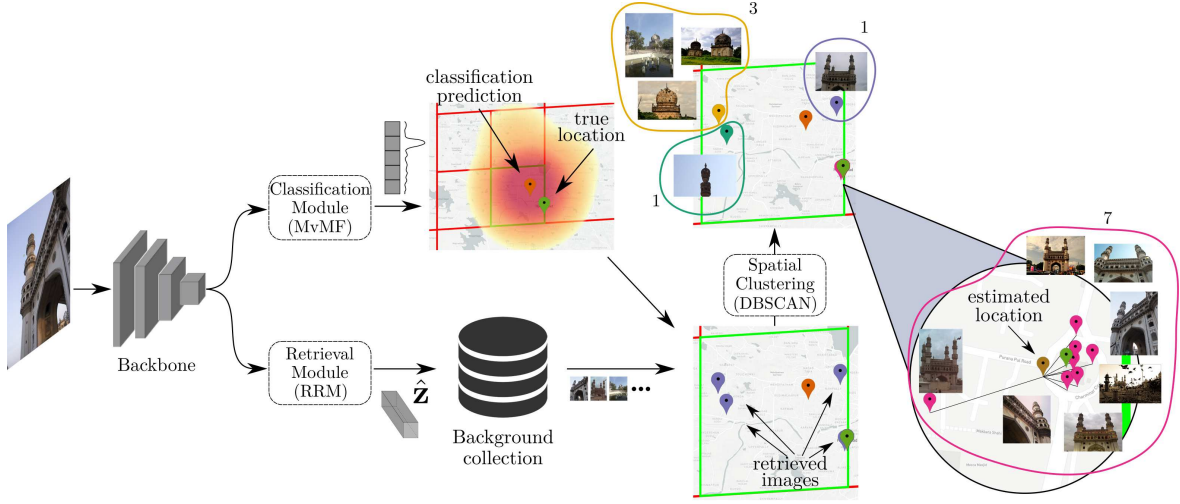
### 3.1 Earth Partitioning

As the probability distribution of the image locations is not uniformly distributed over the earth, it would make sense to create an adaptive partition of the Earth based on the training data. Similar to [29, 47], the steps of the process we follow are: (1) create a first coarse partition of the Earth, (2) assign to each cell the included images, (3) choose the cell with the most images and split it, (4) remove the parent cell and assign to each child cell the images located within its borders, (5) repeat steps 3-4 until the desired number of classes is reached. The above process requires a hierarchical representation of the Earth's surface; we used Google's S2 Geometry Library [11] similar to [29, 47]. It is also possible to choose different termination criteria, e.g., terminate when all classes have less images than a fixed threshold, or enforce additional restrictions, e.g., disallow classes to have less than a fixed number of images. For each cell of the resulting partition, we define its center that will be used as a point prediction, estimated as the average location of the image locations in the cell.

### 3.2 Classification module

We describe three ways the classification module can be implemented, for which we provide experimental results in Section 5.

*3.2.1 Discrete probability model trained with cross-entropy.* The most straightforward way to implement the classification module is with a CNN that outputs a discrete probability distribution over the cells defined in Section 3.1, as done by PlaNet [47]. The CNN weights can then be trained with the cross-entropy loss, which is commonly used in classification tasks.

*3.2.2 Hierarchical Classification (HC).* Another possibility is to follow a hierarchical approach [29] and simultaneously train three different classifiers, each implemented as described in Section 3.2.1, at different geographical resolutions. That is, we create three different partitions of the Earth, ranging from coarse-grained with a small number of cells to fine-grained with a large number of cells, and attach three different classification heads to the backbone

**Figure 1: Overview of the proposed Search within Cell (SwC) scheme. Given a query image, a backbone network is first used to extract representative features. Then: (i) a Classification Module predicts a cell on the earth's surface, and (ii) a Retrieval Module extracts an image embedding. These are combined with the Search within Cell scheme, where the most similar images that belong to the predicted cell are retrieved. The final location is estimated based on a Spatial Clustering scheme, where the most similar images are clustered based on their GPS coordinates.**

CNN, one per partition. The loss is then calculated as the average of the cross-entropy loss of each head.

Training the backbone with this loss, as done in [29], could potentially allow the network to achieve greater generalization power. However, due to the high computational cost, we fix the backbone network weights and train only the different classification heads, which is equivalent to training three independent models at different partitions. For the inference, we can take advantage of the multiple heads by combining the three outputs. Specifically, we 1) calculate the output of each head, 2) find for every cell of the fine partition its parent in the mid and coarse partitions, 3) multiply the probability of the cell with that of its parents, 4) select the cell of the fine partition with the highest probability.

*3.2.3 MvMF trained with log-likelihood loss.* A shortcoming of the two previous methods is that they do not take into account that the data points and classes are defined on the surface of the Earth, and as such are related to each other in a common spherical coordinate system. An alternative is to model the task in the continuous probability space using the von Mises-Fisher (vMF) probability distribution [16] that is specifically targeted towards modeling spherical data. The vMF distribution is defined as

$$\text{vMF}(x \mid \mu, \kappa) = \frac{\kappa}{4\pi \sinh \kappa} e^{\kappa \mu^T x} \quad (1)$$

where $\|\mu\| = \|x\| = 1$ is the mean and $\kappa > 0$ the concentration, and was used by the authors of [16] to build a probabilistic model of the geolocation task based on a Mixture of vMF distributions (MvMF) that would be trained with the log-likelihood loss: first, a partitioning of the Earth is constructed, as described in section 3.1, and each cell of this partitioning corresponds to a single component of the MvMF with mean the center of the contained images. Then, the probability that a given image $I$ is located at $x$ is given

by

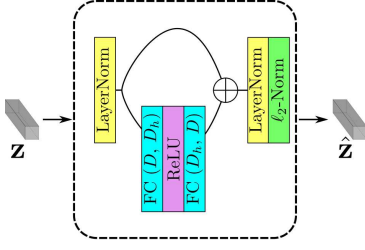$$\text{MvMF}(x \mid I) = \sum_{i=1}^{N} w_i(I)\text{vMF}(x \mid \mu_i, \kappa_i) \quad (2)$$

where $w(I)$ sums up to one and is calculated from the output of a CNN. The concentration, $\kappa$, of each vMF component is a parameter of the model that is learned during training, but is fixed and independent of the input image $I$ during inference. The network is trained with the negative log-likelihood loss, a common choice for training probabilistic mixture models.

As the final result, ideally, we would choose the location that maximizes the probability density function. However, this would involve solving a computationally intensive non-convex optimization problem; instead, we opt for approximating it with the location of the mean of the mixture component with the highest weight $w$.

## 3.3 Retrieval module

A robust retrieval system has to map the images in the dataset to an embedding space where the images from the same location are closer to each other than the rest. The goal is to alleviate the high inter-class ambiguity and intra-class diversity introduced by the weak supervision from training with GPS coordinates and noisy data. To this end, we build a network architecture and train it with supervised contrastive learning.

*3.3.1 Network architecture.* To build our retrieval system, we employ a backbone CNN training based on a classification scheme, and we extract feature representations for the images in our dataset. Given an input image, we feed it to the CNN and apply Global Average Pooling (GAP) on the activations of the final convolutional layer to extract a feature vector $z \in \mathbb{R}^D$, where $D$ is the dimensionality of the output vectors. Inspired by the Feed-Forward Layer

**Figure 2: Overview of the Residual Retrieval Module. The symbol ⊕ indicates element-wise summation.**

from [41], we build a residual network for the projection of the image vectors to the embedding feature space. Figure 2 displays the network architecture of the retrieval module. The network comprises two fully-connected layers, with $D_h$ hidden size and Rectified Linear Unit (ReLU) [21] activation between them. The output is added with the input feature vector with a residual connection. Before and after the residual connection, feature vectors are normalized with LayerNorm [4]. Finally, $\ell_2$-normalization is applied on the feature vectors to transform them to the unit sphere. The network output is an enriched embedding vector, $\hat{\mathbf{z}} \in \mathbb{R}^D$, with the same dimensionality as the input. With this architecture, our model is able to capture the relevant information from the feature vectors and enrich the image representations without drastically altering them. This is achieved with the use of the residual connection that we empirically found to boost the performance of the proposed system. Our main intuition for the proposed scheme is that the image vectors are already good representations and that the network learns to extract and amplify useful information in the output embeddings.

*3.3.2 Training process.* For training, we propose a supervised contrastive learning method [18] based on the image locations. We aim to map the images captured from the same location closer in the embedding space than the rest. Hence, given the Earth's partitioning as described in Section 3.1, images that belong to the same geographic cell are considered as positive pairs; instead, images from different cells are considered as negatives. Let $\hat{\mathbf{z}}_q$, $\hat{\mathbf{z}}_p$, and $\hat{\mathbf{z}}_n^i$, $i = 1, 2, ..., N - 2$, be the embeddings, derived from our network, of a query image, a positive image (an image from the same cell with the query), and several negative images (images from different cells). We train our network so that the similarity between the query-positive pair is significantly larger than the similarities between the query-negatives. Since all vectors are $\ell_2$-normalized, their dot product measures the similarity between images in the embedding space. To train our network, we employ the infoNCE [33] contrastive loss function, as follows:

$$\mathcal{L}_{nce} = -\log \frac{exp(\hat{\mathbf{z}}_q \cdot \hat{\mathbf{z}}_p / \tau)}{exp(\hat{\mathbf{z}}_q \cdot \hat{\mathbf{z}}_p / \tau) + \sum_{i=1}^{N-2} exp(\hat{\mathbf{z}}_q \cdot \hat{\mathbf{z}}_n^i / \tau)} \quad (3)$$

where $\tau$ is a temperature hyperparameter [48]. The training loss drops as the similarity between the query-positive pair is significantly greater than the similarities of the query and all negatives. Therefore, by minimizing this loss, we force the network to assign

high similarity values on the positive pairs, i.e., the images originating from the same cell, and low similarity values on the negative pairs, i.e., the images that come from different cells. To utilize more negative samples during the loss calculation, we employ a cross-batch memory bank [45, 48]. Additionally, to eliminate the bias from the cells that contain many images, in each training epoch, we sample one image pair from each cell. In that way, all cells are equally represented during the training process.

*3.3.3 Background collection.* In a retrieval system, it is essential to build a background collection where retrieval is performed. In the current problem, the selection of a representative background collection can affect the system performance, as pointed out in [42]. However, a huge background collection significantly increases the total time needed for retrieval. In this work, we build the background collection with images from the training set, and we populate it with those that our classifier is able to place in the correct geographic cell. In that way, we compose a background collection of "placeable" images, i.e., suitable images that serve our retrieval scheme for the visual location estimation task.

## 3.4 Search within Cell

We employ an aggregation scheme to combine the two geolocation modules, classification and retrieval, called Search within Cell (SwC). First, we perform classification to derive the cell with the largest probability. Then, we retrieve the top-most similar images of the background collection, with the constraint that the retrieved images fall within the borders of the estimated cell.

Finally, to enhance the robustness of the results, we develop a density-based spatial clustering scheme. Given a query image with its predicted cell, we first retrieve the top $K$ most similar images from the background collection that belongs to the same cell. Our intuition is that the most visually similar images are the most appropriate to infer the location of the query. Then, we apply the DBSCAN [10] algorithm on the GPS coordinates of the $K$ images to form clusters based on their spatial proximity. DBSCAN is selected because it does not require setting a predefined number of clusters, which would be impractical to select optimally at global scale. We use the geodesic distance as the function to calculate the distance between images. DBSCAN requires setting an $\varepsilon$ threshold, which corresponds to the maximum distance between two samples for the first to be considered in the neighborhood of the second. We empirically set $\varepsilon$ equal to 1km, as we found it to yield marginally better results. Also, DBSCAN may receive as argument the minimum number of samples in a neighborhood for a point to form a cluster. Since we do not want to force the merging of isolated images in the clusters during the final location estimation, we set this parameter to 1. In the end, the largest cluster (or the first one in rare cases of equal size) is selected. The final location estimation derives from the mean of the locations of the cluster's images.

## 4 EVALUATION SETUP

### 4.1 Datasets

**MediaEval Placing Task 2016 (MP16)** [8] is used for training and evaluation of our approach as provided by the original authors. It consists of approximately 5.8 million geotagged images

randomly selected from the YFCC100m [38] collection, without performing any filtering based on the image metadata. The dataset is split into two parts: (i) the training set, composed of 4.5 million images, which we use to train our models, and (ii) the test set, including the remaining 1.3 million images, which is used for evaluation.

**Im2GPS** [12] is used for comparison against existing methods as provided by the original authors. It consists of 237 images from the original Im2GPS dataset, manually selected by the authors of [12] from a set of 400 images based on their localizability.

**Im2GPS3k** [42] is also used for comparison against existing methods as provided by the original authors. It consists of 3,000 images from the original Im2GPS dataset. The images were not manually filtered; hence, it is a more challenging test compared to the previous one.

**YFCC4k** [42] is used for comparison against existing methods as provided by the original authors. It consists of 4,000 images from the YFCC100m [38] dataset without applying any filtering. Since it derives from a general purpose dataset, its image distribution is different from Im2GPS, making it more challenging.

## 4.2 Implementation details

For the proposed approach, we train an EfficientNet-B4 model [37], initialized with pre-trained weights on ImageNet [9]. For the training of the backbone CNN, we employ the simple cross-entropy scheme using 32K cells, which we consider as our re-implementation of PlaNet. The model is trained for 25 epochs with the entire training dataset after removing all the images of the users that appeared in the evaluation sets, according to [29, 42]. We train the network with Stochastic Gradient Descent (SGD) and step decay, with 0.01 initial learning rate, 0.5 decay factor with step 5 epochs, momentum 0.9, batch size 64, and weight decay $10^{-4}$. During training, we apply data augmentation by randomly cropping image areas that cover at least 70% of the original images with an aspect ratio in the range from 3/4 to 4/3. The input images are randomly flipped and resized to 300×300 pixels. During inference, we simply resize the images so that the largest side is 300 pixels. For validation, we use the same YFCC100m [38] subset as in [29]. After this training session, the weights of the CNN remain fixed and are not updated during the training of the rest of the modules.

For the development of the other two classification schemes, i.e., HC and MVMF, we use similar training processes with the one described above. For HC, we follow the earth partitioning proposed in the original approach [29]. For the MvMF, we use the same number of cells as in the previous setup. Also, we initialize the weights of the mixture layer with the ones from the classification layer. We train both methods for 5 epochs with the AdamW [27] optimizer for faster convergence, with $10^{-5}$ initial learning rate, and a cosine annealing learning rate [26] scheduler. We use the same batch size, weight decay, and augmentation process as above.

For the training of the retrieval scheme, we only use image features extracted from the CNN as described in Section 3.3.1. The network is trained for 200 epochs with an AdamW [27] optimizer, with $10^{-5}$ initial learning rate, and a cosine annealing learning rate scheduler. Also, we use a batch size of 64 image pairs and weight decay $10^{-4}$. The value of $\tau$ is set to 0.05, the size of the bank is

4096, and the hidden size of the network $D_h$ is 4096. Finally, after filtering the images wrongly placed by the classification network and the images of users that appear in the evaluation sets, the background collection amounts to 700K images. For the SwC scheme, we use the top 10 most similar images for the location estimation.

The training time on 4 Nvidia RTX 2080Ti was one week for the backbone CNN, approximately 8 hours for each classification scheme, and 2 hours for the retrieval scheme. The inference time of our system is 40ms per image.

## 4.3 Evaluation metrics

Following [12, 15, 42], we evaluate geolocation performance based on the percentage of images that are placed within a predefined granularity range. An image is considered correctly placed when the geodesic distance of the estimated location to the ground truth is lower than the granularity range. The geodesic distance is calculated as the Great Circle Distance (GCD) between the two locations. We consider two sets of granularity ranges: (i) the baseline granularity ranges, including 1km, 25km, 200km, 750km, and 2500km, corresponding roughly to street, city, region, country, and continent granularity level, and (ii) the fine-grained granularity ranges, including 100m, 1km, 5km, and 10km, that evaluate the methods' performance in fine granularities, which are more representative of the actual performance of a useful location estimation system.

## 5 EXPERIMENTS

In this section, we report the results of several runs following the proposed methodology. We compare the different versions of the implemented method against several state-of-the-art approaches (Section 5.1). Also, we provide an ablation study to evaluate the proposed approach under different configurations (Section 5.2).

## 5.1 Comparison against the state-of-the-art

Table 1 illustrates the performance of the proposed and state-of-the-art approaches on the four evaluation datasets. The performance is measured in all eight granularity ranges. The proposed approach is compared with several classification approaches, i.e., Hierarchical Classification (HC) and Individual Scene Networks (ISN) from [29], Mixture of von Mises-Fisher (MvMF) [16], the CPlaNet and Planet [47] re-implementation from [15], the retrieval approaches, i.e., RevIm2GPS [42], VGG-PCA [20], and the Feature Fusion approach by [30]. Moreover, we report the performance of our re-implementations for Planet [47], HC [29], and MvMF [16] trained based on our setup described in Section 4.2. Finally, the performance of our proposed Residual Retrieval Module (RRM) with nearest neighbour search is demonstrated, and it is combined with our re-implemented classification modules with our SwC scheme.

According to the results, our approach achieves superior performance, especially in fine granularity ranges. Comparing our re-implemented classification run with the original ones, it is clear that the use of the EfficientNet-B4 model significantly boosts performance in almost all granularity ranges, even though we did not train the backbone with HC, which could lead to even better performance according to [29]. Additionally, our SwC scheme considerably improves geolocation accuracy in fine ranges (i.e., <10km)

| Method | Type | Acc@ Im2GPS | | | | | | | | Acc@ Im2GPS3k | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 100m | 1km | 5km | 10km | 25km | 200km | 750km | 2500km | 100m | 1km | 5km | 10km | 25km | 200km | 750km | 2500km |
| HC [29] | C | - | 15.2 | - | - | 40.9 | 51.5 | 65.4 | 78.5 | - | 9.7 | - | - | 27.0 | 35.6 | 49.2 | 66.0 |
| ISN [29] | C | - | 16.9 | - | - | 43.0 | 51.9 | 66.7 | 80.2 | - | 10.5 | - | - | 28.0 | 36.6 | 47.7 | 66.0 |
| MvMF [16] | C | - | 8.4 | - | - | 32.6 | 39.4 | 57.2 | 80.2 | - | - | - | - | - | - | - | - |
| PlaNet† [47] | C | - | 11.0 | 23.6 | 26.6 | 31.2 | 37.6 | 64.6 | 81.9 | - | 8.5 | 18.1 | 21.4 | 24.8 | 34.3 | 48.8 | 64.6 |
| CPlaNet [15] | C | - | 16.5 | 29.1 | 33.8 | 37.1 | 46.4 | 62.0 | 78.5 | - | 10.2 | 20.8 | 23.7 | 26.5 | 34.6 | 48.6 | 64.6 |
| PlaNet‡ [47] | C | 4.2 | 17.3 | 33.8 | 38.0 | 41.8 | 53.2 | 67.9 | 82.3 | 2.8 | 11.8 | 22.1 | 25.3 | 28.8 | 37.4 | 51.0 | 67.4 |
| HC‡ [29] | C | 1.7 | 13.5 | 28.7 | 32.9 | 39.7 | 54.0 | <u>68.8</u> | **82.7** | 1.8 | 10.1 | 20.1 | 23.9 | 28.4 | 37.9 | 52.0 | **68.1** |
| MvMF‡ [16] | C | 4.6 | 19.8 | 34.2 | 40.1 | **44.7** | **55.7** | 67.5 | 81.9 | 3.0 | 13.1 | <u>23.5</u> | 26.7 | <u>29.8</u> | **38.0** | **52.3** | 67.6 |
| RevIm2GPS [42] | R | - | 14.4 | - | - | 33.3 | 47.7 | 61.6 | 73.4 | - | 7.2 | - | - | 19.4 | 26.9 | 38.9 | 55.9 |
| RRM | R | 5.1 | 19.4 | 35.0 | 37.1 | 40.5 | 51.1 | 60.4 | 78.1 | 3.6 | 12.4 | 20.4 | 23.5 | 26.0 | 34.0 | 46.9 | 63.6 |
| PlaNet‡ + RRM | SwC | 5.5 | <u>19.8</u> | <u>36.3</u> | 38.0 | 41.8 | 52.7 | 67.9 | 82.3 | **4.3** | <u>13.9</u> | 23.4 | 26.1 | 29.3 | 37.4 | 51.0 | 67.4 |
| HC‡ + RRM | SwC | 5.5 | 18.6 | 34.6 | 36.7 | 41.8 | <u>55.3</u> | **69.2** | 82.7 | 3.8 | 13.2 | 22.5 | 25.5 | 29.1 | 37.8 | 52.0 | **68.1** |
| MvMF‡ + RRM | SwC | **6.3** | **21.9** | **38.0** | **40.5** | <u>44.3</u> | <u>55.3</u> | 67.5 | 81.9 | <u>4.1</u> | **15.0** | **24.3** | **27.0** | **30.0** | **38.0** | **52.3** | 67.6 |

| Method | Type | Acc@ YFCC4k | | | | | | | | Acc@ MP16-test | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | 100m | 1km | 5km | 10km | 25km | 200km | 750km | 2500km | 100m | 1km | 5km | 10km | 25km | 200km | 750km | 2500km |
| PlaNet† [47] | C | - | 5.6 | 10.1 | 12.2 | 14.3 | 22.2 | 36.4 | 55.8 | - | - | - | - | - | - | - | - |
| CPlaNet [15] | C | - | **7.9** | **12.1** | **13.5** | **14.8** | 21.9 | 36.4 | 55.5 | - | - | - | - | - | - | - | - |
| PlaNet‡ [47] | C | 1.8 | 6.1 | 9.7 | 11.3 | 13.0 | 21.0 | 36.4 | 56.2 | 2.0 | 7.3 | 11.7 | 13.5 | 15.3 | 22.4 | 36.9 | 56.1 |
| HC‡ [29] | C | 1.1 | 5.7 | 9.0 | 10.9 | 13.1 | 21.6 | 36.6 | 55.4 | 1.3 | 6.3 | 10.7 | 12.6 | 14.8 | 22.5 | 37.2 | **56.3** |
| MvMF‡ [16] | C | 1.9 | 6.8 | 10.9 | 12.6 | <u>14.4</u> | 21.9 | **37.5** | <u>56.4</u> | 2.0 | 7.8 | <u>12.5</u> | <u>14.3</u> | **16.1** | **23.1** | 37.4 | **56.3** |
| FeatFusion [30] | R | - | - | - | - | - | - | - | - | 0.9 | 2.4 | 4.0 | 4.6 | 5.2 | 7.3 | 17.2 | 35.4 |
| VGG-PCA [20] | R | - | - | - | - | - | - | - | - | 1.8 | 5.6 | 7.5 | 8.2 | 8.8 | 12.1 | 22.4 | 40.8 |
| RevIm2GPS [42] | R | - | 2.3 | - | - | 5.7 | 11.0 | 23.5 | 42.0 | - | - | - | - | - | - | - | - |
| RRM | R | 2.3 | 6.0 | 9.0 | 10.2 | 11.3 | 16.8 | 30.5 | 49.4 | **2.9** | 7.5 | 10.2 | 12.1 | 13.4 | 19.0 | 32.3 | 51.6 |
| PlaNet‡ + RRM | SwC | **2.7** | 7.2 | 10.3 | 11.8 | 13.0 | 20.9 | 36.4 | 56.2 | 2.8 | <u>8.5</u> | 12.4 | 13.8 | 15.4 | 22.4 | 36.9 | 56.1 |
| HC‡ + RRM | SwC | 2.5 | 7.2 | 10.3 | 11.7 | 13.3 | 21.6 | 36.5 | 55.4 | 2.5 | 8.0 | 11.9 | 13.3 | 15.0 | 22.5 | 37.1 | **56.3** |
| MvMF‡ + RRM | SwC | **2.7** | 7.9 | <u>11.3</u> | <u>12.9</u> | 14.3 | 21.9 | <u>37.4</u> | **56.5** | **2.9** | **8.9** | **13.1** | **14.5** | **16.1** | **23.1** | 37.4 | **56.3** |

**Table 1: Accuracy (%) on all eight granularity ranges of the proposed and state-of-the-art approaches on four public datasets. The second column indicates the type of the method, C stands for classification, R for retrieval, and SwC for search within cell. The best performances are highlighted in bold, and the second-best are underlined. † indicates the results of the re-implemented methods by [15]. ‡ indicates the results of our re-implemented methods.**

in all evaluation datasets compared to the individual runs, achieving as high as 2% absolute performance gain at the 1km range. This highlights that the proposed scheme can operate well with various combinations of classification-retrieval systems. Also, SwC hurts the performance of the classification systems only in very rare occasions and in coarse granularity ranges (i.e., >25km). The HC sees the greatest improvement with the application of the SwC since it uses a coarser grid compared with PlaNet and MvMF, leaving more room for improvement. Finally, our RRM method achieves the best results among the retrieval runs utilizing 700K images as the background collection, which is only a small fraction in comparison to other approaches; RevIm2GPS uses 6 million images, and VGG-PCA and Feature Fusion the entire MP16 training set.

In the Im2GPS, the best performing approach in fine granularity ranges is our SwC scheme implemented with MvMF as the classification module and our RRM. More precisely, it achieves 21.9% at 1km granularity range, which is a relative improvement of almost 30% of the previous state-of-the-art achieved by the ISN with 16.9%. It only leads to a marginal drop at the 25km and 200km ranges. Furthermore, our SwC scheme implemented with HC and RRM achieves the best results in the coarser granularity ranges. It also marginally improves the performance of the classification module at 750km. It is noteworthy that our retrieval module outperforms

almost all of the classification methods by a significant margin in fine granularity ranges, i.e., <10km.

In the Im2GPS3k, our SwC scheme with MvMF and RRM outperforms all other approaches by a considerable margin in almost all ranges. It outperforms the previous state-of-the-art ISN method at the 1km range by an absolute difference of 4.5%. The SwC boosts the performance of the classification module in all fine granularity ranges. Our RRM module demonstrates competitive performance, in particular, at 100m and 1km ranges where it has the best, and second-best performance among the individual runs (i.e., classification and retrieval).

Regarding YFCC4k, the SwC run with MvMF+RRM leads to the best results in the finer and coarser ranges, i.e., ≤1km and ≥200km, and the second-best in the ranges from 5km to 25km behind CPlaNet. However, we empirically found that using images of the users in the evaluation set for training of the classification modules considerably improves performance, i.e., more than an absolute 2% in any granularity. Nevertheless, it is not clear in [15] whether such images were used during training of the CPlaNet and PlaNet re-implementation. Also, it is worth noting that the performance of all runs is considerably lower, indicating that YFCC4k is much more challenging than Im2GPS. This is expected since it consists of random images from the YFCC100m without any filtering. Such

| loss | 1km | 25km | 200km | 750km | 2500km |
|---|---|---|---|---|---|
| baseline | 11.9 | 24.7 | 31.6 | 43.4 | 59.4 |
| triplet | 11.6 | 24.4 | 32.0 | 45.0 | 62.2 |
| infoNCE | **12.4** | **26.0** | **34.0** | **46.9** | **63.6** |

**Table 2: Accuracy (%) on the baseline granularity ranges of the baseline and the proposed retrieval module trained with infoNCE and triplet loss on Im2GPS3k.**

| Network | 1km | 25km | 200km | 750km | 2500km |
|---|---|---|---|---|---|
| w/o residual | 11.2 | 24.8 | 32.6 | 45.5 | 62.0 |
| w/ residual | **12.4** | **26.0** | **34.0** | **46.9** | **63.6** |

**Table 3: Accuracy (%) on the baseline granularity ranges of the proposed retrieval module with and without the residual connection on Im2GPS3k.**

| background col. | 1km | 25km | 200km | 750km | 2500km |
|---|---|---|---|---|---|
| proposed | **12.4** | 26.0 | 34.0 | 46.9 | **63.6** |
| all train set | 10.5 | **26.8** | **34.9** | **47.2** | 63.5 |

**Table 4: Accuracy (%) on the baseline granularity ranges of the proposed retrieval module with different background collection on Im2GPS3k.**

| $K$ | 100m | 1km | 5km | 10km |
|---|---|---|---|---|
| 1 | 3.6 | 14.4 | 24.1 | 26.8 |
| 5 | 4.0 | 14.7 | **24.3** | **27.1** |
| 10 | **4.1** | **15.0** | **24.3** | 27.0 |
| 15 | 3.9 | 14.9 | 24.1 | 27.0 |
| 20 | 3.8 | 14.7 | 24.1 | 26.8 |

**Table 5: Accuracy (%) on the fine-grained granularity ranges of the proposed SwC scheme with MvMF and RRM modules for different values of $K$ on Im2GPS3k.**

| Aggregation | 100m | 1km | 5km | 10km |
|---|---|---|---|---|
| Average | 3.9 | 14.1 | 24.0 | 26.8 |
| KDE [42] | 3.8 | 14.5 | 24.1 | 26.8 |
| Spatial clustering | **4.2** | **15.0** | **24.3** | **27.0** |

**Table 6: Accuracy (%) on the fine-grained granularity ranges of the proposed SwC scheme with MvMF and RRM modules with different aggregations on Im2GPS3k.**

images may not be appropriate for the evaluation of the geolocation problem. Yet, this dataset simulates an unconstrained scenario where any arbitrary image has to be geolocated.

Finally, the performance on the MP16-test is similar to the YFCC4k since both datasets derive from the same distribution, i.e., they are random samples from the YFCC100m. The SwC run with MvMF+RRM outperforms all others in all granularity ranges. The results of the two retrieval runs were provided by the organizers of the MediaEval Placing Task 2016. Our RRM achieves significantly better results than the previous retrieval approaches, highlighting the progress in the field over the last years.

## 5.2 Ablation study

This section provides an ablation study on the Im2GPS3k dataset for our proposed RRM module and SwC scheme. We benchmark their geolocation performance to better understand their behavior under different configuration settings.

First, we compare the performance of the proposed retrieval module trained with different loss functions and against a baseline run. For the baseline, the features extracted from the CNN are directly used for retrieval without the application of the retrieval module or any further training. Also, for the training of the RRM, we compare the infoNCE loss against the triplet loss employed in [42]. We use a margin of 0.01, and semi-hard mining [35], which yielded the best performance. Table 2 depicts the results in the baseline granularity ranges. The proposed scheme with infoNCE achieves the best performance in all ranges. It outperforms the baseline by a significant margin. The run with the triplet loss does not lead to competitive results even in comparison to the baseline, in accordance with the observations in [42].

Additionally, in Table 3 we evaluate the impact of the residual connection on the geolocation performance of the RRM module. It is evident that the application of the residual connection considerably improves performance, highlighting its importance to the proposed system. Comparing these runs with the baseline one from Table 2, it appears that the residual connection leads to a clear accuracy increase across all ranges.

Table 4 depicts the performance of the RRM using the proposed background collection (700K images) and the entire training set

(4M images). With the proposed background collection, we achieve considerably better accuracy at the 1km range with almost 2% difference; whereas, for most of the other ranges, the use of the entire training set provides marginally better performance. Considering that the proposed collection is only a fraction (17.5%) of the training set, which translates to much faster retrieval and lower memory requirements, we find that it strikes an excellent trade-off between accuracy and speed.

We also investigate the impact of the selection of the temperature $\tau$ hyperparameter. The best performance is achieved when $\tau$ equals 0.05, which drops for greater or lower values (e.g., for 0.1 and 0.01, Acc@1km is 11.8 and 12.1, respectively). We also tested various sizes of the cross-batch memory bank, and conclude that the larger the size, the better the performance. Due to GPU memory limitations, we experimented with memory size up to 4096 vectors.

Moreover, we assess the impact of the selection of $K$ for the SwC scheme. Table 5 presents the accuracy of the approach for various $K$ values in the fine-grained granularity range. Our system achieves the best results for $K = 10$ in finer ranges, i.e., 100m and 1km, and for $K = 5$ in coarser ranges. For values greater than 10, the performance starts dropping.

Finally, we benchmark three aggregation schemes for the final location estimation. We compare the proposed spatial clustering with a simple averaging of the image coordinates and with Kernel Density Estimation (KDE), as proposed in [42]. For all schemes, the top-10 similar images are used for the location estimation. Table 6

Query                                Retrieved images                              Query                                Retrieved images



(a) correct classification - correct retrieval

(b) wrong classification - correct retrieval



0       1        100      >1000 (km)

(c) correct classification - wrong retrieval

(d) wrong classification - wrong retrieval

**Figure 3: Top-5 images from the background collection retrieved by our RRM given the image in the left as query. Retrieved images are coloured based on their distance to the ground truth location of the query: green indicates less than 1km, yellow within 1km and 100km, and red more than 100km. The images are grouped based on the predictions of the RRM and the MvMF classification modules: (a) both modules found the correct location, (b) correct predictions by the RMM and wrong by the MvMF, (c) wrong predictions by the RMM and correct by the MvMF, (d) both modules wrong. (best viewed in color)**

depicts the results in the fine-grained granularity range. It is evident that the proposed approach achieves the best performance by a considerable margin in all granularity ranges.

## 5.3 Qualitative evaluation

In this section, we provide some visual examples of the retrieved images based on our RRM module, given some queries. The top-5 images are illustrated in colour based on their geodesic distance from the queries' ground truth location. Also, the images are grouped according to the performance of the RRM and MvMF modules. Figure 3(a) displays the images that were placed within 1km from their true location by both methods. It is evident that many visual cues are present, mapping the images to their precise locations. Figure 3(b) presents the queries that were correctly placed by the RRM but missed by the MvMF. There are visual cues in the queries mapping the images to their locations; thus, the retrieval module can find several related images from the same location, highlighting that there is room for improvement in our proposed SwC scheme. Figure 3(c) shows some example queries that were wrongly placed by the retrieval module but correctly placed by the classification module. It is noteworthy that the retrieval module is distracted by the same concepts that appear in both query and reference images, i.e., the child in the first example, the donkeys in the second, and the bus in the third. Such cases are correctly addressed with our

SwC scheme, as it confines the RRM to search for similar images within the borders of the cell predicted by the MvMF. Finally, Figure 3(d) illustrates examples of queries that were wrongly placed by both modules. These cases either lack visual cues to map them to their location, i.e., in the first two examples, or cues are too ambiguous mapping the images to multiple locations, i.e., in the third example where many buildings with similar architectural style exist in different locations.

## 6 CONCLUSIONS

In this paper, we proposed a method for planet-scale location estimation that combines a classification and a retrieval module to estimate the location of a query image. We built three state-of-the-art classification schemes using EfficientNet [37] as backbone and proposed a retrieval module based on a residual architecture trained with contrastive learning. Our method exhibits very competitive performance on four datasets, significantly improving the state-of-the-art in many granularity ranges. In the future, we plan to investigate leveraging text annotations of images during training in order to build more robust classification and retrieval models.

# REFERENCES

[1] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *IEEE conference on computer vision and pattern recognition*. 5297–5307.

[2] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. 2015. Towards life-long visual localization using an efficient matching of binary sequences from images. In *2015 IEEE international conference on robotics and automation (ICRA)*. IEEE, 6328–6335.

[3] Yannis Avrithis, Yannis Kalantidis, Giorgos Tolias, and Evaggelos Spyrou. 2010. Retrieving landmark and non-landmark images from community photo collections. In *18th ACM international conference on Multimedia*. 153–162.

[4] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[5] Andrei Boiarov and Eduard Tyantov. 2019. Large scale landmark recognition via deep metric learning. In *Proceedings of the ACM International Conference on Information and Knowledge Management*. 169–178.

[6] Jan Brejcha and Martin Čadík. 2017. State-of-the-art in visual geo-localization. *Pattern Analysis and Applications* 20, 3 (2017), 613–637.

[7] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*. PMLR, 1597–1607.

[8] Jaeyoung Choi, Claudia Hauff, Olivier Van Laere, and Bart Thomee. 2016. The Placing Task at MediaEval 2016. In *Working Notes Proceedings of the MediaEval 2016 Workshop*.

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 248–255.

[10] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (Portland, Oregon) *(KDD'96)*. 226–231.

[11] google. 2020. S2 Geometry Library. https://github.com/google/s2geometry.

[12] James Hays and Alexei A Efros. 2008. IM2GPS: estimating geographic information from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[13] James Hays and Alexei A Efros. 2015. Large-scale image geolocalization. In *Multimodal location estimation of videos and images*. 41–62.

[14] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9729–9738.

[15] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. 2018. CPlaNet: Enhancing image geolocalization by combinatorial partitioning of maps. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 536–551.

[16] Mike Izbicki, Evangelos E Papalexakis, and Vassilis J Tsotras. 2019. Exploiting the Earth's Spherical Geometry to Geolocate Images. In *Proceedings of the Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

[17] Alex Kendall, Matthew Grimes, and Roberto Cipolla. 2015. Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Proceedings of the IEEE international conference on computer vision*. 2938–2946.

[18] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2020. Supervised Contrastive Learning. *Advances in Neural Information Processing Systems* 33 (2020).

[19] Giorgos Kordopatis-Zilos, Symeon Papadopoulos, and Ioannis Kompatsiaris. 2017. Geotagging text content with language models and feature mining. *Proc. IEEE* 105, 10 (2017), 1971–1986.

[20] G Kordopatis-Zilos, A Popescu, S Papadopoulos, and Y Kompatsiaris. 2016. Placing images with refined language models and similarity search with PCA-reduced VGG features. In *2016 Multimedia Benchmark Workshop, MediaEval 2016*, Vol. 1739. CEUR-WS.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* 25 (2012), 1097–1105.

[22] Xinchao Li, Martha Larson, and Alan Hanjalic. 2017. Geo-distinctive visual element matching for location estimation of images. *IEEE Transactions on Multimedia* 20, 5 (2017), 1179–1194.

[23] Tsung-Yi Lin, Serge Belongie, and James Hays. 2013. Cross-view image geolocalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 891–898.

[24] Liu Liu and Hongdong Li. 2019. Lending orientation to neural networks for cross-view geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 5624–5633.

[25] Liu Liu, Hongdong Li, and Yuchao Dai. 2019. Stochastic attraction-repulsion embedding for large scale image localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2570–2579.

[26] Ilya Loshchilov and Frank Hutter. 2016. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983* (2016).

[27] Ilya Loshchilov and Frank Hutter. 2018. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

[28] Carlo Masone and Barbara Caputo. 2021. A Survey on Deep Visual Place Recognition. *IEEE Access* (2021).

[29] Eric Muller-Budack, Kader Pustu-Iren, and Ralph Ewerth. 2018. Geolocation estimation of photos using a hierarchical model and scene classification. In *Proceedings of the European Conference on Computer Vision*.

[30] Javier AV Muñoz, Lin Tzy Li, Ícaro C Dourado, Keiller Nogueira, Samuel G Fadel, Otávio AB Penatti, Jurandy Almeida, Luís AM Pereira, Rodrigo T Calumby, Jefersson A dos Santos, et al. 2016. RECOD@ Placing Task of MediaEval 2016: A Ranking Fusion Approach for Geographic-Location Prediction of Multimedia Objects. In *2016 Multimedia Benchmark Workshop, MediaEval 2016*, Vol. 1739. CEUR-WS.

[31] Michal Nowicki, Jan Wietrzykowski, and Piotr Skrzypczyński. 2016. Experimental evaluation of visual place recognition algorithms for personal indoor localization. In *2016 International conference on indoor positioning and indoor navigation (IPIN)*. IEEE, 1–8.

[32] Michał R Nowicki, Jan Wietrzykowski, and Piotr Skrzypczyński. 2017. Real-time visual place recognition for personal localization on a mobile device. *Wireless Personal Communications* (2017).

[33] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[34] Krishna Regmi and Mubarak Shah. 2019. Bridging the domain gap for ground-to-aerial image matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 470–479.

[35] Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 815–823.

[36] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. 2019. Spatial-Aware Feature Aggregation for Cross-View Image based Geo-Localization. *Advances in Neural Information Processing Systems* (2019), 10090–10100.

[37] Mingxing Tan and Quoc Le. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*. PMLR, 6105–6114.

[38] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The new data in multimedia research. *Commun. ACM* 59, 2 (2016), 64–73.

[39] Akihiko Torii, Hajime Taira, Josef Sivic, Marc Pollefeys, Masatoshi Okutomi, Tomas Pajdla, and Torsten Sattler. 2019. Are Large-Scale 3D Models Really Necessary for Accurate Visual Localization? *IEEE transactions on pattern analysis and machine intelligence* (2019).

[40] Olivier Van Laere, Steven Schockaert, and Bart Dhoedt. 2011. Finding locations of Flickr resources using language models and similarity search. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*. 1–8.

[41] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6000–6010.

[42] Nam Vo, Nathan Jacobs, and James Hays. 2017. Revisiting IM2GPS in the deep learning era. In *Proceedings of IEEE International Conference on Computer Vision*.

[43] Nam N Vo and James Hays. 2016. Localizing and orienting street views using overhead imagery. In *European conference on computer vision*. Springer, 494–509.

[44] Han Wang, Chen Wang, and Lihua Xie. 2020. Online Visual Place Recognition via Saliency Re-identification. *arXiv preprint arXiv:2007.14549* (2020).

[45] Xun Wang, Haozhi Zhang, Weilin Huang, and Matthew R Scott. 2020. Cross-batch memory for embedding learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6388–6397.

[46] Tobias Weyand, Andre Araujo, Bingyi Cao, and Jack Sim. 2020. Google landmarks dataset v2-a large-scale benchmark for instance-level recognition and retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2575–2584.

[47] Tobias Weyand, Ilya Kostrikov, and James Philbin. 2016. PlaNet-photo geolocation with convolutional neural networks. In *Proceedings of the European Conference on Computer Vision*.

[48] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3733–3742.

[49] Shuhei Yokoo, Kohei Ozaki, Edgar Simo-Serra, and Satoshi Iizuka. 2020. Two-stage discriminative re-ranking for large-scale landmark retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 1012–1013.

[50] Sijie Zhu, Taojiannan Yang, and Chen Chen. 2021. Revisiting street-to-aerial view image geo-localization and orientation estimation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 756–765.