PIGEON: Predicting Image Geolocations

Supplementary Material

Supplementary material overview

Given the page limit of CVPR submissions, some of the contents of our work did not fit within the main body of the paper. Hence, we include additional details in this Supplementary Material. Specifically, we expand on the following topics:

A. Semantic geocell creation
B. Implementation details
C. Ablation study on pretraining captions
D. Ablation study on training datasets
E. Auxiliary data sources
F. Ablation studies on non-distance metrics
G. Additional analyses
H. Deployment to GeoGuessr
I. Acknowledgements

A. Semantic geocell creation

In the body of our work, we described how our semantic geocell creation algorithm works on a high level. Similar to approaches in prior literature such as Theiner et al. [33], we create a hierarchy of administrative areas and merge adjacent geocells until a set minimum number of training samples per geocell is reached. This, however, results in a highly imbalanced classification problem, especially for larger training datasets. A major contribution of our work is that we define a method to split larger geocells into smaller, still semantically meaningful cells, by leveraging the information contained in the training data's geolocations. The key insight is that locations from most training distributions tend to cluster around popular places and landmarks, and these clusters can be extracted.

Algorithm 1 shows a slightly simplified version of how we split large geocells into multiple smaller ones without the help of administrative boundary information, resulting in a much more balanced geocell classification dataset. As one can see, the algorithm only depends on the geocell boundaries or shape definitions g, the training dataset x, an OPTICS clustering algorithm with parameters p (optionally round-specific parameters p_j), and a minimum cell size MINSIZE. The VORONOI algorithm takes a set of points as input and outputs a new geocell shape defined by these points which can be removed from the original cell shape.

Figure 5 shows a small geocell that has been extracted from a larger geocell covering the entire city of Vienna, Austria, via Voronoi tessellation. The partitions within the blue geocells are the result of the Voronoi tesselation algorithm assigning to each training sample all geographic area to which it is closest.

Algorithm 1 Simplified Semantic Geocell Splitting

```
**Input:** geocell boundaries $g$, training samples $x$,
OPTICS parameters $p$, minimum cell size MINSIZE.
Initialize $j = 1$.
**repeat**
  Initialize $C$ = OPTICS($p_j$).
  **for** $g_i$ **in** $g$ **do**
    Define $x_i = \{x_k | x_k \in x \wedge x_k \in g_i\}$.
    **repeat**
      Cluster $c = C(x_i)$.
      $c_{max} = c_k$ where $|x_{i,k}| \geq |x_{i,l}| \forall l$.
      **if** $|c_{max}| >$ MINSIZE and $|x \setminus x_{i,k}| >$ MINSIZE **then**
        New cell $g_{new}$ = VORONOI($x_{i,k}$).
        $g_i = g_i \setminus g_{new}$.
        Assign $x_i$ to cells $i$ and $new$, respectively.
      **end if**
    **until** convergence
  **end for**
  $j = j + 1$
**until** $j$ is $|p|$
```

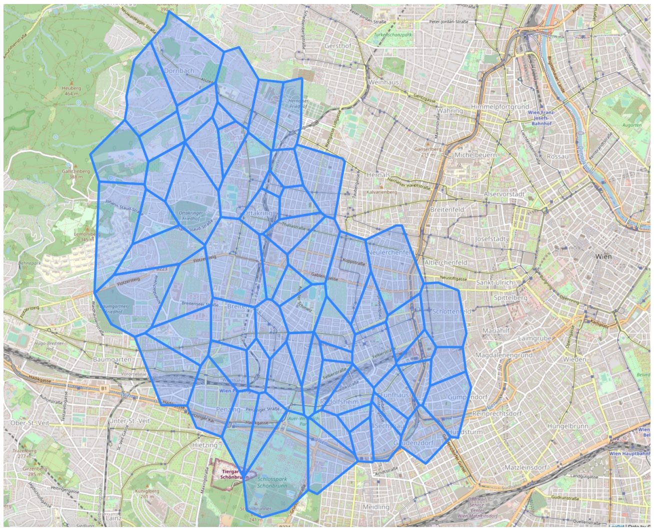


Figure 5. Voronoi tessellation applied in the process of geocell creation for points of an OPTICS cluster in Vienna, Austria, based on political boundaries from GADM [10].

B. Implementation details

In this section, we describe the implementation details of PIGEON and PIGEOTTO and further illustrate how the two models differ from each other.

B.1. Model input

The biggest difference between PIGEON and PIGEOTTO is that PIGEON takes a four-image Street View panorama as input, whereas PIGEOTTO takes a single image as input. Images are always cropped to a square aspect ratio before being fed into the models. Figure 6 shows a representative input for PIGEON, depicting a 360-degree, four-image Street View panorama taken in Pegswood, England.

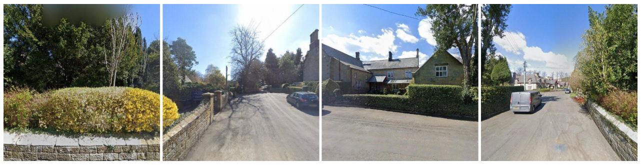


Figure 6. Four images comprising a 360-degree panorama from a location in Pegswood, England, in our dataset.

PIGEOTTO's training dataset is vastly different to PIGEON's Street View input; the model takes a single image as input and was trained on a highly diverse image geolocalization dataset. Figure 7 shows eight images sampled from the MediaEval 2016 dataset [20] which was derived from user-uploaded Flickr images. It is clearly apparent that some of the images are extremely difficult to geolocalize, for example because they were taken indoors.

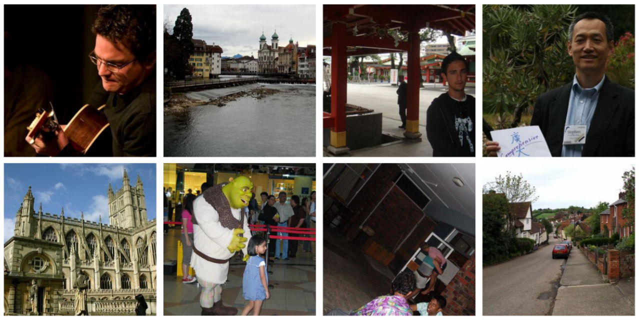


Figure 7. Eight samples from the MediaEval 2016 dataset [20].

B.2. Pretraining

Table 4 shows the hyperparameter settings employed for our contrastive pretraining of CLIP for the task of image geolocalization. The CLIP weights were initialized with the pretrained weights of OpenAI's CLIP implementation.[4]

[4] https://huggingface.co/openai/clip-vit-large-patch14-336.

Table 4. Hyperparameter settings for pretraining CLIP's vision encoder for the task of image geolocalization.

Parameter	PIGEON	PIGEOTTO
GPU Type	A100 80GB	A100 80GB
Number of GPUs	4	4
Dataset Source	Street View	Flickr
Dataset Size (Samples)	$\sim$ 1M	$\sim$ 4.2M
Batch Size	32	32
Gradient Accumulation Steps	8	8
Optimizer	AdamW	AdamW
Learning Rate	$1e^{-6}$	$5e^{-7}$
Weight Decay	$1e^{-3}$	$1e^{-3}$
Warmup (Epochs)	0.2	0.02
Training Epochs	3	2
Adam β_1	0.9	0.9
Adam β_2	0.98	0.98

B.3. Fine-tuning

The fine-tuning of PIGEON and PIGEOTTO consists of adding a linear layer on top of the pretrained vision encoder, mapping image embeddings to a fixed number of geocells. During this process, the weights of the vision encoder remain frozen. Table 5 shows the hyperparameters used in this training step. Both PIGEON and PIGEOTTO were trained until convergence.

Table 5. Hyperparameter settings for fine-tuning CLIP's vision encoder via a linear projection layer onto geocells.

Parameter	PIGEON	PIGEOTTO
GPU Type	A100 80GB	A100 80GB
Number of GPUs	1	1
Dataset Source	Street View	Flickr + Wikipedia
Dataset Size (Samples)	$\sim$ 100k	$\sim$ 4.5M
Number of Geocells	2,203	2,076
Haversine Smoothing τ	75	65
Batch Size	1024	1024
Gradient Accumulation Steps	1	1
Optimizer	AdamW	AdamW
Learning Rate	$5e^{-5}$	$2e^{-5}$
Weight Decay	0.01	0.01
Training Epochs	Convergence	Convergence
Adam β_1	0.9	0.9
Adam β_2	0.999	0.999

B.4. Hierarchical refinement

We use a hierarchical retrieval mechanism over location clusters to refine predictions. As a first step, location clusters are pre-computed using an OPTICS clustering algorithm. Then, during inference, a cluster is selected according to Equation (5). Finally, the location guess is refined within the top selected cluster. The refinement process is also dependent on a number of parameters, the most important of which are listed in Table 6 and contrasted between PIGEON and PIGEOTTO.

Table 6. Parameters used in our hierarchical retrieval mechanism over location clusters.

Parameter	**PIGEON**	**PIGEOTTO**
Number of Geocell Candidates	5	40
Maximum Refinement Distance (km)	1,000	None
Distance Metric	Euclidian	Euclidian
Softmax Temperature	1.6	0.6
OPTICS Min Samples (Cluster Creation)	3	10
OPTICS xi (Cluster Creation)	0.15	0.1

C. Ablation study on pretraining captions

In Section 3.3, we describe a novel multi-task contrastive pretraining method for image geolocalization. The ablation in Table 7 shows that our pretraining reduces PIGEON's median kilometer error significantly from 57.8 to 44.4 kilometers (-23.3%) versus no pretraining as in Wu and Huang [40]. Our innovation is that we are the first to design a multimodal *and* multi-task contrastive pretraining objective for CLIP through the use of synthetic captions, and further find that the multi-task component of our method is highly effective; we observe a positive transfer from the auxiliary tasks embedded in our captions to the task of geolocalization, reducing our median error from 49.4 to 44.4 kilometers (-10.2%) compared to pretraining solely with location captions as in Haas et al. [13]. Our multi-task contrastive pretraining method is general enough that it could also be employed in other problem domains.

Table 7. Ablation study of CLIP pretraining captions for PIGEON on a holdout dataset of 5,000 Street View locations.

Ablation	**Median Error**	**Distance (% @ km)**				
	km	**1 km**	**25 km**	**200 km**	**750 km**	**2,500 km**
PIGEON **[location + auxiliary captions]**	**44.35**	**5.36**	**40.36**	**78.28**	**94.52**	**98.56**
PIGEON **[location captions** as in [13]]	49.37	4.62	38.46	77.10	94.34	98.48
PIGEON **[no pretraining** as in [40]]	57.80	4.48	36.18	74.88	93.24	98.04

D. Ablation study on training datasets

Section 4.1 in the body of our paper describes the different datasets used to train PIGEON and PIGEOTTO. While PIGEON was purely trained on Street View imagery, the training dataset for PIGEOTTO contains a combination of 4,166,186 geo-tagged images from the MediaEval 2016 dataset [20] and 340,579 images from the Google Landmarks v2 dataset [39]. Prior works' benchmark results, listed in Table 3, employ a diverse range of training datasets with the goal of building the best performing and robust image geolocalization models. Since the prior SOTA model Geodecoder [7] was exclusively trained on the MediaEval 2016 dataset [20], we include an additional training dataset ablation for PIGEOTTO in Table 8 to distinguish data selection from system design effects.

Table 8. Ablation study of PIGEOTTO's Google Landmarks v2 [39] data (340k images) against prior SOTA on five benchmarks.

Benchmark	**Method**	**Median Error**	**Distance (% @ km)**				
			Street	*City*	*Region*	*Country*	*Continent*
		km	**1 km**	**25 km**	**200 km**	**750 km**	**2,500 km**
IM2GPS [14]	GeoDecoder [7]	$\sim$ **25**	**22.1**	**50.2**	**69.0**	80.0	89.1
	PIGEOTTO **[ME16]**	75.6	11.8	38.8	63.7	80.6	**91.1**
	PIGEOTTO **[ME16 + Landmarks]**	70.5	14.8	40.9	63.3	**82.3**	**91.1**
IM2GPS3k [37]	GeoDecoder [7]	$>$ 200	**12.8**	33.5	45.9	61.0	76.1
	PIGEOTTO **[ME16]**	163.6	10.9	35.8	52.4	70.7	84.4
	PIGEOTTO **[ME16 + Landmarks]**	**147.3**	11.3	**36.7**	**53.8**	**72.4**	**85.3**
YFCC4k [37]	GeoDecoder [7]	$\sim$ 750	10.3	**24.4**	33.9	50.0	68.7
	PIGEOTTO **[ME16]**	418.8	9.5	22.5	38.8	60.7	76.9
	PIGEOTTO **[ME16 + Landmarks]**	**383.0**	**10.4**	23.7	**40.6**	**62.2**	**77.7**
YFCC26k [25]	GeoDecoder [7]	$\sim$ 750	10.1	23.9	34.1	49.6	69.0
	PIGEOTTO **[ME16]**	356.5	10.1	24.6	41.3	62.6	78.7
	PIGEOTTO **[ME16 + Landmarks]**	**333.3**	**10.5**	**25.8**	**42.7**	**63.2**	**79.0**
GWS15k [7]	GeoDecoder [7]	$\sim$ 2,500	**0.7**	1.5	8.7	26.9	50.5
	PIGEOTTO **[ME16]**	440.8	0.1	8.7	30.1	64.0	84.7
	PIGEOTTO **[ME16 + Landmarks]**	**415.4**	**0.7**	**9.2**	**31.2**	**65.7**	**85.1**

In Table 8, we observe that even when trained using the same data (ME16 [20]), PIGEOTTO outperforms the prior SOTA Geodecoder [7] by a large margin on four out of five benchmarks. The improvements in benchmark results can largely be attributed to the end-to-end design of PIGEOTTO, not our final training data selection. Still, we find that including the 340,579 landmark images [39] improves our model's performance across all benchmarks and distance metrics. We further note that both PIGEOTTO versions are also more robust than Clark et al. [7]'s Geodecoder by almost an order of magnitude, reducing the median geolocalization error by more than $5x$ on the out-of-distribution (OOD) benchmark dataset GWS15k [7]. Given the benchmark and OOD results, PIGEOTTO is currently the only planet-scale image geolocalization model robust to location and image distribution shifts.

E. Auxiliary data sources

Our work relies on a wide range of auxiliary data that we can infer from each image's location metadata. This section details external datasets we are using either in the process of label creation or multi-task training.

Administrative area polygons. We obtain data on country areas from the Database of Global Administrative Areas (GADM) [10]. Additionally, we obtain data on several granularities of political boundaries of administrative areas released by The William & Mary Geospatial Evaluation and Observation Lab on GitHub. These data sources are used both in geocell label creation as well as to generate synthetic pretraining captions. The political boundaries are used in the semantic geocell creation process with Voronoi tesselations, as displayed in Figure 5.

Köppen-Geiger climate zones. We obtain data on global climate zones through the Köppen-Geiger climate classification system [4], visualized in Figure 8. We use climate zone data both for synthetic caption generation for pretrain-

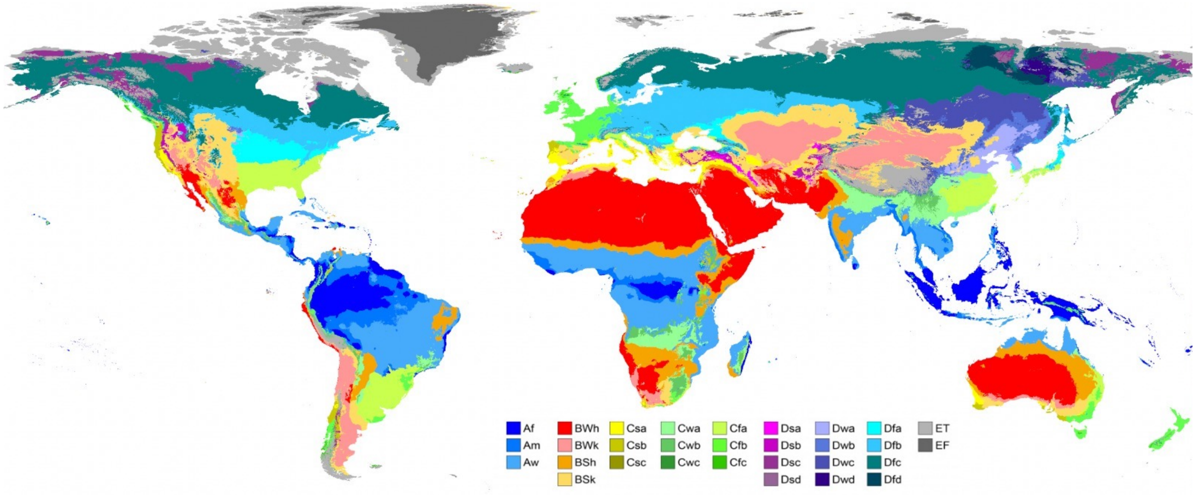


Figure 8. Map of planet–scale Köppen-Geiger climate zones in our dataset. Adapted from Beck et al. [4].

ing but also employ it in PIGEON's ablation study as a classification task (ablating "Multi-task Prediction Heads"), described in Tables 1 and 2. The final PIGEON and PIGEOTTO versions only use climate zone data as part of their CLIP pretraining captions.

Elevation. We obtain data on elevation through the United States Geological Survey's Earth Resources Observation and Science (EROS) Center. As elevation data was missing for several locations in our dataset, we augmented our data with missing values from parts of Alaska[5] and Europe[6]. We use elevation data exclusively in a multi-task prediction setting via a log-transformed regression.

GHSL population density. We obtain data on population density through the Global Human Settlement Layer (GHSL). This data is also used in a multi-task prediction setting via a log-transformed regression.

WorldClim 2 temperature and precipitation. We obtain data on the average temperature, annual temperature range, average precipitation, and annual precipitation range through WorldClim 2. Similarly to prior auxiliary data, temperate and precipitation data is used in a multi-task regression setup, however, temperature values are not log-transformed before training.

Driving side of the road. We obtain data on the traffic direction through WorldStandards. This data is exclusively employed in generating synthetic pretraining captions.

F. Ablation studies on non-distance metrics

Beyond the distance-based analysis of PIGEON described in the body of the paper, we also run ablation studies on non-distance metrics related to auxiliary data described in Appendix E. In Table 9, we observe that our final PIGEON model version actually does not perform best on non-distance metrics related to a location's elevation, population density, season, and climate. The reason for this is that PIGEON does not share trainable model weights between the multi-task prediction heads and the location prediction tasks because joint multi-task training was already performed implicitly at the pretraining stage via synthetic captions. When sharing parameters between prediction heads (ablating "Freezing Last Clip Layer"), a positive transfer between the tasks is observed and better performances are achieved on these auxiliary prediction tasks.

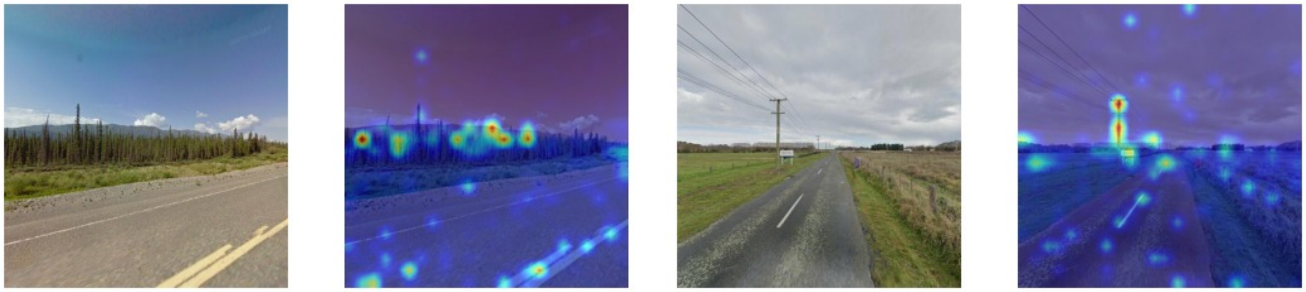
A key takeaway from Table 9 remains that geographical, climate, demographic, and geological features can all be inferred from Street View images with potential applications in climate research and related fields.

Table 9. Results from the ablation study beyond the standard distance metrics, inferring geographical, climate, demographic, and geological labels from Street View imagery.

Ablation	**Elevation Error** m	**Pop. Density Error** $^{people}/_{km^2}$	**Temp. Error** $^\circ C$	**Precipitation Error** $^{mm}/_{day}$	**Month Accuracy** %	**Climate Zone Accuracy** %
PIGEON	149.6	1,119	1.26	15.08	45.42	75.22
− Freezing Last CLIP Layer After Pretraining	**132.8**	1,072	**1.18**	**12.82**	**50.64**	**75.76**
− Contrastive CLIP Pretraining	147.1	**1,064**	1.36	14.71	45.74	74.66
− Semantic Geocells	141.7	1,094	1.37	14.48	45.74	74.10

[5] http://stacks.stanford.edu/file/druid:sg962yb7367/data.zip

[6] https://land.copernicus.eu/imagery-in-situ/eu-dem/eu-dem-v1.1/view



(a) Attention attribution map for an image in Canada.

(b) Attention attribution map for an image in New Zealand.

Figure 9. Attention attribution maps for two locations in our Street View validation dataset.

G. Additional analyses

G.1. Attention attribution examples

The contrastive pretraining used in CLIP gives the model a deeper semantic understanding of scenes and thereby enables it to discover strategies that are interpretable by humans. We observe that our model was able to learn strategies that are taught in online GeoGuessr guides without ever having been directly supervised to learn these strategies.

For the visualizations in Figure 9, we generated attribution maps for images from the validation dataset and the corresponding ground-truth caption, e.g. "This photo is located in Canada". Indeed, the model pays attention to features that professional GeoGuessr players consider important, like vegetation, road markings, utility posts, and signage, for example. This makes the strong performance of the model explainable and could furthermore enable the discovery of new strategies that professional players are not yet aware of.

G.2. Urban vs. rural performance

In order to elucidate interesting patterns in our model's behavior, we investigate whether a performance differential exists for PIGEON in inferring the locations of urban versus rural images. Presumably, the density of relevant cues should be higher in Street View images from urban locations. Our analysis focuses on PIGEON because it has been trained on many rural images, whereas PIGEOTTO was trained predominantly on user-captured, urban images.

We bin our holdout Street View dataset into quintiles by population density and visualize PIGEON's median kilometer error. In Figure 10, we observe that higher population density indeed correlates with much more precise location predictions, reaching a median error of less than 10 kilometers for the 20 percent of locations with the highest population density.

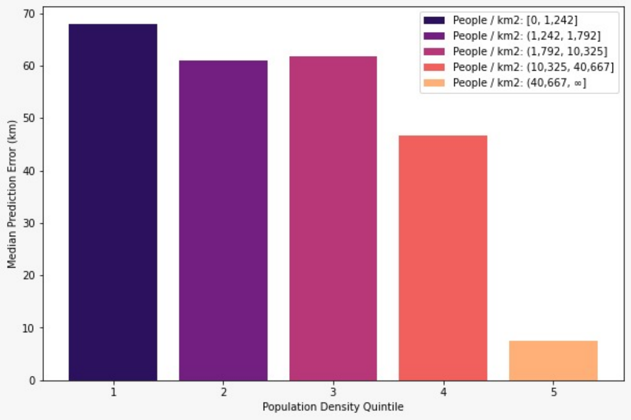
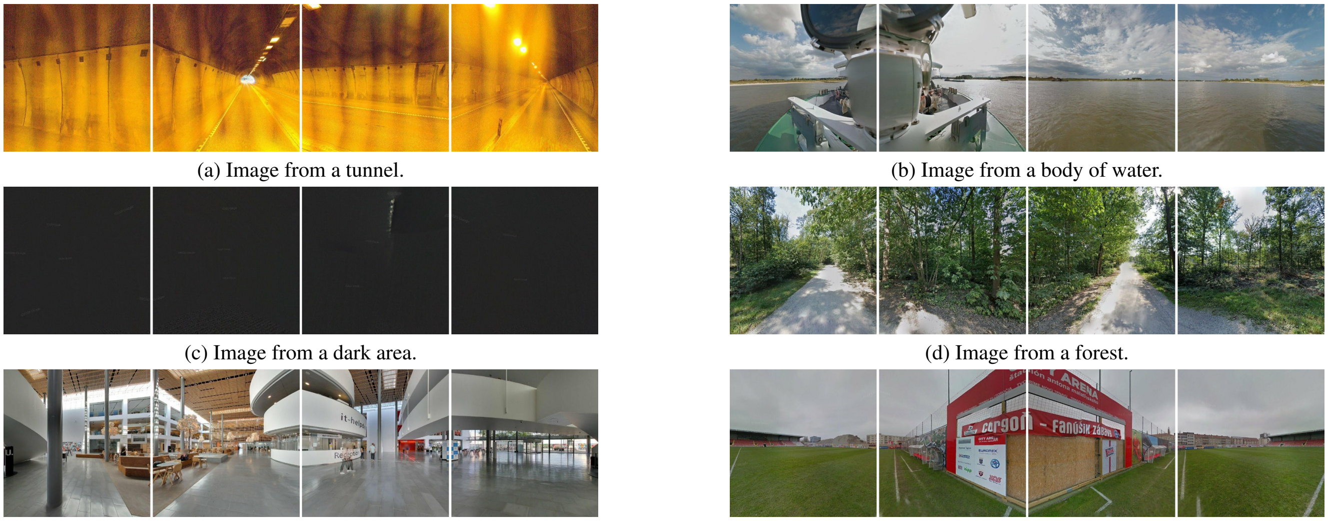


Figure 10. Median km error by population density quintile.

G.3. Qualitative analyses of failure cases

Despite our models' generally high accuracy in estimating image geolocations, there were several scenarios where they failed. We assess situations where our models were most uncertain and also identify the types of images for which our models made incorrect predictions.

Uncertainty. By computing the entropy over the probabilities of all geocells for each location in our validation set, we identified images where our models were most uncertain. For PIGEOTTO, these images were almost exclusively corrupted images remaining in the original Flickr corpus. For PIGEON, however, which was solely trained on Street View images, we observe some interesting failure cases in Figure 11. The features of poorly classified images are aligned with our intuitions and prior literature about difficult settings for image geolocation. Figure 11 shows that images from tunnels, bodies of water, poorly illuminated areas, forests, indoor areas, and soccer stadiums are amongst the cases that are the most difficult to pinpoint by PIGEON.



(a) Image from a tunnel.

(b) Image from a body of water.

(c) Image from a dark area.

(d) Image from a forest.

(e) Image from an indoor area.

(f) Image from a soccer stadium.

Figure 11. Examples of images from our test set where PIGEON was the most uncertain about the correct location.

Incorrect Predictions. While for PIGEON, most failure cases are out-of-distribution (OOD) images that are atypical for Street View imagery (Figure 11), PIGEOTTO was trained to be highly robust to distribution shifts with the goal of being a general image geolocation system. To evaluate in what cases PIGEOTTO fails, we collected representative images from the YFCC26k [25] test set and plotted PIGEOTTO's predictions against the ground truth coordinates on a map in Figure 13. We observe that real-world images (in this case derived from Flickr) are highly diverse, containing both indoor and outdoor images, vast differences in image depths, blurry images, images of people, filters that have been applied, and photos taken at night.

PIGEOTTO performs astoundingly well across a wide range of conditions; it correctly identifies popular places within circa one kilometer, as demonstrated by the third image in Figure 13 of the Capilano Suspension Bridge near Vancouver and the fourth image taken around the Kathmandu Durbar Square in Nepal. Our model fails in situations where the image contains very little information about the location, such as the fifth image containing only a boye in the sea, the sixth image containing a water bottle, or images taken at night with almost no visible features such as the eighth and thirteenth image, albeit in the later, PIGEOTTO does correctly predict Europe from the fireworks alone. PIGEOTTO further seems to work surprisingly well in indoor scenarios, even when images are blurred, as is the first image. Other examples include the eleventh image where our model still predicts the country correctly and the last image showing a person drinking from a red cup, guessed correctly to within 13 kilometers of the correct location. Finally, while PIGEOTTO confounds the wine regions of Victoria, Australia and Marlborough, New Zealand, it still is able to make accurate predictions from the flora alone, as evidenced by the close-up image of leaves, correctly guessed to within 671 kilometers.

H. Deployment to GeoGuessr

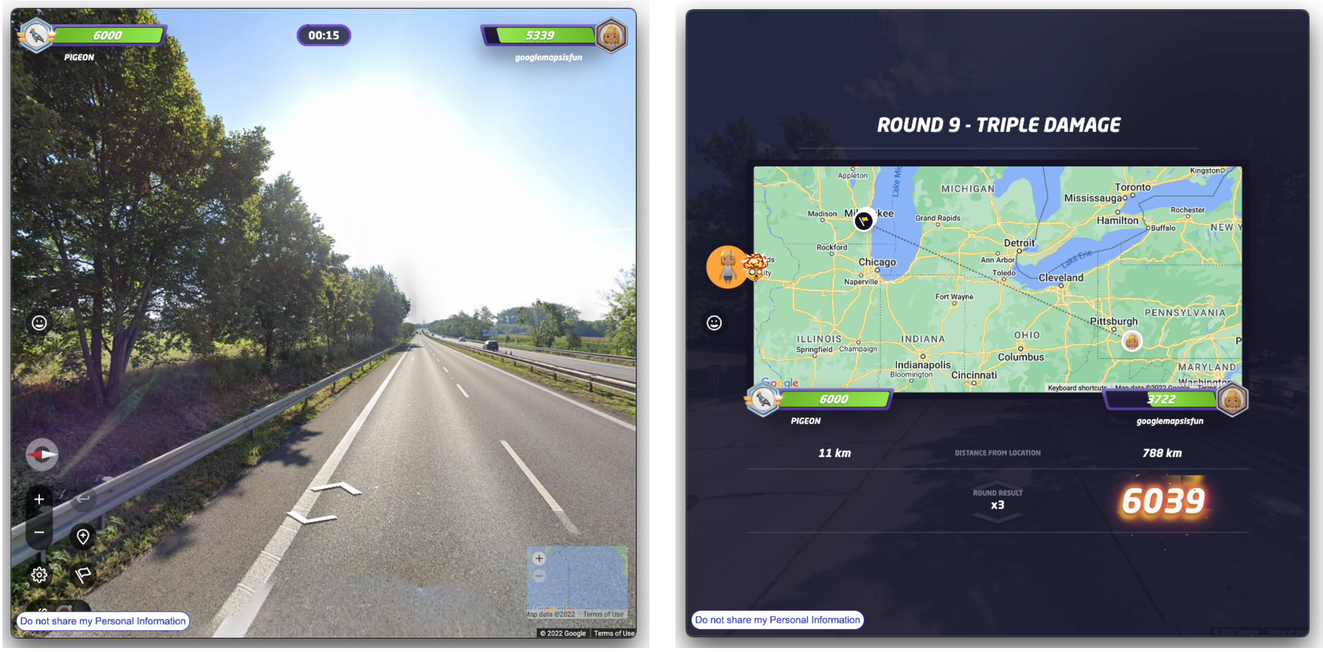
As part of our quantitative evaluation of PIGEON against human players, we develop a Chrome extension bot that uses PIGEON's coordinate output to directly place guesses within the game. This section is a high-level overview of our model serving pipeline.

H.1. Data overlap in live games

When deploying PIGEON in live games against human players, controlling for locations not in PIGEON's training dataset is impossible. Across all live games from Figure 4, we find that 1.8% of game locations are within less than 100 meters of any location in our training dataset. This slight overlap between training and live evaluation locations is not problematic because top human players would see more unique locations over the course of their GeoGuessr career, resulting in an even larger overlap between already seen and new live data for them.

H.2. Game mode

GeoGuessr can be played in both single and multi-player modes. In our live performance evaluation of PIGEON, we decided to focus on GeoGuessr's *Competitive Duels* mode, whereby the user directly competes with an opponent in a multi-round game with increasing round difficulty. Notably,



(a) Sample image from a GeoGuessr location.

(b) Comparison of a guess made by PIGEON and a human player.

Figure 12. Sample screenshots from PIGEON deployed in the GeoGuessr game.

while our GeoGuessr bot simply takes four images spanning the entire GeoGuessr panorama, other players can additionally move around in the Street View scene for at least 15 seconds which is the minimum time available to the opponent once a guess is made, resulting in them gathering more relevant information to refine their prediction. Each guess is subsequently translated into a GeoGuessr score whose formula we reverse-engineered by recording results from the game. The formula for the GeoGuessr score on the world map is approximately:

$$\text{score}(x) = 5000 \cdot e^{-\frac{x}{1492.7}}, \tag{6}$$

where x is the prediction error in kilometers.

To provide a better understanding of the GeoGuessr game, Figure 12 shows two screenshots. The screenshots were taken while deploying PIGEON in-game against a human opponent in a blind experiment.

H.3. Chrome extension

We develop a GeoGuessr Chrome extension which is automatically activated once it detects that a game has started. It then autonomously places guesses in subsequent rounds, obtaining coordinate guesses from a PIGEON model API. The procedure to place a guess in the game works as follows and is repeated for each round until one player – PIGEON or its human opponent – has won:

1. Resize the Chrome window to a square aspect ratio.
2. Wait until the Street View scene is fully loaded.
3. Repeat the following for all four cardinal directions:

   (a) Hide all UI elements.
   (b) Take a screenshot.
   (c) Unhide all UI elements.
   (d) Rotate by 90° using simulated clicks.

4. Perform a POST request to our backend server with the four screenshots encoded as Base64 in the payload.
5. Receive the predicted latitude & longitude from our server.
6. Optional: Random delay before making a guess to make the model's behaviour seem more human-like.
7. Place a coordinate guess in the game by sending a request to GeoGuessr's internal API via the browser.
8. Collect statistics about the true location & human performance and submit them to the server using an additional POST request.

H.4. Inference API

To serve image geolocalization predictions to our Chrome extension, we write code to serve PIGEON via an API on a remote machine with an A100 GPU. We utilize the Python library FastAPI to implement two API endpoints:

- **Inference endpoint.** A POST endpoint that receives either one or four images, passes them through a preprocessing pipeline and then runs inference on a GPU. In addition, it saves the images temporarily on disk for later evaluation. Finally, the API returns the latitude and longitude predictions of PIGEON to the client.

- **Statistics endpoint.** A POST endpoint that receives the statistics about the correct location, the score and distance of our guess, and human performance. This data is saved on disk and later used to generate summary statistics.

Our work demonstrates that PIGEON can effectively be applied in real-time scenarios as a system capable of end-to-end planet-scale image geolocalization.

I. Acknowledgements

We extend our gratitude to Erland Ranvinge, the Chief Technical Officer of GeoGuessr. Mr. Ranvinge's expertise and willingness to share in-depth knowledge about the GeoGuessr web API was instrumental in deploying PIGEON to GeoGuessr. His support in facilitating live evaluations against human players not only enhanced the quality of our evaluations but also provided unique insights that enriched our findings. We further want to highlight that this work was initially conceived in an independent class project as part of *CS 330: Deep Multi-Task and Meta Learning*, taught by Professor Finn at Stanford University, and subsequently evolved into the comprehensive study presented in this work. Finally, we wish to express our appreciation to Stanford's faculty and administrators who supported our work through cloud and compute resources.

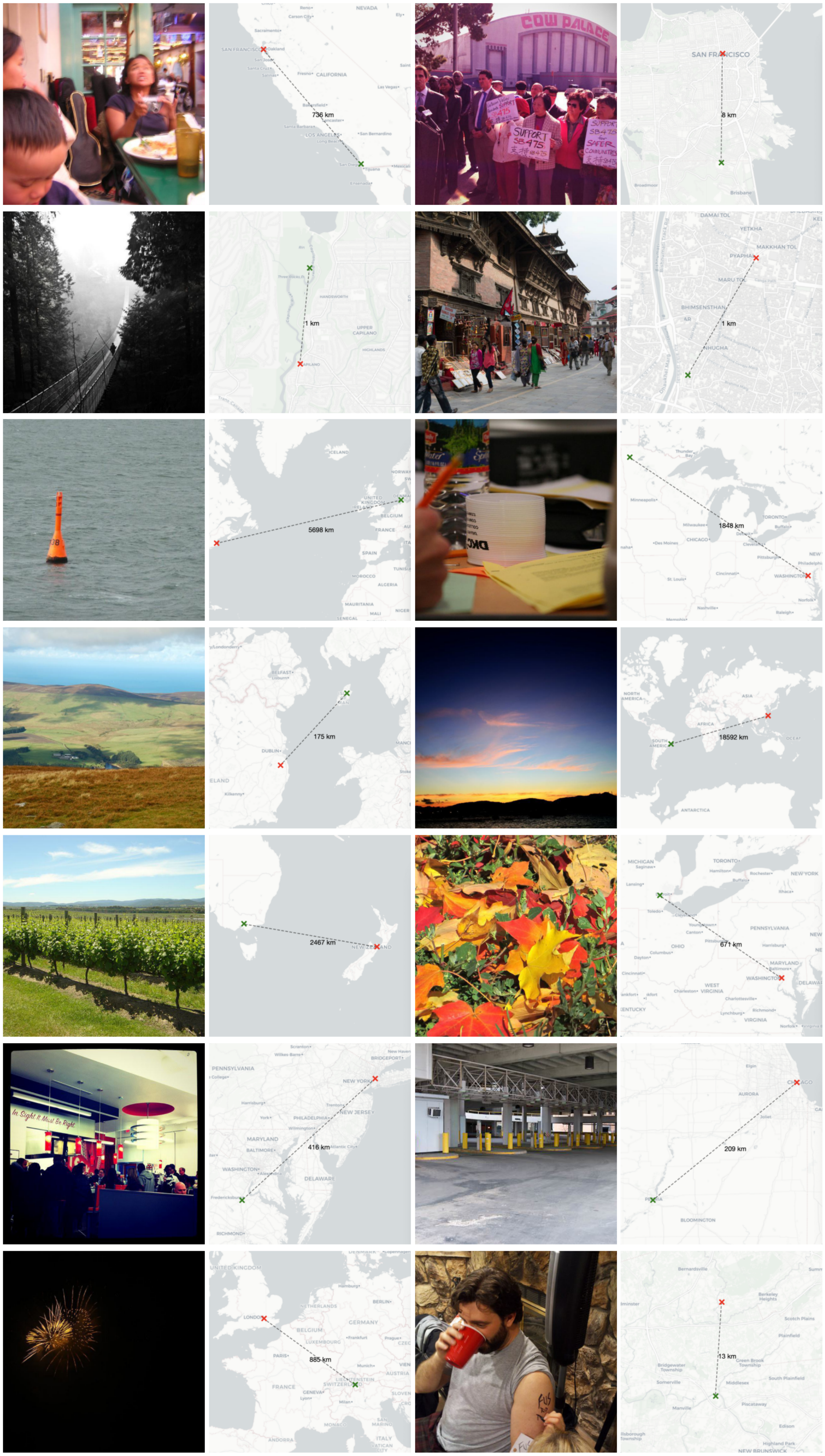


Figure 13. Example predictions of PIGEOTTO on fourteen images from the YFCC26k [25] test set.