Using Geoembeddings to Predict Image Geolocations

Kenneth Ma
Stanford University
Stanford, CA
kenma25@stanford.edu

Parker Stewart
Stanford University
Stanford, CA
parkers@stanford.edu

Wesley Tjangnaka
Stanford University
Stanford, CA
wesleytj@stanford.edu

Abstract

Predicting the geolocations of street-view images is a difficult problem because of the diverse set of images that originate from across the world. We attempt to tackle this challenge by approaching it as a classification problem and defining geocells and intra-geocell clusters as geographic classes. Based on the architecture outlined by PIGEON, the current state-of-the-art work in the area of geolocation, we utilize a pre-trained Vision Transformer (ViT) model to generate a geo-embedding that characterizes each of these clusters. To evaluate loss, we employ the commonly used Haversine loss function and add a continuous implementation of supervised contrastive loss, which promotes the model to push apart the embeddings of distant images. We aim to see if the implementation of contrastive loss makes a significant impact on performance, finding that it slightly improves the rate at which the model classifies an image within 50km of the target location. These results are also leagues more accurate than both our human baseline as well as a ResNet-152 model with a neural network head. Our findings suggest that the use of contrastive loss may provide a slight advantage because it enhances the model's ability to learn meaningful and discriminative feature representations. We hope that this loss concept can be incorporated in future geolocation prediction models.

1. Introduction

The ability to globally geolocate images that lack specific location metadata remains a challenging problem, even for human experts. Despite these difficulties, geolocation of images still remains an interesting and relevant task with a myriad of applications, ranging from locating the origins of photographs in photo albums, identifying locations of crime scenes from criminal photographs, and the trending online video game GeoGuessr, where players compete to guess the precise location of Google Street View images. When we focus on the techniques applied today in geolocation, individuals hone in on certain details, such as species of fauna and flora, geological features, and architectural differences which provide clues to an image's exact location; however, it still remains difficult to predict the exact location from which a photograph originated just relying on these larger image characteristics. With the breakthroughs in deep learning applied to the field of computer vision, stronger and more advanced model architectures have been used to continually improve image classifiers by analyzing them on a more granular scale. This project will explore the usage of image encoders applied to the context of image geolocation to generate geoembeddings, with the goal of strengthening performance through identifying and highlighting subtle details that may be missed by human reviewers, but captured through these deep models.

1.1. Related Works

Although deep neural networks have long been used in the task of image classification, a major breakthrough in the use of computer vision for image classification came with the introduction of residual neural networks (ResNets), first discussed in Deep Residual Learning for Image Recognition (He et al., 2015) [7]. While testing the hypothesis of improving models by stacking more and more layers, researchers noticed that increased depth caused a problem of degradation when the accuracy gets saturated by so many layers. By introducing shortcuts between layers, this allowed for gradients to be backpropogated through the model more easily, yielding stronger models even when it contained a large number of layers. These ResNets significantly outperformed deep convolutional neural networks with image classification, and became one of the strongest architectures used for image classification. Despite its success, ResNet inherently suffers from limitations in capturing global context due to its localized convolution operation.

Since then, the field of image classification has significantly progressed. Vision Transformers (ViTs), first introduced in An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale (Dosovitskiy et. al., 2020) [4], employed self-attention mechanisms to process image patches as sequences. This approach allows ViTs to capture long-range dependencies more effectively, thereby providing a richer understanding of the image context. Contrastive Language-Image Pretraining (CLIP), proposed by Radford et al. (2021) in Learning Transferable Visual Models From Natural Language Supervision [8], made further improvements by leveraging the advantages of transformer-based models and integrating them with a contrastive learning framework. Not only do CLIP models feature global context understanding and multi-modal learning, but they also provide better scalability and more efficient transfer learning compared to previous models. Taking advantage of ViTs ability to efficiently scale, several pre-trained models of various sizes have emerged from CLIP, including StreetCLIP, which used the proprietary dataset of OpenAI.

On another track, the field of study in the geolocation of images has been always been a relevant application of computer vision that has been continually explored. For instance, one of the first known large scale studies discoverying how to effectively geolocate images was introduced in 2008 with IM2GPS (Hays et al., 2008) [6]. In this study, they used many simple data-driven features of images, such as line features, color histograms, and more in order to find similarity between images and be able to correspond them to nearest neighbors in their dataset. Though this was an early attempt, this method is not as effective because it just looks at raw image data without doing any complex analysis into it, and rather just looks at statistical occurrences. In fact, most of these methods were limited to nearest-neighbor approaches after a few preprocessing steps that extracted simple features using methods such as Scale Invariant Feature Transform (SIFT) (Zamir et al., 2014) [10]. However, nearest-neighbor approaches are heavily dependent on what exists in the dataset, and it will have significant issues generalizing to unseen data. As deep learning advanced more, it began to be applied significantly to this area of image geolocation. A notable paper that showed early success was IM2City, where the researchers leveraged multi-modal learning in order to not only learn image features, but also add in understanding of their labels and captions through natural language processing in order to add to its predictions (Wu et al., 2022) [9].

Concurrently, further advancement has occurred towards our intended task of geolocating images. For instance, with respect to training these models for geolocation, in Rethinking Visual Geo-localization for Large-Scale Applications (Berton et al., 2022), the authors introduce CosPlace, a training method for the task of geolocating images that is stronger and more efficient when the dataset sizes become increasingly large [3]. Some tasks that come with geolocating image data, such as visual assistance or autonomous vehicles, require the analysis of large and extremely detailed datasets. By converting the problem of geolocating images into a classification problem followed by an image retrieval related action, both training and inference costs of geolocation dropped significantly, and allowed these models to be trained on much larger datasets which strongly increased their accuracy as well. Moreover, the current state of the art work in the field includes the two models PIGEON and PIGEOTTO, introduced in PIGEON: Predicting Image Geolocations (Haas et. al, 2024) [5]. The only difference between these models is the source of their training data; while PIGEON is trained on the game of GeoGuessr (which are Google Street View images), PIGEOTTO is trained on images found on Flickr and Wikipedia. In these models, they are built as classification problems where the images are classified into clusters of geocells, which can be thought of different ”classes” or regions or Earth. They introduce a new loss function, Haversine loss, which is calculated by the distance between two coordinate points on Earth's spherical surface. On top of that, the authors add OpenAI's CLIP model to provide synthetic captions of an image, discussing details like weather, compass direction, traffic, and more parts of the image that could provide clues to the photo's location. By combining all of these details preoprocessed through CLIP, PIGEON and PIGEOTTO are able to classify the image into geocells based on CLIP's pretrained embeddings, which were shown to have strong zero-shot performance on image classification. This classification occurs on 4 layers of granularity, from general location on Earth's surface, to a specific spot in a town, predicting the coordinate where the image was taken.

2. Data

The dataset we are using to train and evaluate our model is the OpenStreetView-5M dataset [2], which provides over 5.1 million images sourced from over 225 countries and territories around the world. These images are paired with a corresponding longitude and latitude coordinate, pinpointing exactly where the image was taken. Due to the high training time associated with training a deep model on such a large amount of images, we randomly sampled 500,000 of the original 5.1 million images to be the dataset to be used throughout our experiments. We performed random sampling in order to get a dataset that is representative of the images around the world, minimizing the risk of accidentally creating a dataset that is focused on a small subset of the countries and regions within the original data. Within the dataset, we designate 80% of the images as our training set, and 10% each for validation and testing set.

After collecting our dataset, we performed some preprocessing and data manipulation tasks in order to standardize the images to the same format. Specifically, we changed the dimensions of each image to be a 224 x 224 image to be fed into our model. We also performed a secondary task to normalize each picture into the means and standard deviations of the images in the ImageNet dataset in order to effectively extract features using Vision Transformer [4] models.

3. Methods

Our approach is based on the overall model architecture from PIGEON and PIGEOTTO [5], especially since we believe that the geocell implementation would be the strongest way to iteratively determine a precise location from larger general regions. The main difference between our implementation and the one introduced by PIGEON is the use of a vision transformer over the use of CLIP, which necessitates text captions on top of the images, which is not readily availble in our current OSV5m dataset.

3.1. Baselines

To measure the success of our model, we will compare with multiple baselines, the first of which being a human benchmark by testing on the interface provided on the OpenStreetView-5M dataset's HuggingFace page (https://huggingface.co/spaces/osv5m/plonk). The motivation behind having a human baseline is because one major application of this project is into the game GeoGuessr, and one way to test our model's success is how it compares to a human player on the game. Since a human identifying the location of an image is limited to only looking at simpler features, such as signs on the road, or species of plants and wildlife in the image, we want to compare how our model does when it looks at the image on a more granular level.

Furthermore, another baseline we implemented and want to compare our final model to is to a pretrained RestNet-152 [7] model with a neural network head. Since the ResNet was a former state-of-the-art model for image classification, the ResNet would also serve as a fair baseline for the task of geolocating images. Since the ResNet was not built for completely the same task, as the original implementation of a ResNet was a pure classification task, and we are comparing it to a model identifying longitude and latitude coordinates, we modified the head of the ResNet model with a neural network in order for it to output coordinates rather than classes. However, the tasks are similar enough to allow us to use the feature extraction capabilities of the pretrained ResNet-152 for downstream prediction.

3.2. Loss

Haversine Loss: Similar to the technique introduced by PIGEON and PIGEOTTO, the loss function we use is the Haversine loss function, which is derived from Haversine distance. Haversine distance measures the distance between two points on a sphere given their longitudes and latitudes, and this is relevant in our project as we are attempting to measure the distance between the user's guess and the actual coordinates of the location. The formula for Haversine distance is as follows:

$$d = 2r \arcsin\left(\sqrt{\frac{1-\cos(\varphi_2-\varphi_1)+\cos\varphi_1\cdot\cos\varphi_2\cdot(1-\cos(\lambda_2-\lambda_1))}{2}}\right)$$

where r is the radius of the sphere, φ_1, φ_2 are the latitudes of point 1 and point 2, and λ_1, λ_2 are the longitudes of point 1 and point 2.

As a loss function, this can also be interpreted as the model's accuracy in how close or far from the true location the model guesses.

Contrastive Loss: In addition to Haversine loss, we experiment with the implementation of contrastive loss, which attempts to minimize the distance between similar examples, and maximize the distance between dissimilar ones. Given m positive examples and $N-m$ negative examples, the loss function can be defined as:

$$L = -\mathbb{E}_X[\log \frac{score(+)}{score(+) + score(-)}],$$

where $score(+) = \sum_{i=1}^{m} \exp(s(f(x), f(x_i^+)))$ is the sum of the scores for the m positive pairs, and $score(-) = \sum_{j=1}^{N-m} \exp(s(f(x), f(x_i^-)))$ is the sum of the scores the $N-m$ negative pairs. Here, our score function $s(,)$ is the Haversine distance between the two examples. We experiment with the weighting scale of these two loss values.

In our case, we implement contrastive loss slightly differently - when we have examples close to each other geographically, we want to push their embeddings closer together, and when we have further examples geographically, we puth their embeddings further apart.

3.3. Geocells and Intra-geocell Clusters

As with many contemporary methods, we approach Image Geolocation as a classification problem. We utilize a set of pre-computed geocells and intra-geocell clusters to discretize the Earth's surface into a pre-set number of classes. This strategy aligns with the notion that clusters will naturally arise by nature of geographical terrain and urban development. Each cluster should represent a meaningful geographic region and have relatively balanced sizes. This method was deeply expanded upon by the PIGEON paper [5], and we wanted to build our models upon this iterative method of pinpointing a location as we felt it was the most accurate way to find location on the globe. Though we originally had a plan of building a model to output longitude and latitude values, there would have been too much variance (as evidenced by the results of the baseline), and the geocell method would be significantly more accurate.

Clusters and Centroids: Firstly, using training image metadata (image longitude and latitude) and the OPTICS clustering algorithm (Ankerst et. al., 1999) [1], we compute a set of initial clusters, using Silhouette scores and the

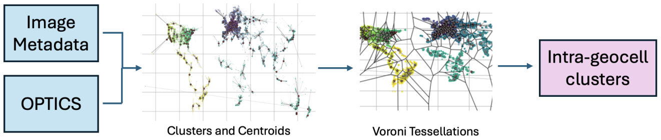

Figure 1. Geocell and Cluster computation pipeline.

Davies-Bouldin index to evaluate the quality of our clustering and determine the best hyperparameters for our data. A total of 174 clusters are created. Note that these preliminary clusters are relatively large, omit certain unassigned outlier data points. Next, we compute the centroids of each cluster and assign each unassigned data point to its closest centroid, measuring distance using the previously defined Haversine distance function. Once all data points are properly assigned, we recompute the centroids to account for the newly assigned points. Figure 2 depicts the results of our clustering operations.

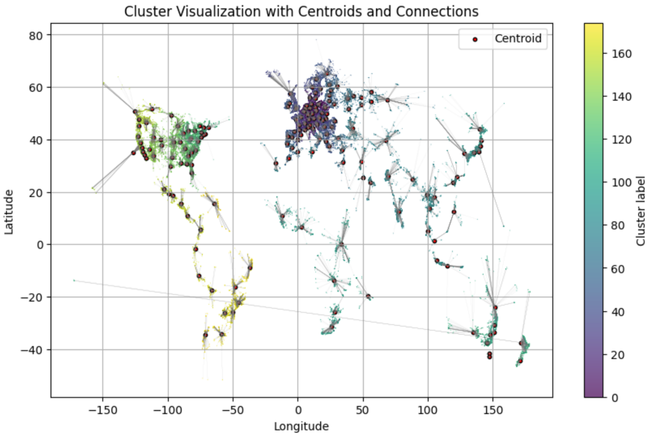

Figure 2. Visualization of centroids and connections. We see that each cluster is geographically coherent, and that the cluster placesments somewhat represent the shapes of the earth's land masses, as expected. There are also more clusters in more image-dense areas, as expected.

Voronoi Tessellations: Once our clusters are created, we use Voronoi Tessellation to define a contiguous set of clusterings by partitioning the plane into regions close to each of the given clusters. Because normal Voronoi Tessellation does not account for the curvature of the Earth, we use an adapted version to ensure that points on the edges are properly integrated. This new set of clusterings is our set of geocells, as depicted in Figure 3.

Intra-geocell clusters: These generated geocells are relatively large, ranging in area between 10 and 4,000 square kilometers. We want to refine our clustering to accurately capture the desired geographic classifications. To do so, we re-cluster data points in each tessellation using the same process, generating a set of over 2,700 total intra-geocell clusters, with 5 to 100 clusters per geocell.

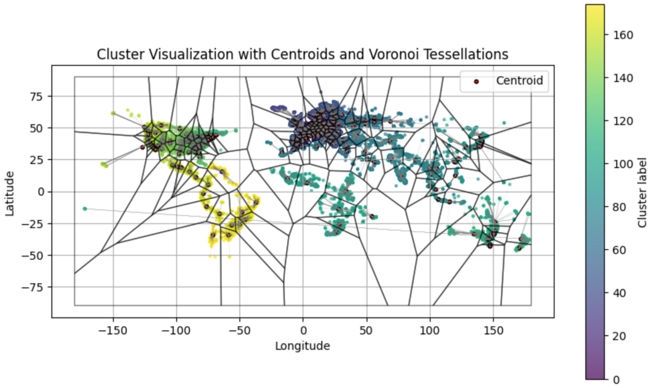

Figure 3. Visualization of geocells after Voronoi Tessellation. Now, every location on Earth is accounted for by one of the tessellations, creating a set of classes that encompass every possible point.

3.4. Creating Geo-Embeddings

Many approaches to this problem have demonstrated that vision encoder models, such as ResNet, ViT, and CLIP, are extremely powerful at extracting the important features on an image for geo-location use. We employ a pre-trained Vision Transformer (ViT) [4] model to create geo-encodings of given images. We selected ViT because of its self-attention mechanism, which allows it to understand and encode global context more effectively, and ability to learn complex and high-level features that may be more relevant for distinguishing between different geographic locations. Additionally, ViT is compatible with image-only data, without the need for captions, which aligns with our OSV5M dataset. On top of these encodings, we add a linear layer to learn a geo-embedding for each geocell as well as each intra-geocell cluster.

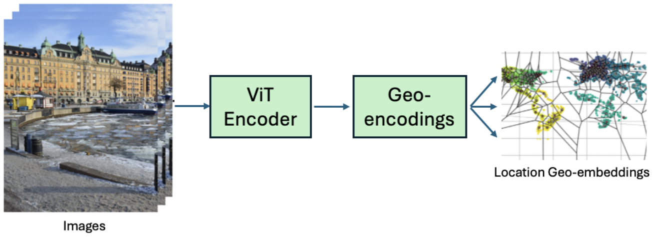

Figure 4. Visualization of geocells after Voronoi Tessellation. Now, every location on Earth is accounted for by one of the tessellations, creating a set of classes that encompass every possible point (image source: Haas et al., 2024 [5])

3.5. Location Cluster Retrieval

During inference time, we select the top k geocells that most closely match the geo-encoding of a given image. This allows for diversity of location to account for the possibility of two geocells on opposite sides of the Earth having similar geo-embeddings. The model then selects the cluster with the closest geo-encoding, and outputs the coordinates of the centroid of the cluster. We can safely expect correct outputs to be within 50km of the actual image location.

4. Experiments

For our experiments, we compare the results of the our ViT model with the various baselines proposed. We modify the hyperparameters of the ViT model, specifically whether or not to use contrastive loss, different levels of contrastive weight (α), as well as different values of temperature (τ). We trained the ViT model with a learning rate $\eta = 1 \times 10^{-7}$, and the prediction head was trained with a learning rate $\eta = 1 \times 10^{-3}$. We trained each model with 10 epochs, and used the AdamW optimizer. We did an $80 - 10 - 10\%$ split of our data for training, validation, and testing, with a total of 100,000 images used for the training. For the computation and clustering of the geocells, we used a separate 100,000 images from the same dataset in order to generate this. The embeddings for the geocells were also generated using the ViT transformer model.

4.1. Baseline

For the human baseline, the authors and 5 friends (sample size of $n = 8$) played the interactive demonstration on the HuggingFace webpage, and recorded how far their guesses were from the actual location of the image. After completing all 50 images, the distance was averaged and that was the final score kept for each player. At the end, the score for each player was again averaged across the 8 individuals to give the final score for the human baseline.

The neural network head (and final two layers of ResNet-152) was trained on the same image dataset that was used to train the final model. It is trained for 5 epochs, with the hyperparameters including a learning rate of 1×10^{-3}, weight decay of 1×10^{-4}, batch size of 32, as well as using the AdamW optimizer. After training is complete, it is evaluated on the same 50 images testing set, and the average distance away from the true points is kept as its score.

4.2. Models

The main pretrained model we used in order to compute the embeddings of the pictures was the ViT transformer model. The benefits of the ViT model was that it is stronger than the ResNet architecture due to it having self-attention, allowing it to capture the relation between different objects in the image and ultimately giving it a stronger embedding for us to find its location from. Unlike the PIGEON model where we based our models on, we didn't use a CLIP model that combines synthetic image captions with images learn embeddings, we believed that using ViT would allow us to focus more on image features and not require the use of external data.

We experimented with multiple configurations of our vision encoders in order to determine what would yield the strongest accuracy when predicting the correct geocells that the images were taken in. Specifically, we tested between training with or without the use of contrastive loss, and if it is being used, then we also modify how much we want to weight the contrastive loss in our overall loss calculation. Not only that, we had another hyperparameter of temperature. used in the contrastive loss in order to determine how much we "soften" or "harden" the final probability distribution.

4.3. Results

Model	Avg. Dist. (km)
Human Baseline	1,219.8
ResNet (Non-Clustering)	9,328.7
ViT + Haversine Loss	208.6
ViT + contrastive loss ($\tau = 0.3, \alpha = 0.5$)	173.3
ViT + contrastive loss ($\tau = 0.5, \alpha = 0.5$)	167.2
ViT + contrastive loss ($\tau = 0.5, \alpha = 0.75$)	185.2

Table 1. Results of experiments

As we can see in Table 1, initially, the most significant observation is that the human baseline vastly outperformed the ResNet implementation. One possibility is that the ResNet model was originally a classification model, not one that outputs a continuous range. As a result, if the ResNet incorrectly classifies anything, the output could be entirely off (consider an example where the ResNet guessed that a picture was taken in the United States when it was actually in London - it would have a really high Haversine loss). Since our model will be designed to output a continuous range of longitude and latitude points, it should easily be able to outperform the ResNet baseline, as when it is incorrect, it will more likely be incorrect by a smaller margin since it will output closer coordinates.

However, beyond the baselines, we see that the implementation of ViT is significantly stronger than both the human and ResNet baselines, by a factor of around 50 times better than ResNet, and 7 times better than human performance. The incorporation of contrastive loss further improves performance by a factor of approximately 1.2 times. However, with the variation of all hyperparameters, we see that contrastive loss gives similar results with very marginal differences.

4.4. Discussion

First of all, the results of the non-clustering algorithms do not perform as well as the geocell clustering, and this makes sense due to the iterative nature of determining an exact location using geocells. By slowly looking from a set of larger geocells to smaller clusters, we are able to zoom in step by step to determine an exact location. However, without the use of geocells, the model essentially just outputs a random longitude and latitude coordinate. As a result, if the model was wrong, it could have completely missed the general area of where it should have been predicting in. When the geocells predict incorrectly, at least it is resultant of smaller and smaller regions of searching, allowing for smaller error.

Moreover, we see that introducing a contrastive loss component to the often used Haversine loss model allows for further improvement. This intuitively makes sense, as contrastive loss allows for the model to further learn the difference between positive and negative examples per class, which reduces the chances that the model classifies the images in the wrong geocell.

When examining the hyperparameters of the contrastive loss model, we notice that there aren't significant differences between the different hyperparameters, but there are important differences to note. With higher levels of τ, the model seems to perform stronger because it makes the probability distributions softer through normalization. As a result, the model is able to generalize better to unseen examples because the probability distributions it generates doesn't overly favor certain examples, which lead to more instances of false predictions. With a level of $\alpha \approx 0.5$, the model seems to do a better job of balancing between the regular loss and contrastive loss, which shows for the better results. When we increase the contrastive weight too much, then the model performance seems to slip a bit.

5. Conclusion

Through the course of this project, we explored the application of ViT models in geolocating images. Instead of looking at individual featuers of the image and comparing them with other images in a nearest neighbors approach like that was done in early works in geolocation, we primarily approached the task as a classification problem (similar to the work done by the authors of PIGEON) and built upon the use of geocells and classifying images into geocells based on their embeddings from vision transformers. Overall, the results demonstrated that the vision transformers, when compared with human baselines or simple ResNet architectures, had stronger results, but still falls short against state of the art models like PIGEON.

Within the use of vision transformers, we compared the differences between the use of contrastive loss, and without contrastive loss, during the training step of these transformers. Using contrastive loss demonstrated a slight improvement in the overall accuracy of the model, primarily due to it's capacity to distinguish between correct and incorrect examples and be able to generalize better than its non-contrastive counterpart. We also find that with higher temperature, the results are a more normalized which gives stronger results than lower levels of temperature. Finally, we find that weighting the contrastive loss generally less allows for better results, which indicates that it is helpful in the accuracy but Haversine distance already has a strong effect.

Overall, contrastive learning techniques are quite strong in geolocation tasks just because there are many extremely similar images, and the model's ability to distinguish between subtleties differentiates a good model from a great one. When you think about geolocating images with extremely similar features, such as a beach, the contrastive learning allows us to differentiate beaches from opposite ends of the globe despite them almost all having the same exact sand, ocean, and sky features.

5.1. Future Work

To improve on our results, we could explore different variations of the pretrained model we fed into computing the embeddings for the image to be fed into the geocells. Not only that, we could have experimented with multimodal data - the original PIGEON paper discussed adding synthetic captions to supplement the model in order to make the embeddings stronger. With a stronger dataset, we could also explore the use of image metadata. With the knowledge of the time the picture was taken, the direction the user was facing during the image, and maybe a panoramic view of the picture, our model could yield significantly stronger results as well.

Finally, a bias that needs to be acknowledged within the current datasets that were used in this experiment was that the world wasn't represented proportionally in the dataset - for example, there were regions of the world significantly unrepresented. Though this could be a result of the fact that some places naturally have lower population density, our model underperforms on places where there aren't as many images from just because the geocells are generated from images existing in the training set. To make our model more generalizable, it is also important that we have a strong dataset that properly represents different areas of the world equally.

References

[1] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: Ordering points to identify the clustering structure, 1999. 3

[2] G. Astruc, N. Dufour, I. Siglidis, C. Aronssohn, N. Bouia, S. Fu, R. Loiseau, V. N. Nguyen, C. Raude, E. Vincent, L. XU, H. Zhou, and L. Landrieu. Openstreetview-5m: The many roads to global visual geolocation, 2024. 2

[3] G. Berton, C. Masone, and B. Caputo. Rethinking visual geo-localization for large-scale applications, 2022. 2

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer,

G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. 1, 3, 4

[5] L. Haas, M. Skreta, S. Alberti, and C. Finn. Pigeon: Predicting image geolocations, 2024. 2, 3, 4

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

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition, 2015. 1, 3

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. Learning transferable visual models from natural language supervision, 2021. 2

[9] M. Wu and Q. Huang. Im2city: image geo-localization via multi-modal learning. In D. D. Lunga and S. D. Newsam, editors, *Proceedings of the 5th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery, GeoAI 2022, Seattle, Washington, 1 November 2022*, pages 50–61. ACM, 2022. 2

[10] A. R. Zamir and M. Shah. Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(8):1546–1558, 2014. 2