# Predicting Image Geolocation Using Feature-Based Fine-Tuning

Kai Qi (Jennifer) Wu
Stanford University
jwkaiqi@stanford.edu

## Abstract

*This paper investigates deep learning approaches for image-based geolocation, the task of predicting a location's latitude and longitude from a single image. A baseline was established using a ResNet50 convolutional neural network adapted for regression and pre-trained on ImageNet. The ResNet50 was subsequently replaced with a Vision Transformer (ViT) backbone, trained with Supervised Contrastive Loss to better capture global visual cues relevant to geography. The contrastive supervision is based on S2 cell IDs, which treat images within the same geographic cell as a single class. Building on this, a novel two-stage ViT architecture was developed. The first stage classifies an image into a coarse geographic region using Supervised Contrastive learning based on S2 cells. The second stage refines the prediction by training with Triplet Margin Loss.*

*Models were trained on the OpenStreetView-5M dataset [1] and evaluated on its test split. Experimental results show a clear progression, with the ViT model significantly outperforming the ResNet50 baseline and the two-stage ViT further showed mixed results, which improved precision for continuous landscapes like Europe but underperformed for regions such as North America. This research highlights the effectiveness and ineffectiveness of a coarse-to-fine learning strategy and demonstrates the potential of transformer-based architectures for global-scale image geolocation.*

## 1. Introduction

This project aims to develop a deep learning model that predicts the geographic location of a scene from a single input image. This task, known as image-based geolocation, holds practical importance for environmental monitoring, cultural heritage preservation, and interactive mapping.

The input of the models is a single RGB image from the OpenStreetView-5M dataset [1], depicting a street-level scene. The output is a predicted pair of geographic coordinates. To address this problem, the modeling approach was progressively refined through four architectures:

1. **ResNet50 CNN** [14]: A baseline was established using a ResNet50 Convolutional Neural Network (CNN) for regression [12].

2. **Vision Transformer (ViT) [13]**: To better capture long-range spatial relationships, the CNN was replaced with a Vision Transformer, which processes images in a way that captures global context more effectively.

3. **Two-Stage ViT**: A novel two-stage architecture was designed to decompose the problem into a coarse classification task followed by fine-grained regression based on distances, leading to a substantial boost in geolocation accuracy.

4. **Multi-Stage ViT**: Aim to address limited VRAM while still leveraging the strengths of supervised contrastive loss, which is known to perform better with larger batch sizes. Due to its exploratory nature, the details are provided in the Appendix. Despite its experimental status, this model achieved the best inference performance among all the models listed here.

Experimental results show a clear progression, with the ViT model significantly outperforming the ResNet50 baseline and the two-stage ViT further showed mixed results compared with the single-stage ViT, which improved precision for continuous landscapes like Europe but underperformed for regions such as North America.

## 2. Related Work

Image geolocation has been approached from several angles in the past.

### 2.1. Image Retrieval and Matching

Early work like IM2GPS [2] pioneered geolocation by using k-nearest-neighbor search on hand-crafted visual features across a large, geotagged database. Subsequent research refined this retrieval-based strategy but remained limited by the need for dense reference imagery and computationally expensive matching. These scalability issues prompted a shift toward more constrained settings, such as specific cities [6] or landscapes [9], highlighting the limits of retrieval-based methods at a global scale.

### 2.2. Classification-Based Approaches

PlaNet [3] framed geolocation as a classification problem by dividing the Earth's surface into thousands of discrete S2 cells [15] and training a CNN to predict the correct cell for an image. This approach achieved significant improvements over retrieval methods but was limited by its coarse granularity, as it could not provide precise coordinates within a cell. Its reliance on a fixed grid also led to inconsistent performance across regions with varying data density.

### 2.3. Transformer-Based Geolocation

A major advancement came with PIGEON [4], which uses a pretrained Vision Transformer (ViT) to map images to locations on a 3D globe. By predicting a probability distribution over hierarchical GeoCells, PIGEON enables coarse-to-fine localization and can estimate continuous coordinates. It achieves state-of-the-art performance, capable of placing over 40% of its predictions within 25 km of the target.

## 3. Methods

### 3.1. S2 Cell Partitioning

To structure the geolocation task for supervised contrastive learning, the OpenStreetView-5M dataset [1] was partitioned using Google's S2 geometry library [15]. This library divides the Earth's surface into a quadtree of hierarchical, non-overlapping cells. S2 cell level 6 was used when training on 500,000 images, and level 7 was used for the one million image dataset, balancing spatial granularity with data density. Each image was assigned a class label corresponding to its S2 cell (with levels ranging from 3 to 14), providing the discrete labels required for contrastive learning. This information was saved as a CSV file for use during training, See Figure 1.

### 3.2. Model Architectures

The initial approach used a pretrained ResNet50 model as a feature extraction backbone. The final fully-connected layer was replaced with an identity layer, and a new two-layer MLP projection head was attached to produce a normalized 256-dimensional embedding vector for each image. For regularization, a ReLU activation and a dropout layer (rate=0.3) were included.The model was trained using Supervised Contrastive Loss, which encourages embeddings of images from the same S2 cell (positive pairs) to be closer than embeddings from different S2 cells (negative pairs). During training, layers 3 and 4 of the ResNet50 backbone were unfrozen, resulting in approximately 24 million learnable parameters.

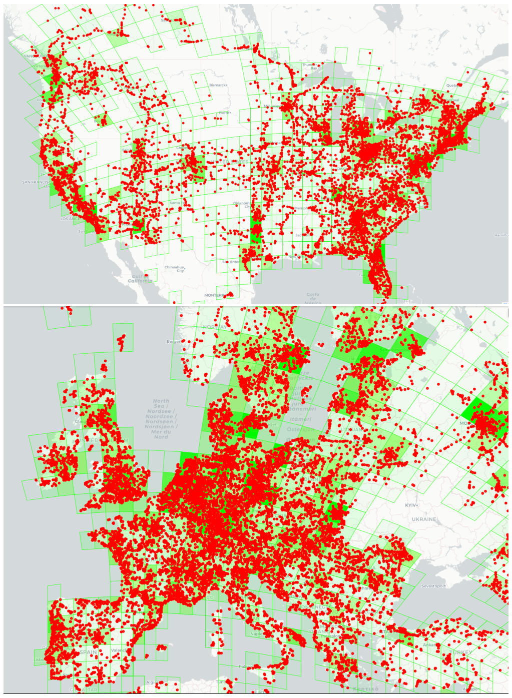

Figure 1. Example of S2 Cells (green cells) partitioning at level 6 with 500K randomly sampled images.

The second model replaced the ResNet50 backbone with a Vision Transformer (ViT-B/16). While CNNs excel at local patterns, ViTs use self-attention to assess the relationship between image patches, enabling them to capture long-range dependencies and global context more effectively. A similar projection head was attached to the ViT backbone to produce geo-embeddings. The model was trained using the same Supervised Contrastive Loss, allowing for a direct comparison between CNN and Transformer architectures. The last 4 of the 12 encoder blocks were unfrozen during training, resulting in approximately 29 million learnable parameters.

Lastly, the third architecture is a two-stage, coarse-to-fine ViT. This approach first learns a coarse mapping and then fine-tunes the embedding space to be aware of continuous real-world distances.

**Stage 1: Coarse Location Classification.** This stage is identical to the second model. A ViT is trained with Supervised Contrastive Loss [16] using S2 cell IDs as labels, creating an embedding space where images are clustered by coarse geographical region.

**Stage 2: Fine-Tuning with Distance-Aware Triplet Loss.** After Stage 1, the model can group images into regions but does not understand that some regions are neighbors while others are on opposite sides of the planet.

Stage 2 teaches the model this concept of continuous distance. The model from Stage 1 is further fine-tuned using Triplet Margin Loss. Crucially, S2 cell labels are ignored, and real-world geographic coordinates are used to form triplets within each batch: an anchor image, a positive image (geographically closest to the anchor image), and a negative image (geographically farthest to the anchor image).

The loss function then pushes the model to ensure that the distance between the anchor and positive embeddings is smaller than the distance to the negative embedding by a given margin. This approach is an implementation of ”haversine smoothing,” from PIGEON [4] which aims to make the model aware of the geographic proximity of its output classes.

Lastly, a final regression head can be trained on this distance-aware embedding space to predict precise coordinates.

## 3.3. Loss Functions

### 3.3.1 Supervised Contrastive Loss [16]

I mainly used the Supervised Contrastive when training the models. This loss function encourages the embeddings of images from the same S2 cell to be closer together (positive pair) than the embedding of images from different S2 cells (negative pairs). Each S2 cell ID is treated as a label, so this loss function clusters together images from the same geographic region (S2 cell), making it ideal for learning meaningful geo-aware image embeddings.

$$\mathcal{L}_{supcon} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp(\mathbf{z}_i \cdot \mathbf{z}_p / \tau)}{\sum_{a \in A(i)} \exp(\mathbf{z}_i \cdot \mathbf{z}_a / \tau)} \tag{1}$$

For each image embedding in a batch (the ”anchor”), this loss contrasts it with all other embeddings. The numerator is the sum of similarities between the anchor and all ”positive” embeddings (those from the same S2 cell). The denominator is the sum of similarities between the anchor and all embeddings in the batch. By maximizing this fraction, the loss pulls embeddings from the same S2 cell together while simultaneously pushing them apart from the embeddings of all other cells.

### 3.3.2 Triplet Margin Loss [17]

In the second stage of Model 3, Triplet Margin Loss was used. The loss is defined as:

$$\mathcal{L}_{triplet} = \max\left(\|e_a - e_p\|_2^2 - \|e_a - e_n\|_2^2 + m, 0\right) \tag{2}$$

Here, triplets are formed using geographic coordinates: an anchor ($e_a$), a geographically close positive ($e_p$), and a distant negative ($e_n$). The loss function’s goal is to ensure the squared Euclidean distance between the anchor and positive embeddings is smaller than the distance to the negative by at least a margin $m$ (set to 0.2). This process forces the embedding space to reorganize itself into a continuous map where embedding distance correlates with geographic distance, overcoming the artificial boundaries of S2 cells.

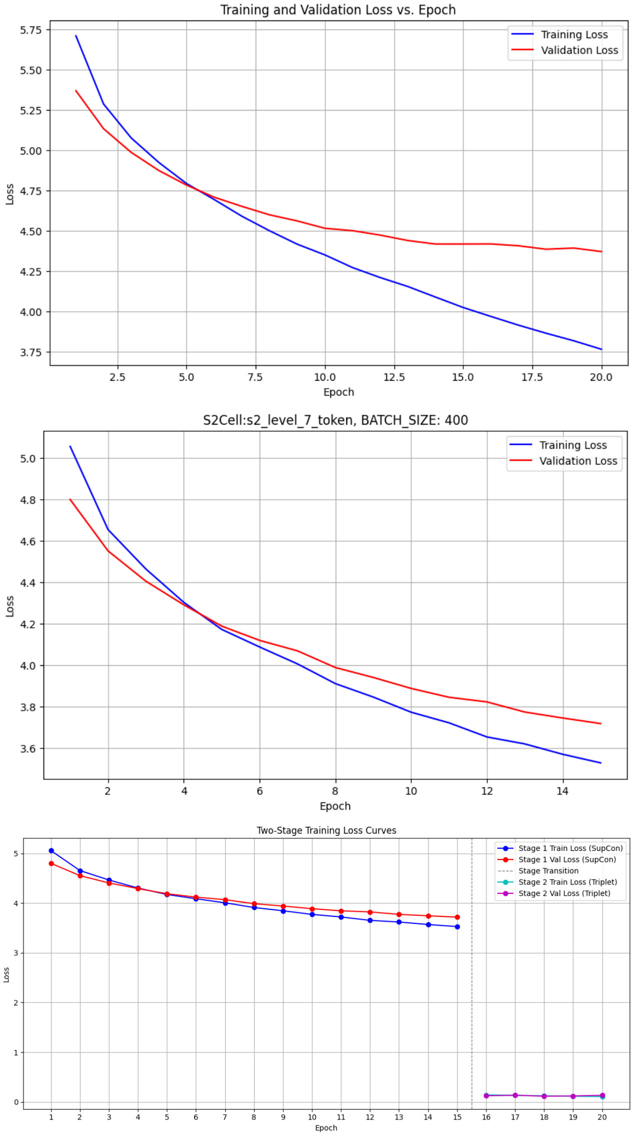

Figure 2. Traning and validation loss v.s epoch of the three models. ResNet50 (top), ViT single-stage (middle), ViT two-stages (bottom). The loss function of ResNet50 started to show a sign of overfitting around epochs 10.

## 3.4. Geo-Embedding Database Creation

In order to effectively compute inference, a pre-computed database of the reference embedding vectors is required, as the inference stage will perform K-NN look-ups on this database.

After training, one million embedding vectors were generated for the training images (same amount as the training dataset). These embeddings, along with their ground-truth S2 labels and coordinates, were stored in a FAISS (Facebook AI Similarity Search) [18] index for efficient similarity search.

### 3.5. Inference with K-Nearest Neighbors

The inference stage process predicts the location of a query image from the test set by leveraging the precomputed reference embedding database. The process consists of three main steps:

- **Step 1:** For each batch of the test images, generate a corresponding batch of embedding vectors using the same trained model.
- **Step 2:** A query embedding vector within the batch is then used to search against the reference database using the FAISS index [18]. This search retrieves the top-k nearest neighbours, which are the reference embedding vectors from the training set that are most similar to the query embedding.
- **Step 3:** Lastly, compute the predicted lat/long of the query embedding. I use different algorithms to compute the final latlng, such as top-1 neighbor, and weighted geographic mean by averaging the location of the K selected neighbors.

## 4. Dataset and Features

The models were trained and evaluated on subsets of the OpenStreetView-5M (OSV-5M) dataset [1], a large-scale collection of street-level imagery. Due to computational constraints, randomly sampled subsets were used while maintaining geographic diversity.

### 4.1. Dataset and Split Sizes

Two main subsets were created:

- A **500,000-image subset** for the initial ResNet50 baseline model.
- A **1-million-image subset** for the more complex Vision Transformer models.

Each subset was split 80/20 into training and validation sets. A separate, non-overlapping set of images was reserved for final testing.

### 4.2. Data Preprocessing and Augmentation

All images underwent a standardized preprocessing pipeline. They were resized to 224x224 pixels and normalized using the standard ImageNet mean and standard deviation. No data augmentation (e.g., random flipping, color jittering) was applied, as such transformations could mislead the model by altering important geographic indicators, such as which side of the road vehicles drive on.

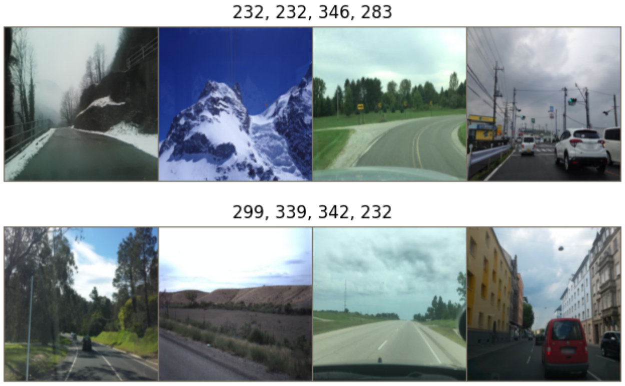

Figure 3. Example of input images and its corresponding S2 Cell label index.

## 5. Experiments and Results

### 5.1. Experimental Setup

- **Model 1 (ResNet50 Baseline):** Trained for 20 epochs with the Adam optimizer. The top two layers were unfrozen. Supervised Contrastive Loss was used with a temperature $\tau$ of 0.07, and batch size of 128.
- **Model 2 (ViT Single-Stage):** Trained for 15 epochs with the AdamW optimizer. The top four encoder blocks were unfrozen. Supervised Contrastive Loss was used with $\tau = 0.07$, and batch size of 400.
- **Model 3 (ViT Two-Stage):** Both stages use batch size of 400.
  - **Stage 1:** Trained for 15 epochs as per Model 2.
  - **Stage 2:** Fine-tuned for 5 epochs using Triplet Margin Loss with a margin $m$ of 0.2. The last four encoder blocks remained unfrozen,

For all models, a differential learning rate was used: 1e-4 for the projection head and 1e-5 for the unfrozen backbone layers. The ResNet50 model was trained on an L4 GPU, while ViT models were trained on an A100 GPU.

### 5.2. Evaluation Metrics

Model performance was evaluated using standard geolocation metrics. The great-circle distance between predicted and true coordinates was calculated. Results are reported as mean and median error (km), alongside the percentage of predictions falling within various error radii: 25km, 200km, 750km, and 2500km.

### 5.3. Results

The models were evaluated on three separate test sets, each consisting of 1,500 randomly selected images: Global, Europe-only, and North America-only. Tables 1, 2, and 3 present the best-K, median, and mean localization errors in kilometers. Additionally, Tables 4, 5, and 6 report the percentage of predictions falling within various error radii. The OpenStreetView-5M model [1], which represents the current state of the art, is used as the baseline for comparison.

On the other hand, figure 4 and figure 5 display the accuracy maps of the three models in North America and Europe. Errors are color-coded based on distance thresholds. Green indicates errors of 500 km or less, yellow for errors up to 1500 km, and orange for errors up to 2500 km. Errors greater than 2500 km are labeled red.

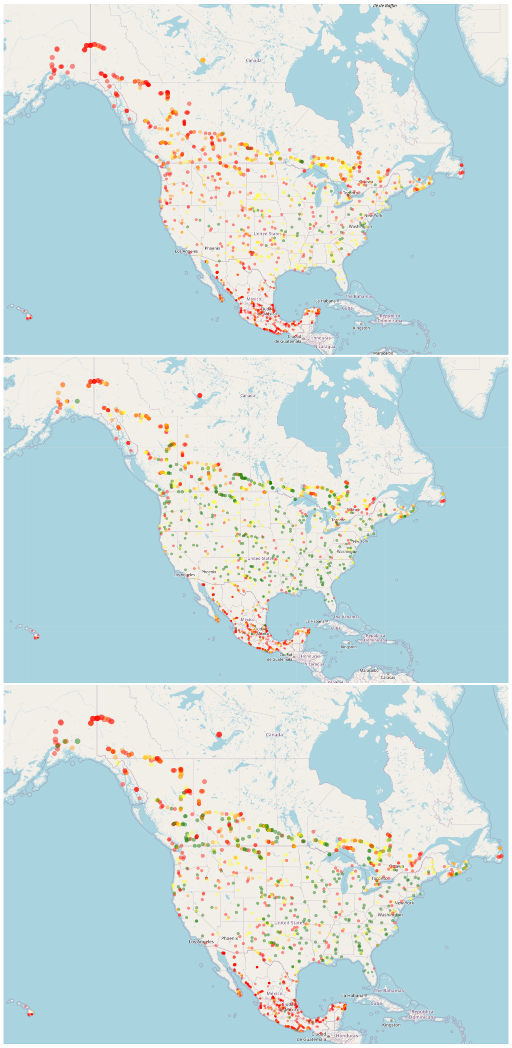

Figure 4. Accuracy map for ResNet50, ViT (single-stage), and ViT (two-stages), showing the ViT models outperformed the ResNet model

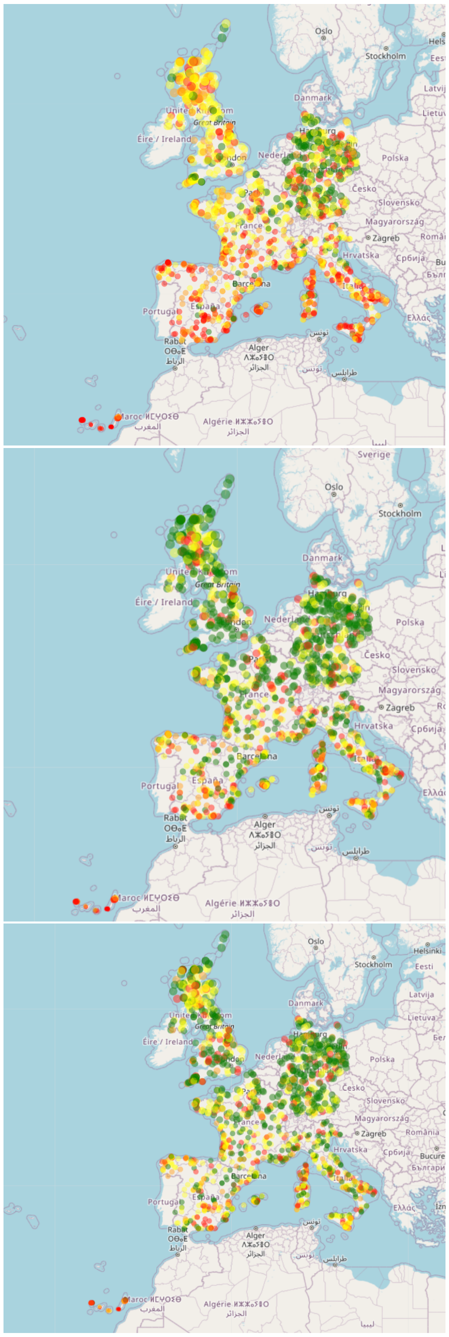

Figure 5. Accuracy maps for ResNet50, ViT (single-stage), and ViT (two-stage) in EU, showing the ViT models outperformed the ResNet model

Table 1. Global Test Set: Error Metrics

| **Model** | **Best k** | **Median Error (km)** | **Mean Error (km)** |
|---|---|---|---|
| ResNet50 | 10 | 4625.30 | 5784.09 |
| ViT (Single Stage) | 1 | 1628.45 | 4296.87 |
| ViT (Two-Stage) | 1 | **1567.39** | **3907.71** |
| OSV5M (Baseline) | N/A | 196.12 | 1726.75 |

Table 2. Europe-Only Test Set: Error Metrics

| **Model** | **Best k** | **Median Error (km)** | **Mean Error (km)** |
|---|---|---|---|
| ResNet50 | 10 | 1200.12 | 2303.76 |
| ViT (Single Stage) | 1 | **648.60** | 1988.10 |
| ViT (Two-Stage) | 1 | 671.56 | **1755.32** |
| OSV5M (Baseline) | N/A | 61.98 | 592.88 |

Table 3. North America-Only Test Set: Error Metrics

| **Model** | **Best k** | **Median Error (km)** | **Mean Error (km)** |
|---|---|---|---|
| ResNet50 | 1 | 2616.33 | 4718.30 |
| ViT (Single Stage) | 1 | **1338.02** | **3284.21** |
| ViT (Two-Stage) | 1 | 1594.37 | 3868.25 |
| OSV5M (Baseline) | N/A | 256.73 | 1471.77 |

Table 4. Global Test Set: Accuracy Metrics

| **Model** | **Acc@25km (%)** | **Acc@750km (%)** | **Acc@2500km (%)** |
|---|---|---|---|
| ResNet50 | 0.00 | 8.20 | 31.40 |
| ViT (Single Stage) | 7.13 | **38.73** | 55.93 |
| ViT (Two-Stage) | **9.00** | 38.60 | **58.27** |
| OSV5M (Baseline) | 16.6 | 69.33 | 83.27 |

Table 5. Europe-Only Test Set: Accuracy Metrics

| **Model** | **Acc@25km (%)** | **Acc@750km (%)** | **Acc@2500km (%)** |
|---|---|---|---|
| ResNet50 | 0.13 | 28.60 | 74.73 |
| ViT (Single Stage) | 8.67 | **53.73** | 84.07 |
| ViT (Two-Stage) | **9.33** | 52.87 | **86.40** |
| OSV5M (Baseline) | 32.33 | 85.00 | 95.87 |

Table 6. North America-Only Test Set: Accuracy Metrics

| **Model** | **Acc@25km (%)** | **Acc@750km (%)** | **Acc@2500km (%)** |
|---|---|---|---|
| ResNet50 | 0.27 | 14.60 | 48.47 |
| ViT (Single Stage) | 6.80 | **37.47** | **64.60** |
| ViT (Two-Stage) | **7.20** | 36.60 | 60.07 |
| OSV5M (Baseline) | 16.53 | 69.60 | 86.73 |

### 5.4. Discussion

Unfortunately, our models did not outperform the state-of-the-art OSV-5M baseline. However, the comparison among the three proposed models clearly demonstrates a notable performance improvement as the model architecture and training strategy became more sophisticated. The results also highlight that model performance varies depending on the geographic scope of the test set.

**ResNet50 vs. ViT:** The shift from a CNN to a ViT yielded a dramatic improvement. On the global set, the ViT reduced the median error by nearly 3,000 km, supporting the hypothesis that a ViT's global self-attention mechanism is better suited for capturing the wide-ranging visual cues required for geolocation.

**Effectiveness of the Two-Stage Model:** The two-stage ViT showed insightful, mixed performance. It offered a modest improvement on the global test set. It reduced the median error of single-stage ViT by 61 km, and +2% increase on accuracy within 25 km error radii.

**Limitations of the Two-Stage Model:** Overall, it seems the two-stage model underperformed compared to the single-stage ViT on the North America test set. The hypothesis of this result is that the "haversine smoothing" process [4], while beneficial for Europe's denser and more continuous visual landscape, may have been detrimental for North America. By imposing geographic smoothness, the model might have been forced to "smear" well-defined visual clusters from Stage 1 to satisfy the distance constraint, reducing certainty in less-dense regions.

## 6. Conclusion and Future Work

This paper explored image-based geolocation through a series of increasingly sophisticated deep learning models. This work confirms two key findings. First, the architectural shift from a CNN to a Vision Transformer (ViT) yields a substantial performance improvement, confirming that the global self-attention mechanism is better suited for capturing geographic cues than a standard convolutional approach. Second, a novel two-stage, coarse-to-fine ViT architecture demonstrates a more complex, region-dependent performance. Its performance degrades in regions like North America compared to the single-stage ViT. This suggests that the distance-aware fine-tuning in Stage 2, while powerful, is not universally applicable and may require further fine-tuning and adjustments.

Future work could explore several promising improvements:

- **Dynamic Data Partitioning:** The use of static S2 cells is a limitation, as data-sparse regions are disadvantaged by contrastive loss. Partitioning data using natural boundaries, such as rivers or mountain ranges, could encode more meaningful geological information.

- **Advanced Distance-Aware Loss:** The current Triplet Loss implementation uses only the single closest and farthest samples. A more advanced loss function that considers multiple samples at varying distances or directly incorporates Haversine distances could create a more accurate embedding space and potentially resolve the under-performance in North America.

- **Hyperparameter Optimization:** Due to computational constraints, extensive hyperparameter tuning was not feasible. Further optimization of learning rates, the number of unfrozen layers, weight decay, and contrastive temperature could yield additional performance gains.

## A. Appendix

In addition to the three models listed above, I also trained a large Vision Transformer ('google/vit-large-patch16-224-in21k') on the full 5 million image dataset, equivalent to approximately two epochs of training. Due to hardware constraints of a single 16GB GPU, I employed several strategies to enable training.

### A.1. Training Strategy and Rationale

This approach was designed to overcome limited VRAM while leveraging the benefits of supervised contrastive loss, which performs better with larger batch sizes. I used 'torch.autocast' for mixed-precision training to reduce the memory footprint.

The core of the strategy was a "progressive training schedule". I began by training all layers with a small batch size. In subsequent stages, I progressively froze the lower layers of the network and increased the batch size. This approach is based on the hypothesis that lower layers of the ViT can learn low-level geographic features (e.g., S2 cell level 3) first, while higher layers can be fine-tuned on more granular, higher-level geographic data with the benefit of larger, more effective batches.

### A.2. Hierarchical Loss Function

A hierarchical supervised contrastive loss function is implemented for this traning, inspired by the work of Weyand et al. [21]. The total loss is a weighted sum of individual supervised contrastive losses calculated at different S2 geospatial cell levels. Because S2 cells are organized hierarchically, this method encourages the model to learn geographic relationships at multiple scales simultaneously.

$$\mathcal{L}_{total} = \sum_{i \in S} w_i \cdot \mathcal{L}_{s2cell}^{(i)}$$

## A.3. Staged Training Setup Details

The training was conducted continuously across four distinct stages, with the model's weights carried over from one stage to the next. The specific parameters for each stage are detailed in Table 7. The initial ”warm-up” stage used a combined loss of S2 Cell level 3 Supervised Contrastive Loss and Haversine distance to stabilize training with a small batch size.

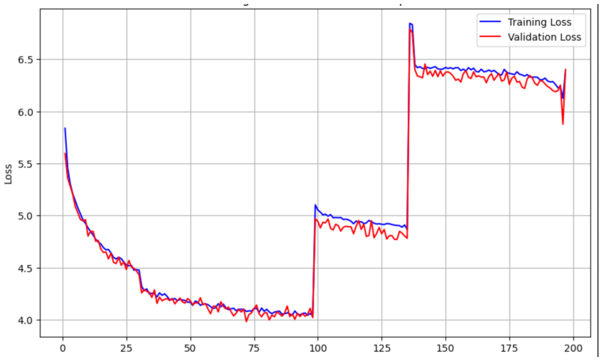

Figure 6. The loss curve of training and validation v.s folder

| Stage | Data Used | Layers Unfrozen | Batch Size | Loss Configuration (Levels & Weights factors) |
|---|---|---|---|---|
| 1 | 10 folders | All (24) | 16 | S2Cell L3 + Haversine Distance |
| 2 | 97 folders | Top 12 | 128 | S2Cell L3, L4, L5: $[1, 0.3, 0.05]$ |
| 3 | 30 folders | Top 7 | 256 | S2Cell L3-L6: $[1, 0.3, 0.1, 0.03]$ |
| 4 | 60 folders | Top 1 | 1024 | S2Cell L3-L6: $[0.8, 0.35, 0.15, 0.05]$ |

Table 7. Progressive training schedule for the ViT-Large model. ”Folders” refers to subsets of the 5M image dataset, each containing 50k images.

## A.4. Staged Training Results

The following tables show the inference results for this multi-stage ViT large model using the same random Global, Europe-only, NA-only test sets.

| Model | Best k | Median Error (km) | Mean Error (km) |
|---|---|---|---|
| ResNet50 (Baseline) | 10 | 4625.30 | 5784.09 |
| ViT (Single Stage) | 1 | 1628.45 | 4296.87 |
| ViT (Two-Stage) | 1 | 1567.39 | 3907.71 |
| ViT-Large (Multi-Stage) | 1 | **1071.67** | **3115.58** |

Table 8. Global Test Set: Error Metrics

| Model | Best k | Median Error (km) | Mean Error (km) |
|---|---|---|---|
| ResNet50 (Baseline) | 10 | 1200.12 | 2303.76 |
| ViT (Single Stage) | 1 | 648.60 | 1988.10 |
| ViT (Two-Stage) | 1 | 671.56 | 1755.32 |
| ViT-Large (Multi-Stage) | 10 | **418.32** | **1047.86** |

Table 9. EU Test Set: Error Metrics

| Model | Best k | Median Error (km) | Mean Error (km) |
|---|---|---|---|
| ResNet50 (Baseline) | 1 | 2616.33 | 4718.30 |
| ViT (Single Stage) | 1 | 1338.02 | 3284.21 |
| ViT (Two-Stage) | 1 | 1594.37 | 3868.25 |
| ViT-Large (Multi-Stage) | 1 | **917.57** | **2756.05** |

Table 10. NA Test Set: Error Metrics

| Model | Acc@25km (%) | Acc@750km (%) | Acc@2500km (%) |
|---|---|---|---|
| ResNet50 (Baseline) | 0.00 | 8.20 | 31.40 |
| ViT (Single Stage) | 7.13 | 38.73 | 55.93 |
| ViT (Two-Stage) | **9.00** | 38.60 | 58.27 |
| ViT-Large (Multi-Stage) | 3.33 | **42.80** | **68.13** |

Table 11. Global Test Set: Accuracy Metrics

| Model | Acc@25km (%) | Acc@750km (%) | Acc@2500km (%) |
|---|---|---|---|
| ResNet50 (Baseline) | 0.13 | 28.60 | 74.73 |
| ViT (Single Stage) | 8.67 | 53.73 | 84.07 |
| ViT (Two-Stage) | **9.33** | 52.87 | 86.40 |
| ViT-Large (Multi-Stage) | 1.67 | **68.70** | **91.50** |

Table 12. EU Test Set: Accuracy Metrics

| Model | Acc@25km (%) | Acc@750km (%) | Acc@2500km (%) |
|---|---|---|---|
| ResNet50 (Baseline) | 0.27 | 14.60 | 48.47 |
| ViT (Single Stage) | 6.80 | 37.47 | 64.60 |
| ViT (Two-Stage) | **7.20** | 36.60 | 60.07 |
| ViT-Large (Multi-Stage) | 2.47 | **44.27** | **72.27** |

Table 13. EU Test Set: Accuracy Metrics

# B. Conclusion

Building on the three baseline models presented in the paper, I developed a more advanced pipeline that leverages a Vision Transformer (ViT) combined with a hierarchical supervised contrastive loss. This enhanced model achieved superior performance compared to the other variants, including ResNet50, single-stage ViT, and two-stage ViT. These results highlight a promising direction for improving geolocation accuracy through more structured training strategies.

Future work could focus on further reducing localization error by investigating deeper or more specialized model architectures, improving the geocell partitioning strategy, and employing more effective loss functions.

## References

[1] Astruc, G., Dufour, N., Siglidis, I., Aronssohn, C., Bouia, N., Fu, S., Loiseau, R., Nguyen, V. N., Raude, C., Vincent, E., Xu, L., Zhou, H., & Landrieu, L. (n.d.). OpenStreetView-5M: The Many Roads to Global Visual Geolocation In https://osv5m.github.io, 2024

[2] J. Hays and A. A. Efros. IM2GPS: Estimating Geographic Information from a Single Image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[3] T. Weyand, I. Kostrikov, and J. Philbin. PlaNet - Photo Geolocation with Convolutional Neural Networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[4] PIGEON: Predicting Image Geolocations. https://arxiv.org/pdf/2307.05845, accessed on [June 4, 2025].

[5] N. Vo and J. Hays. Revisiting IM2GPS in the Deep Learning Era. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] Wu, M., & Huang, Q. (2022). IM2City: Image Geo-localization via Multi-modal Learning. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery (GeoAI '22)*, pp. 50–61. https://dl.acm.org/doi/10.1145/3557918.3565868, accessed on [June 4, 2025].

[7] Zamir, A. R., & Shah, M. (2010). Accurate Image Localization Based on Google Maps Street View. In Daniilidis, K., Maragos, P., & Paragios, N. (Eds.), *Computer Vision – ECCV 2010*, pp. 255–268. Springer, Berlin, Heidelberg. ISBN 978-3-642-15561-1.

[8] Suresh, S., Chodosh, N., & Abello, M. (2018). DeepGeo: Photo Localization with Deep Neural Network. https://arxiv.org/abs/1810.03077, accessed on [June 4, 2025].

[9] Baatz, G., Saurer, O., Köser, K., & Pollefeys, M. (2012). Large Scale Visual Geo-Localization of Images in Mountainous Terrain. In Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., & Schmid, C. (Eds.), *Computer Vision – ECCV 2012*, pp. 517–530. Springer, Berlin, Heidelberg. ISBN 978-3-642-33709-3.

[10] Tzeng, E., Zhai, A., Clements, M., Townshend, R., & Zakhor, A. (2013). User-Driven Geolocation of Untagged Desert Imagery Using Digital Elevation Models. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 237–244. doi: 10.1109/CVPRW.2013.42.

[11] Cao, L., Smith, J. R., Wen, Z., Yin, Z., Jin, X., & Han, J. (2012). BlueFinder: Estimate Where a Beach Photo Was Taken. In *Proceedings of the 21st International Conference on World Wide Web, WWW '12 Companion*, pp. 469–470. Association for Computing Machinery, New York, NY, USA. ISBN 9781450312301. doi: 10.1145/2187980.2188081. https://doi.org/10.1145/2187980.2188081, accessed on [June 4, 2025].

[12] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet Classification with Deep Convolutional Neural Networks. In Pereira, F., Burges, C., Bottou, L., & Weinberger, K. (Eds.), *Advances in Neural Information Processing Systems*, Vol. 25. Curran Associates, Inc. https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf, accessed on [June 4, 2025].

[13] Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., & Zhai, X. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. https://arxiv.org/abs/2010.11929, accessed on [June 4, 2025].

[14] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778. https://doi.org/10.1109/CVPR.2016.90, accessed on [June 4, 2025].

[15] Google S2 Geometry Library. S2Cell Hierarchical Spatial Indexing. https://s2geometry.io/, accessed on [June 4, 2025].

[16] Khosla, P., Teterwak, P., Wang, C., Sarna, A., Tian, Y., Isola, P., Maschinot, A., Liu, C., & Krishnan, D. (2020). Supervised Contrastive Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33, pp. 18661–18673. https://arxiv.org/abs/2004.11362, accessed on [June 4, 2025].

[17] Hoffer, E., & Ailon, N. (2015). Deep Metric Learning Using Triplet Network. In *International Workshop on Similarity-Based Pattern Recognition (SIMBAD)*,

pp. 84–92. https://arxiv.org/abs/1412.6622, accessed on [June 4, 2025].

[18] Johnson, J., Douze, M., & Jégou, H. (2017). Billion-scale similarity search with GPUs. In *IEEE Transactions on Big Data*. https://faiss.ai/, accessed on [June 4, 2025].

[19] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255. https://www.image-net.org/, accessed on [June 4, 2025].

[20] Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., & Torralba, A. (2017). Places: A 10 Million Image Database for Scene Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6), 1452–1464. http://places.csail.mit.edu/, accessed on [June 4, 2025].

[21] GCN4Geo: Graph Convolutional Networks for Geolocation Prediction. https://arxiv.org/pdf/2204.13207, accessed on [June 4, 2025].