Where We Are and What We're Looking At: Query Based Worldwide Image Geo-localization Using Hierarchies and Scenes

Brandon Clark*, Alec Kerrigan*, Parth Parag Kulkarni, Vicente Vivanco Cepeda, Mubarak Shah
Center for Research in Computer Vision, University of Central Florida, Orlando, USA
{brandonclark314, aleckerrigan, parthpkulkarni.pk, vicente.vivanco}@knights.ucf.edu,
shah@crcv.ucf.edu

Abstract

Determining the exact latitude and longitude that a photo was taken is a useful and widely applicable task, yet it remains exceptionally difficult despite the accelerated progress of other computer vision tasks. Most previous approaches have opted to learn single representations of query images, which are then classified at different levels of geographic granularity. These approaches fail to exploit the different visual cues that give context to different hierarchies, such as the country, state, and city level. To this end, we introduce an end-to-end transformer-based architecture that exploits the relationship between different geographic levels (which we refer to as hierarchies) and the corresponding visual scene information in an image through hierarchical cross-attention. We achieve this by learning a query for each geographic hierarchy and scene type. Furthermore, we learn a separate representation for different environmental scenes, as different scenes in the same location are often defined by completely different visual features. We achieve state of the art accuracy on 4 standard geo-localization datasets : Im2GPS, Im2GPS3k, YFCC4k, and YFCC26k, as well as qualitatively demonstrate how our method learns different representations for different visual hierarchies and scenes, which has not been demonstrated in the previous methods. Above previous testing datasets mostly consist of iconic landmarks or images taken from social media, which makes the dataset a simple memory task, or makes it biased towards certain places. To address this issue we introduce a much harder testing dataset, Google-World-Streets-15k, comprised of images taken from Google Streetview covering the whole planet and present state of the art results. Our code can be found at https://github.com/AHKerrigan/GeoGuessNet.

*These authors contributed equally to the work

1. Introduction

Image geo-localization is the task of determining the GPS coordinates of where a photo was taken as precisely as possible. For certain locations, this may be an easy task, as most cities will have noticeable buildings, landmarks, or statues that give away their location. For instance, given an image of the Eiffel Tower one could easily assume it was taken somewhere in Paris. Noticing some of the finer features, like the size of the tower in the image and other buildings that might be visible, a prediction within a few meters could be fairly easy. However, given an image from a small town outside of Paris, it may be very hard to predict its location. Certain trees or a building's architecture may indicate the image is in France, but localizing finer than that can pose a serious challenge. Adding in different times of day, varying weather conditions, and different views of the same location makes this problem even more complex as two images from the same location could look wildly different.

Many works have explored solutions to this problem, with nearly all works focusing on the retrieval task, where query images are matched to a gallery of geo-tagged images to retrieve matching geo-tagged image [14,16,17,20,24,25]. There are two variations of the retrieval approach to this problem, same-view and cross-view. In same-view both the query and gallery images are taken at ground level. However, in cross-view the query images are ground level while the gallery images are from an aerial view, either by satellite or drone. This creates a challenging task as images with the exact same location look very different from one another. Regardless of same-view or cross-view, the evaluation of the retrieval task is costly as features need to be extracted and compared for every possible match with geo-tagged gallery images, making global scale geo-localization costly if not infeasible.

If, instead, the problem is approached as a classification task, it's possible to localize on the global scale given enough training data [8,11,12,15,21,22]. These approaches

segment the Earth into Google's S2[1] cells that are assigned GPS locations and serve as classes, speeding up evaluation. Most previous classification-based visual geo-localization approaches use the same strategy as any other classification task: using an image backbone (either a Convolutional Neural Network or a Vision Transformer [2]), they learn a set of image features and output a probability distribution for each possible location (or class) using an MLP. In more recent works [11, 12], using multiple sets of classes that represent different global scales, as well as utilizing information about the scene characteristics of the image has shown to improve results. These approaches produce one feature vector for an image and presume that it is good enough to localize at every geographic level. However, that is not how a human would reason about finding out their location. If a person had no idea where they were, they would likely search for visual cues for a broad location (country, state) before considering finer areas. Thus, a human would look for a different set of features for each geographic level they want to predict.

In this paper, we introduce a novel approach toward world-wide visual geo-localization inspired by human experts. Typically, humans do not evaluate the entirety of a scene and reason about its features, but rather identify important objects, markers, or landmarks and match them to a cache of knowledge about various known locations. In our approach, we emulate this by using a set of learned latent arrays called "hierarchy queries" that learn a different set of features for each geographic hierarchy. These queries also learn to extract features relative to specific scene types (e.g. forests, sports fields, industrial, etc.). We do this so that our queries can focus more specifically on features relevant to their assigned scene as well as the features related to their assigned hierarchy. This is done via a Transformer Decoder that cross-attends our hierarchy and scene queries with image features that are extracted from a backbone. We also implement a "hierarchy dependent decoder" that ensures our model learns the specifics of each individual hierarchy. To do this our "hierarchy dependent decoder" separates the queries according to their assigned hierarchy, and has independent weights for the Self-Attention and Feed-Forward stages that are specific to each hierarchy.

We also note that the existing testing datasets contain implicit biases which make them unfit to truly measure a model's geo-location accuracy. For instance, Im2GPS [4, 21] datasets contain many images of iconic landmarks, which only tests whether a model has seen and memorized the locations of those landmarks. Also, YFCC [18, 21] testing sets are composed entirely of images posted online that contained geo-tags in their metadata. This creates a bias towards locations that are commonly visited and posted online, like tourist sites. Previous work has found this introduces significant geographical and often racial biases into the datasets [7] which we demonstrate in Figure 4. To this end, we introduce a challenging new testing dataset called Google-World-Streets-15k, which is more evenly distributed across the Earth and consists of real-world images from Google Streetview.

The contributions of our paper include: (1) The first Transformer Decoder for worldwide image geo-localization. (2) The first model to produce multiple sets of features for an input image, and the first model capable of extracting scene-specific information without needing a separate network for every scene. (3) A new testing dataset that reduces landmark bias and reduces biases created by social media. (4) A significant improvement over previous SOTA methods on all datasets. (5) A qualitative analysis of the features our model learns for every hierarchy and scene query.

2. Related Works

2.1. Retrieval Based Image Geo-Localization

The retrieval method for geo-localization attempts to match a query image to target image(s) from a reference database (gallery). Most methods train by using separate models for the ground and aerial views, bringing the features of paired images together in a shared space. Many different approaches have been proposed to overcome the domain gap, with some methods implementing GANs [3] that map images from one view to the other [14], others use a polar transform that makes use of the prior geometric knowledge to alter aerial views to look like ground views [16, 17], and a few even combine the two techniques in an attempt to have the images appear even more similar [20].

Most methods assume that the ground and aerial images are perfectly aligned spatially. However, this is not always the case. In circumstances where orientation and spatial alignment aren't perfect, the issue can be accounted for ahead of time or even predicted [17]. VIGOR [25] creates a dataset where the spatial location of a query image could be located anywhere within the view of its matching aerial image. Zhu [24] strays from the previous methods by using a non-uniform crop that selects the most useful patches of aerial images and ignores others.

2.2. Image Geo-Localization as Classification

By segmenting the Earth's surface into distinct classes and assigning a GPS coordinate to each class, a model is allowed to predict a class directly instead of comparing features to a reference database. Treating geo-localization this way was first introduced by Weyand et al. [22]. In their paper, they introduce a technique to generate classes that utilizes Google's S2 library and a set of training GPS coordinates to partition the Earth into cells, which are treated

[1] https://code.google.com/archive/p/s2-geometry-library/

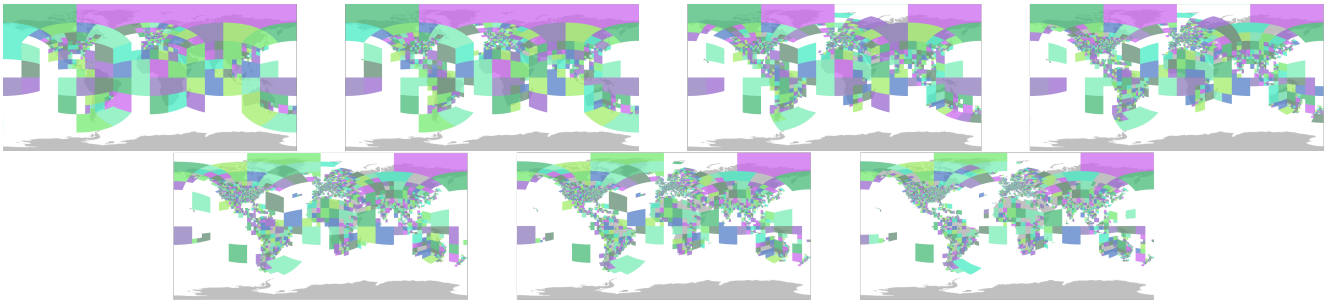


Figure 1. A visualization of all 7 hierarchies used. The t_{max} value is set to 25000, 10000, 5000, 2000, 1000, 750, and 500 respectively for hierarchies 1 to 7, while the t_{min} value is set at 50 for every hierarchy. This generates 684, 1744, 3298, 7202, 12893, 16150, and 21673 classes for hierarchies 1 to 7 respectively.

as classes. Vo [21] was the first to introduce using multiple different partitions of varying granularity. In contrast, CPlaNet [15] develops a technique that uses combinatorial partitioning. This approach uses multiple different coarse partitions and encodes each of them as a graph, then refining the graph by merging nodes. More details on class generation will be discussed in Section 3.1.

Up until Individual Scene Networks (ISNs) [11], no information other than the image itself was used at training time. The insight behind ISNs was that different image contexts require different features to be learned in order to accurately localize the image. They make use of this by having three separate networks for indoor, natural, and urban images respectively. This way each network can learn the important features for each scene and more accurately predict locations. The use of hierarchical classes was also introduced in [11]. While previous papers had utilized multiple geographic partitions, the authors in [11] observed that these partitions could be connected through a hierarchical structure. To make use of this, they proposed a new evaluation technique that combines the predictions of multiple partitions, similar to YOLO9000 [13], which helps refine the overall prediction. Kordopatis-Zilos [8] developed a method that combines classification and retrieval. Their network uses classification to get a predicted S2 cell, then retrieval within that cell to get a refined prediction.

Most recently, TransLocator [12] was introduced, which learns from not only the RGB image but also the segmentation map produced by a trained segmentation network. Providing the segmentation map allows TransLocator to rely on the segmentation if there are any variations in the image, like weather or time of day, that would impact a normal RGB-based model.

All of these methods fail to account for features that are specific to different geographic hierarchies and don't fully utilize scene-specific information. We solve these problems with our query-based learning approach.

3. Method

In our approach, we treat discrete locations as classes, obtained by dividing the planet into Schneider-2 cells at different levels of geographic granularity. The size of each cell is determined by the number of training images available in the given region, with the constraint that each cell has approximately the same number of samples. We exploit the hierarchical nature of geo-location by learning different sets of features for each geographic hierarchy and for each scene category from an input image. Finally, we classify a query image by selecting the set of visual features correlated with the most confident scene prediction. We use these sets of features to map the image to an S2 cell at each hierarchical level and combine the predictions at all levels into one refined prediction using the finest hierarchy.

3.1. Class Generation

With global geo-localization comes the problem of separating the Earth into classes. A naive way to do this would be to simply tessellate the earth into the rectangles that are created by latitude and longitude lines. This approach has a few issues, for one the surface area of each rectangle will vary with the distance from the poles, producing large class imbalances. Instead, we utilize Schneider 2 cells using Google's S2 Library. This process initially projects the Earth onto 6 sides of a cube, thereby resulting in an initial 6 S2 cells. To create balanced classes, we split each cell with more than t_{max} images from the training set located inside of it. We ignore any cells that have less than t_{min} to ensure that classes have a significant number of images. The cells are split recursively until all cells fall within t_{min} and t_{max} images. This creates a set of balanced classes that cover the entire Earth. These classes and hierarchies are visualized in Figure 1 where we can see the increasing specificity of our hierarchies. We begin with 684 classes at our coarsest hierarchy and increase that to 21673 at our finest. During evaluation we define the predicted location as the mean of the location of all training images inside a predicted class.

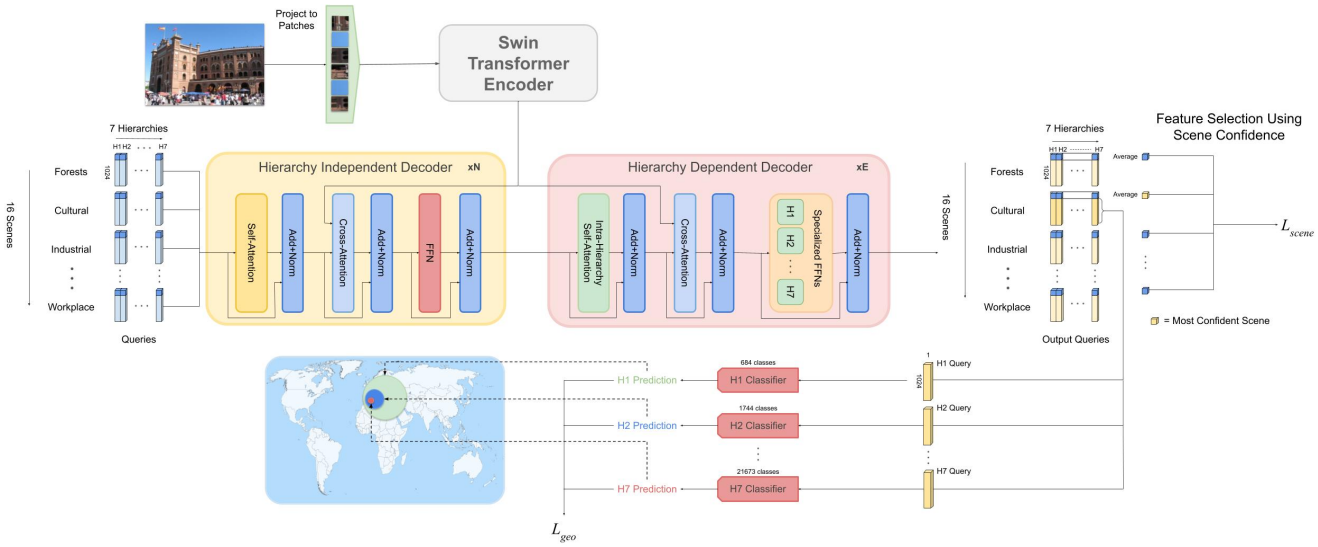


Figure 2. Our proposed network. We randomly initialize a set of learned queries for each hierarchy and scene. An image is first encoded by Transformer Encoder and decoded by two decoders. The first decoder consists of N layers as a Hierarchy Independent Decoder, followed by E layers of our Hierarchy Dependent Decoder; this decoder only performs self-attention within each hierarchy, instead of across all hierarchies, and has separate Feed-Forward Networks for each hierarchy. To determine which scene to use for prediction, the scene with the highest average confidence (denoted by the 0^{th} channel) is selected and queries are fed to their corresponding classifier to geo-localize at each hierarchy. We get a final prediction by multiplying the class probabilities of the coarser hierarchies into the finer ones so that a prediction using all hierarchical information can be made.

3.2. Model

Our model is shown in Figure 2, which is consists of a SWIN encoder, two decoders, and seven hierarchy classifiers. Here we outline the details behind our model's design.

One problem faced in geo-localization is that two images in the same geographic cell can share very few visual similarities. Two images from the same location could be taken at night or during the day, in sunny or rainy weather, or simply from the same location but one image faces North while the other faces South. Additionally, some information in a scene can be relevant to one geographic hierarchy (e.g. state) but not another (e.g. country). To that end, we propose a novel decoder-based architecture designed to learn unique sets of features for each of these possible settings. We begin by defining our geographic queries as $GQ \in \mathbb{R}^{HS \times D}$ where H is the number of geographic hierarchies, S is the number of scene labels, and D is the dimension of the features. We define each individual geographic query as gq_s^h where h and s represent the index of the hierarchy and scene, respectively. The scene labels we use are provided by *Places2* dataset [23]. We implement a pre-trained scene classification model to get the initial scene label from the coarsest set of labels and finer labels are extracted using their hierarchical structure. We find that the middle set of 16 scenes gives the best results for our model, we show ablation on this in supplementary material.

3.3. GeoDecoder

Hierarchy Independent Decoder The geographic queries are passed into our GeoDecoder, whose primary function is, for each hierarchical query, to extract geographical information relevant to its individual task for the image tokens which have been produced by a Swin encoder [10]. As previously stated, our decoder performs operations on a series of learned latent arrays called *Geographic Queries* in a manner inspired by the Perceiver [6] and DETR [1]. We define X as the image tokens, GQ^k as the geographic queries at the k^{th} layer of the decoder. Each layer performs multi-head self-attention (MSA) on the layer-normalized (LN) geographic queries, followed by cross-attention between the output of self-attention and the image patch encodings, where cross-attention is defined as $CA(Q, K) = softmax(\frac{QK^T}{\sqrt{d_k}})K$. where Q, K are Query and Key respectively. Finally, we normalize the output of the cross-attention operation and feed it into an feed-forward network (FFN) to produce the output of the decoder layer. Therefore, one decoder layer is defined as

$$y^{SA} = MSA(LN(GQ^{k-1})) + GQ^{k-1}. \quad (1)$$

$$y^{CA} = CA(LN(y^{SA}, LN(X)) + y^{SA}, \quad (2)$$

$$GQ^k = FFN(LN(y^{CA})) + y^{CA}. \quad (3)$$

Hierarchy Dependent Decoder

We find that a traditional transformer decoder structure for the entire GeoDecoder results in a homogeneity of all hierarchical queries. Therefore, in the final layers of the decoder, we perform self-attention only in an *intra* hierarchical manner rather than between all hierarchical queries. Additionally, we assign each hierarchy its own feed-forward network at the end of each layer rather than allowing hierarchies to share one network. We define the set of geographic queries *specifically* for hierarchy h at layer k as GQ_h^k. The feed-forward network for hierarchy h is referred to as FFN_h

$$y^{SA} = MSA(LN(GQ_h^{k-1})) + GQ_h^{k-1}, \quad (4)$$

$$y^{CA} = CA(LN(y^{SA}), LN(X)) + y^{SA}, \quad (5)$$

$$GQ_h^k = FFN_h(LN(y^{CA})) + y^{CA}. \quad (6)$$

After each level, each GQ_h^k is concatenated to reform the full set of queries GQ. In the ablations Table 4, we show the results of these *hierarchy dependent layers*.

3.4. Losses

As shown in Figure 2, our network is trained with two losses. The first loss is scene prediction loss, L_{scene}, which is a Cross-Entropy loss between the predicated scene label $\hat{s_i}$ ground truth scene labels s_i. Our second loss is a geo-location prediction loss, L_{geo}, which is a combination of Cross-Entropy losses for each hierarchy. Given an image X we define the set of location labels as h_1, h_2, ..., h_7, where h_i denotes the ground-truth class distribution in hierarchy i, and the respective predicted distribution as $\hat{h_i}$, we define $L_{scene(X)} = CE(s_i, \hat{s_i})$ and $L_{geo}(X) = \sum_{i=1}^{7} CE(h_i, \hat{h_i})$ and $L(X) = L_{geo}(X) + L_{scene}(X)$.

3.5. Inference

With the output of our GeoDecoder GQ^{out} we can geolocalize the image. As our system is designed to learn different latent embeddings for different visual scenes, we must first choose which features to proceed with. For $gq_s^h \in GQ$ we assign the confidence that the image belongs to scene s to that vector's $0th$ element. This minimizes the need for an additional individual scene network like in [11], while allowing specific weights within the decoder's linear layers to specialize in differentiating visual scenes. Once we have GQ^{out}, the queries are separated and sent to the classifier that is assigned to their hierarchy. This gives us 7 different sets of class probabilities, one for each hierarchy. To condense this information into one class prediction, and to exploit the hierarchical nature of our classes, we multiply the probabilities of the classes in the coarser hierarchies by their sub-classes found in the finer hierarchies. If we define a class as $C_j^{H_i}$ where i denotes the hierarchy and j denotes the class label within that hierarchy, we can define the probability of predicting a class $C_a^{H_7}$ for image X as: $p(X|C_a^{H_7}) = p(X|C_a^{H_7}) * p(X|C_b^{H_6}) * ... * p(X|C_g^{H_1})$, given that $C_a^{H_7}$ is a subclass of $C_b^{H_6}$, $C_b^{H_6}$ is a subclass of $C_c^{H_5}$ and so on. We perform this for every class in our finest hierarchy so that we can use the finest geographic granularity while also using the information learned for all of the hierarchies.

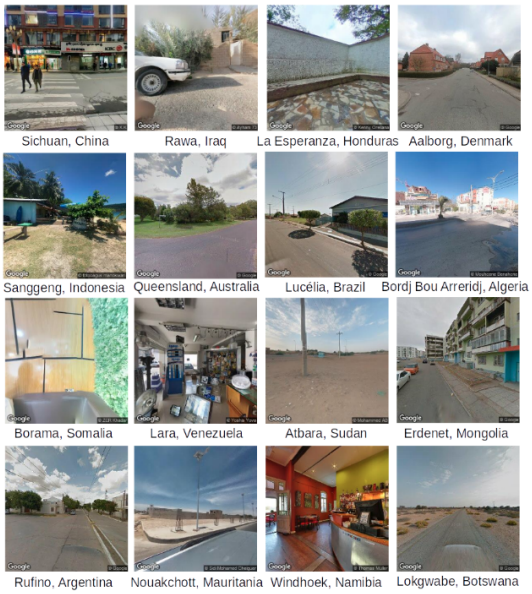


Figure 3. Example images from 16 different countries in the Google-World-Streets-15k dataset

4. Google-World-Streets-15K Dataset

We propose a new testing dataset collected using Google Streetview called Google-World-Streets-15k (see Figure 3 for some representative examples). As previous testing datasets contain biases towards commonly visited locations or landmarks, the goal of our dataset is to eliminate those biases and have a more even distribution across the Earth. In total, our dataset contains 14,955 images covering 193 countries.

In order to collect a fair distribution of images, we utilize a database of 43,000 cities[2], as well as the surface area of every country. We first sample a country with a probability proportional to its surface area compared to the Earth's total surface area. Then, we select a random city within that country and a GPS coordinate within a 5 Km radius of the center of the city to sample from the Google Streetview API. This ensures that the dataset is evenly distributed according to landmass and not biased towards the countries and locations that people post online. Google Streetview

[2]https://simplemaps.com/data/world-cities

also blurs out any faces found in the photos, so a model that is using people's faces to predict a location will have to rely on other features in the image to get a prediction.

In Figure 4 We show a heatmap of Google-World-Streets-15k compared to heatmaps of YFCC26k and Im2GPS3k. We note that a majority of YFCC26k and Im2GPS3k are located in North America and Europe, with very little representation in the other 4 populated continents. While Google-World-Streets-15k's densest areas are still the Northeastern US and Europe, we provide a much more even sampling of the Earth with images on all populated continents. We also note that the empty locations on our dataset's heatmap are mostly deserts, tundras, and mountain ranges.

5. Experiments

5.1. Training Data

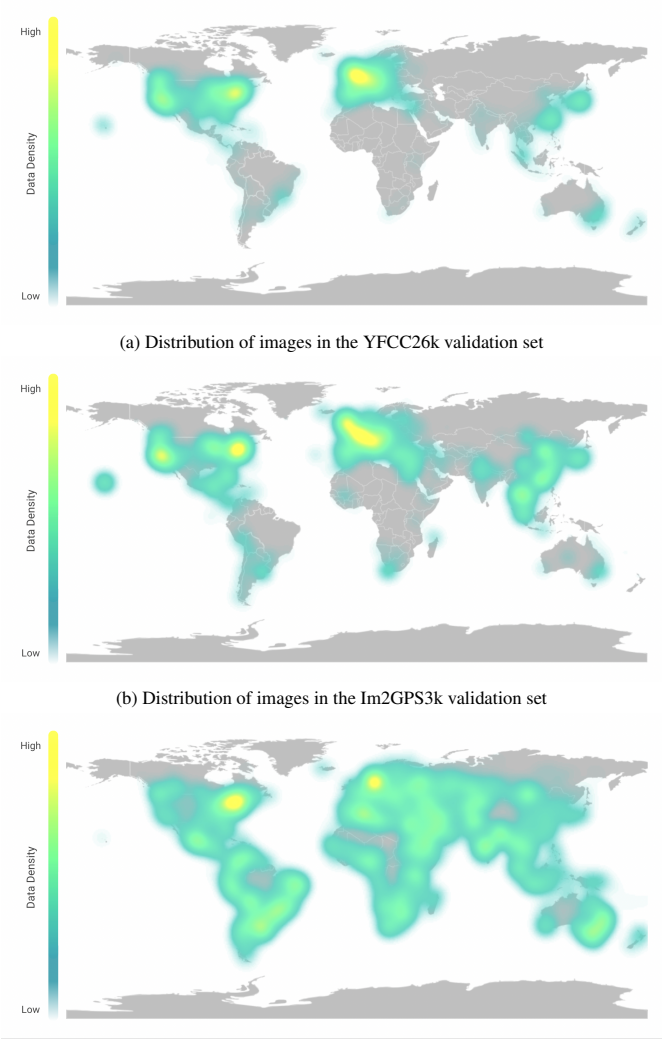
Our network is trained on the MediaEval Placing Tasks 2016 (MP-16) dataset [9]. This dataset consists of 4.72 million randomly chosen geo-tagged images from the Yahoo Flikr Creative Commons 100 Million (YFCC100M) [19] dataset. Notably, this subset is fully uncurated, and contains many examples that contain little if any geographic information. These photos include pets, food, and random household objects. We ensure that no photographer's images appear in both the testing and training sets, to guarantee that our model learns from visual geographic signals rather than the styles of individual photographers.

5.2. Testing Data

We test our method on five datasets: Im2GPS [4], Im2GPS3k [21], YFCC dataset: YFCC26k [18] YFCC 4k [21], and proposed new dataset Google-World-Street-15K described in the previous section. Im2GPS [4] and Im2GPS3k [21], contain 237 and 2997 images respectively. While small in size, both datasets are manually selected and contain popular sights and landmarks from around the world. We note that many of the landmarks that appear in Im2GPS appear multiple times in the MP-16 dataset, which may cause a bias towards those locations, this is accounted for in our proposed testing dataset. YFCC dataset: YFCC26k [18] and YFCC 4k [21], contain 25,600 and 4,536 images respectively. In contrast to Im2GPS and like our training set MP-16, these images are randomly selected and often contain very little geo-localizable information, and therefore pose a more difficult challenge than the Im2GPS datasets.

5.3. Evaluation

During evaluation we utilize the finest hierarchy class to get an image's predicted location. We report our accuracy at the street (1 Km), city (25 Km), region (200 Km), country (750 Km), and continent (2500 Km) scales. However, training on multiple hierarchies allows us to employ a parent-child relationship and multiply the probabilities across all hierarchies [11]. This allows the finest set of probabilities to be enhanced to include all of the learned hierarchical information. We also use TenCrop during evaluation, which is a cropping technique that returns the four corner crops, center crop, and their flipped versions. All crops are passed through the model and their outputs are averaged to get one set of probabilities per hierarchy for each image.



(a) Distribution of images in the YFCC26k validation set

(b) Distribution of images in the Im2GPS3k validation set

(c) Distribution of images in our Google World Streets 15k validation set

Figure 4. A comparison of YFCC26k, Im2GPS3k, and our Google World Streets 15k dataset. We see that popular datasets for testing geo-localization systems are heavily concentrated in heavily populated, metropolitan areas, particularly in America and western Europe. By contrast, our dataset more evenly blankets the earth, better representing all countries on earth.

Table 1. **Geo-localization accuracy of our proposed method compared to previous methods, across four baseline datasets, and our proposed dataset.** Results denoted with * are using our recreation of the given model.

Dataset	Method	Distance (a_r [%] @ km) Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS [4]	Human [21]	–	–	3.8	13.9	39.3
	[L]kNN, $\sigma = 4$ [21]	14.4	33.3	47.7	61.6	73.4
	MvMF [5]	8.4	32.6	39.4	57.2	80.2
	PlaNet [22]	8.4	24.5	37.6	53.6	71.3
	CPlaNet [15]	16.5	37.1	46.4	62.0	78.5
	ISNs (M, f, S_3) [11]	16.5	42.2	51.9	66.2	81.0
	ISNs (M,f*,S_3) [11]	16.9	43.0	51.9	66.7	80.2
	Translocator	19.9	48.1	64.6	75.6	86.7
	Ours	**22.1**	**50.2**	**69.0**	**80.0**	**89.1**
Im2GPS 3k [21]	[L]kNN, $\sigma = 4$ [21]	7.2	19.4	26.9	38.9	55.9
	PlaNet† [22]	8.5	24.8	34.3	48.4	64.6
	CPlaNet [15]	10.2	26.5	34.6	48.6	64.6
	ISNs (M, f, S_3) [11]	10.1	27.2	36.2	49.3	65.6
	ISNs (M,f*,S_3) [11]	10.5	28.0	36.6	49.7	66.0
	Translocator	11.8	31.1	**46.7**	58.9	**80.1**
	Ours	**12.8**	**33.5**	45.9	**61.0**	76.1
YFCC 4k [21]	[L]kNN, $\sigma = 4$ [21]	2.3	5.7	11.0	23.5	42.0
	PlaNet† [22]	5.6	14.3	22.2	36.4	55.8
	CPlaNet [15]	7.9	14.8	21.9	36.4	55.5
	ISNs (M, f, S_3)‡ [11]	6.5	16.2	23.8	37.4	55.0
	ISNs (M,f*,S_3)‡ [11]	6.7	16.5	24.2	37.5	54.9
	Translocator	8.4	18.6	27.0	41.1	60.4
	Ours	**10.3**	**24.4**	**33.9**	**50.0**	**68.7**
YFCC 26k [18]	PlaNet‡ [22]	4.4	11.0	16.9	28.5	47.7
	ISNs (M, f, S_3)‡ [11]	5.3	12.1	18.8	31.8	50.6
	ISNs (M, f*, S_3)‡ [11]	5.3	12.3	19.0	31.9	50.7
	Translocator	7.2	17.8	28.0	41.3	60.6
	Ours	**10.1**	**23.9**	**34.1**	**49.6**	**69.0**
GWS 15k	ISNs (M, f*, S_3)‡ [11]	0.05	0.6	4.2	15.5	38.5
	Translocator*	0.5	1.1	8.0	25.5	48.3
	Ours	**0.7**	**1.5**	**8.7**	**26.9**	**50.5**

6. Results, Discussions and Analysis

In this section, we compare the performance of our method with different baselines, and conduct a detailed ablation study to demonstrate the importance of different components in our system. Furthermore, we visualize the interpretability of our method by showing the attention map between each query and the image patches from our encoder.

Our results are presented in Table 1. On Im2GPS, our method achieves state of the art accuracy across all distances, improving by as much as 1.7% on the baseline. For Im2GPS3k our method manages to beat the previous techniques on a majority of distances, only falling short on the 200 and 2500 kilometer accuracies. More notably, our system's performance on the far more challenging YFCC4k and YFCC26k datasets vastly outperforms previous geo-localization works. On YFCC4k, our method achieves a score of 10.3%, an improvement of 2.2% over Translocator. Similarly on YFCC26k, we achieve a 1KM accuracy of 10.1%, improving over Translocator by 2.9%. Additionally, we compare our method to [12] on our Google-World-Streets-15k(GWS) validation dataset. As expected, the more realistic and fair nature of this dataset, in contrast to the training set MP-16, resulted in poor performance on all systems. However, we still outperform Translocator by 0.2% on 1KM accuracy and 0.4% on 25KM accuracy, suggesting a stronger ability to focus on defining features of a scene, rather than singular landmarks.

Table 2. **Ablation Study on GeoDecoder Depth** We find that larger depths offer marginal increases in performance, and there are diminishing returns for more than 8 layers.

Dataset	Depth	Distance (a_r [%] @ km) Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS3k [21]	3	11.9	32.9	45.0	59.5	75.4
	5	12.5	33.3	45.2	60.1	75.9
	8	**12.8**	**33.5**	**45.9**	**61.0**	76.1
	10	12.5	33.2	45.2	60.1	**76.2**
YFCC26k [18]	3	9.7	23.5	33.4	49.3	68.3
	5	9.9	23.6	33.8	49.6	68.5
	8	**10.1**	**23.9**	**34.1**	49.6	69.0
	10	10.0	23.7	33.6	**50.1**	**69.2**

Table 3. **Ablation Study on scene prediction method** We show our max score selection method of scene queries outperforms both scene prediction approach of [12], as well as treating scenes as an additional task.

Dataset	Method	Distance (a_r [%] @ km) Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS3k [21]	No Scene Prediction	11.7	31.5	42.3	57.0	72.3
	Scene Prediction [12]	12.2	32.8	44.3	59.5	75.8
	Ours	**12.8**	**33.5**	**45.9**	**61.0**	**76.1**
YFCC26k [18]	No Scene Prediction	9.4	22.9	32.6	48.0	65.4
	Scene Prediction [12]	9.7	23.2	33.0	48.8	67.0
	Ours	**10.1**	**23.9**	**34.1**	**49.6**	**69.0**

Table 4. **Ablation Study on Hierarchy Dependent Decoder** We show that converting the final two layers of the GeoDecoder to be hierarchy dependent layers offers marginal returns.

Dataset	Layers	Distance (a_r [%] @ km) Street 1 km	City 25 km	Region 200 km	Country 750 km	Continent 2500 km
Im2GPS 3k [21]	0	12.2	33.2	45.5	60.3	75.8
	2	**12.8**	**33.5**	**45.9**	**61.0**	**76.1**
	4	12.8	33.4	45.0	60.7	75.6
	6	12.6	33.2	44.5	59.9	75.3
YFCC26k [18]	0	9.7	23.5	33.8	49.2	68.7
	2	**10.1**	**23.9**	**34.1**	**49.6**	**69.0**
	4	9.9	23.4	33.6	49.0	68.3
	6	8.7	22.6	33.0	48.6	67.6

6.1. Qualitative Results

We provide a number of qualitative results, outlined in Figure 5. For our attention maps, we use the attention between the image backbone features and the fine-level query (corresponding to the correct scene). First, these results show that each hierarchy query attends to different parts of the image, as per our original hypothesis. Second, we can see that the attention for the *correct* scene query is far more precise than *incorrect* scene queries, demonstrating how our system learns different features specific to each scene.

6.2. Ablations

Ablation Study on Encoder Type We perform an ablations study on different image encoders. We show that our method outperforms using ViT or Swin on their own. See Table 5.

GeoDecoder Depth We perform two ablations on the

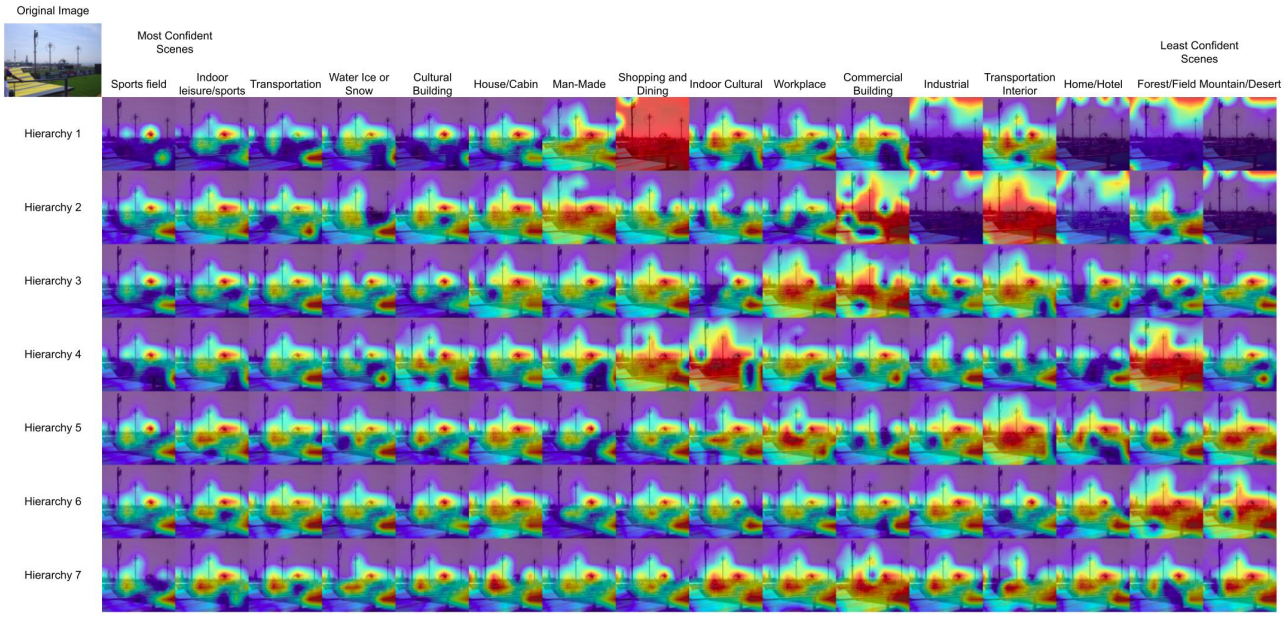


Figure 5. A qualitative analysis of different queries. Here we show the attention maps between every query our model produces when probed with the original Im2GPS3k image seen in the top left. Each row shows a hierarchy query for all scenes, while each column shows each scene query for all hierarchies. This specific query image is of an outdoor sports field. We observe that the most relevant scene labels were predicted as most confident and that their attention maps are more localized to specific features that would define a sports field. Looking at the less confident scenes, we see that the attention maps look at more general features or at random areas of the image. This is because those queries are trained to find features for their specific scenes. For example, the shopping and dining query will be looking for things like tables, chairs, or storefronts that aren't present in this query image, which is why we see the attention maps looking more generally at the image rather than looking at specific features.

Table 5. **Ablation Study on Encoder Type** We show our method performs better than simple image encoders.

Dataset	Model	Distance (a_r [%] @ km)				
		Street 1 **km**	**City** 25 **km**	**Region** 200 **km**	**Country** 750 **km**	**Continent** 2500 **km**
YFCC26k [18]	ViT	6.9	17.3	27.5	40.5	59.5
	Swin	9.6	22.3	33.6	48.0	67.5
	Ours (ViT)	8.7	21.4	31.6	47.8	66.2
	Ours (Swin)	**10.1**	**23.9**	**34.1**	**49.6**	**69.0**

architecture of the GeoDecoder. First, we experiment with the GeoDecoder's depth, varying it at $n = 3, 5, 8, 10$ (Table 2). We see a steady improvement from 3 through 8, but then a clear reduction in performance on all metrics at $n = 10$. This suggests a point of diminishing returns. Additionally, we experiment with the hierarchy dependent layers on the end of the GeoDecoder (Table 4). Recall, these layers restrict attention operations to queries within the same hierarchy, and utilize specialized feed-forward layers. For these experiments the total number of independent and dependent decoder layers remains static at 8, and we increase the number of dependent decoder layers from 0 to 6.

Scene Prediction One contribution of our method is our approach toward distinguishing between different visual scenes of the same location. To show the effectiveness of our separated scene queries, we ablate on scene prediction by evaluating performance with no scene prediction, as well as using scene prediction as a secondary task as in [12]. We then compare it to our scene prediction method. See (Table 3). We find that our scene queries selection method outperforms treating scenes as a secondary task by 0.6% and 0.4% on Im2GPS3k and YFCC26k, respectively.

Additional Ablations We perform additional ablations on the number of scenes and the number of hierarchies in the supplementary.

7. Conclusion

In this work, we reformulated visual geo-localization via the learning of multiple sets of geographic features. Given an RGB image of any location on planet earth, our system first learns a set of image features employing a SWIN encoder, then uses the GeoDecoder to extract hierarchy-specific features for each possible scene, choosing the most confident scene before prediction. Our proposed method improves over other geo-localization methods on multiple benchmarks, especially on uncurated datasets most similar to real-world use cases.

This work was supported by the US Army contract W911NF-2120192.

References

[1] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 4

[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[3] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014. 2

[4] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *2008 ieee conference on computer vision and pattern recognition*, pages 1–8. IEEE, 2008. 2, 6, 7

[5] Mike Izbicki, Evangelos E Papalexakis, and Vassilis J Tsotras. Exploiting the earth's spherical geometry to geolocate images. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 3–19. Springer, 2019. 7

[6] Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR, 2021. 4

[7] Sebastian Kalkowski, Christian Schulze, Andreas Dengel, and Damian Borth. Real-time analysis and visualization of the yfcc100m dataset. In *Proceedings of the 2015 workshop on community-organized multimodal mining: opportunities for novel solutions*, pages 25–30, 2015. 2

[8] Giorgos Kordopatis-Zilos, Panagiotis Galopoulos, Symeon Papadopoulos, and Ioannis Kompatsiaris. Leveraging efficientnet and contrastive learning for accurate global-scale location estimation. In *Proceedings of the 2021 International Conference on Multimedia Retrieval*, pages 155–163, 2021. 1, 3

[9] Martha Larson, Mohammad Soleymani, Guillaume Gravier, Bogdan Ionescu, and Gareth JF Jones. The benchmarking initiative for multimedia evaluation: Mediaeval 2016. *IEEE MultiMedia*, 24(1):93–96, 2017. 6

[10] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 4

[11] Eric Muller-Budack, Kader Pustu-Iren, and Ralph Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 563–579, 2018. 1, 2, 3, 5, 6, 7

[12] Shraman Pramanick, Ewa M Nowara, Joshua Gleason, Carlos D Castillo, and Rama Chellappa. Where in the world is this image? transformer-based geo-localization in the wild. *arXiv preprint arXiv:2204.13861*, 2022. 1, 2, 3, 7, 8

[13] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 3

[14] Krishna Regmi and Mubarak Shah. Bridging the domain gap for ground-to-aerial image matching. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 470–479, 2019. 1, 2

[15] Paul Hongsuck Seo, Tobias Weyand, Jack Sim, and Bohyung Han. Cplanet: Enhancing image geolocalization by combinatorial partitioning of maps. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 536–551, 2018. 1, 3, 7

[16] Yujiao Shi, Liu Liu, Xin Yu, and Hongdong Li. Spatial-aware feature aggregation for image based cross-view geo-localization. *Advances in Neural Information Processing Systems*, 32, 2019. 1, 2

[17] Yujiao Shi, Xin Yu, Dylan Campbell, and Hongdong Li. Where am i looking at? joint location and orientation estimation by cross-view matching. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4064–4072, 2020. 1, 2

[18] Jonas Theiner, Eric Müller-Budack, and Ralph Ewerth. Interpretable semantic photo geolocation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 750–760, 2022. 2, 6, 7, 8

[19] Bart Thomee, David A Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. Yfcc100m: The new data in multimedia research. *Communications of the ACM*, 59(2):64–73, 2016. 6

[20] Aysim Toker, Qunjie Zhou, Maxim Maximov, and Laura Leal-Taixé. Coming down to earth: Satellite-to-street view synthesis for geo-localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6488–6497, 2021. 1, 2

[21] Nam Vo, Nathan Jacobs, and James Hays. Revisiting im2gps in the deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 2621–2630, 2017. 1, 2, 3, 6, 7

[22] Tobias Weyand, Ilya Kostrikov, and James Philbin. Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer, 2016. 1, 2, 7

[23] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1452–1464, 2017. 4

[24] Sijie Zhu, Mubarak Shah, and Chen Chen. Transgeo: Transformer is all you need for cross-view image geo-localization. *arXiv preprint arXiv:2204.00097*, 2022. 1, 2

[25] Sijie Zhu, Taojiannan Yang, and Chen Chen. Vigor: Cross-view image geo-localization beyond one-to-one retrieval. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3640–3649, 2021. 1, 2