Hybrid Geolocation Approach: Using StreetCLIP and Boundary-Based Regression

Shyam Patel

***Abstract*—Accurately geolocating images has long time been a problem computer vision experts have been trying to solve. Identifying images with no additional information except for the contents of an image seems extremely difficult. While this problem may seem insurmountable at first glance, mathematical models interpret the contents of an image as numerical representations, enabling computational approaches to address the task. In the current scope of the problem, enhancements have been made in models such as PIGEON and VIGOR, which showcase state-of-the-art enhancement and performance, by utilizing robust architectures and training on large datasets. However, these models often require significant computational resources and may not address the challenges of geolocating images in data-sparse regions. Unlike existing models, this project proposes a novel hybrid approach for geolocating images by fusing classification and regression models, using ground image features and geographic data. The primary goal was to enhance image geolocation accuracy in diverse environments, more specifically countries in rural regions where data is sparse. The results show that the implementation of boundary-based regression greatly improved accuracy by 55%, compared to individual regression and classifaction models. This approach highlights the potential of combining lightweight and interpretable techniques to address geolocation tasks effectively in diverse environments.**

***Index Terms*—Image geolocation, ground imagery, feature matching, classification, regression, geoshape, boundary-based regression, geocell.**

I. Introduction

Image Geolocation is the process of attaching precise geographical value points (latitude and longitude) associated with an image. This problem is simple to describe but much more difficult and complex to solve. Predicting precise location of imagery can greatly benefit and forward technology. Firstly, it will connect the globe by bridging isolated and rural areas together with the rest of the world by improving navigation and mapping. In terms of safety and security, it can help identify missing persons, or wanted fugitives by geolocating images.

Geolocating images is a challenging problem in computer vision, particularly when images lack distinctive landmarks or geographic cues. Existing methods, such as those using Large Vision-Language Models (LVLMs)[1] approaches or Image Retrieval via Retrieval-Augmented Generation (RAG)[1], or image classification[1]. These methods have achieved significant progress by combining visual feature extraction with retrieval techniques.

One model that has really advanced the field is PIGEON. PIGEON is a recent model published by researchers at Stanford[2] is a model that has successfully geolocates images that places it among 0.01% of GeoGuessr players globally. GeoGuessr is an online game that is aimed at correctly identifying the location a player is positioned at, given just the streetview perspective. Finding a project that is this successful and efficient at geolocating made me realize that adding a novelty to this would be very complex and very time consuming as the research team did not publish any of their material, except source code. My aim for this project was to create a small scale, simple, and creative version of PIGEON.

My approach to create a small scale, simple, and creative implementation of a geolocation model was to understand how to structure this problem. From reading articles and papers online[3] the plan started forming that a hierarchical layout was needed to get some sort of accuracy[2].

One week was spent on deciding whether to use an out-the-box model to implement and build novel features on top of, or to create a new model from scratch which would have novel techniques. Initially, the route of using an out-of-box model was chosen and PIGEON was the first model that was used. However, given that PIGEON's research team did not publish substantial details of their work[4] beyond the source code, replicating or extending their work would require considerable time and resources. Next the VIGOR[5] was explored which is an older model, but is very effective and utilizes both ground and satellite imagery, with their own visual transformer (ViT) TransGeo. The issue that was encountered with this model is that it was not small scale enough. The dataset required to train it was in total 100 GB, which was a major concern as the machine where the project was created contains limited memory. The option of Google Colab was explored - Colab is a hosted Jupyter Notebook service that requires no setup to use and provides free of charge access to computing resources, including GPUs and TPUs[6]. However, even though the premium version was used, the computations were too memory and resource consuming for any of these models to be viable to use for the project.

Thus, the option left to explore was to create a model that approached the problem in a novel direction. From the previous iterations of failed out-the-box implementations, a lot of reading and research was done to streamline the process of creating an innovative approach. The PIGEON model uses hierarchical layers and splits the into many smaller tasks that come together to produce the whole picture. In the same tone, from the early stages of the design stage, combining 2 or more models was the idea, rather than one complicated model. Anecdotally, these do-it-all single models caused a lot of headaches and frustration when trying to understand and implement, so straying away from that was ideal. Creating a novel model using both classification and regression was much more successful than the first option. However, there are still many features that can be improved and implemented to increase accuracy, due to limited memory resources and extended running time, a limited model was developed.

II. Problem Description

As computer vision evolves more, we see a lot of exploration into the field, scientists try and solve difficult problems with innovative techniques to be as accurate as possible. One problem that has been quite challenging since the emergence of computer vision is image geolocation. Which is the idea of the precise location (latitude and longitude values) of a given image. The central problem of image geolocation is determining the exact geographic coordinates of an image based solely on visual content. The task becomes particularly difficult when the image lacks clear landmarks, offers few visual cues, and the dataset is imbalanced, with limited geographic diversity in training samples. Furthermore, even when landmarks are present, the similarity of visual features across multiple regions can lead to ambiguity in predictions. For instance, architectural styles, vegetation patterns, and urban layouts in different parts of the world may resemble one another, adding a layer of complexity to the problem.

Another critical obstacle in this scope is dataset imbalance and geographic bias. Many geolocation datasets are heavily skewed toward urban areas or regions with abundant data, leaving rural or remote areas underrepresented. The lack of diversity in training samples affects the model's ability to generalize to less-documented regions, making predictions less reliable. Additionally, these datasets often focus on well-photographed landmarks, narrowing the scope of applicability and limiting the performance of models in less documented areas.

Addressing this problem not only requires technical innovations but also demands creativity to overcome the challenges of current methodologies, often revisiting previous work[2][5]. One very creative approach that has been taken is to use the GeoGuessr game. GeoGuessr is a wildly popular online game where players are placed somewhere on the globe randomly, everything they see is pulled from Google street view and the goal is to guess where you are on the map of the world. The PIGEON team actually got the most popular GeoGuessr player in the world who is among the best of the best at the game, to play against their model.[7]

III. Approach

This project explores a novel approach to geolocating images by integrating classification and regression models in a hybrid framework, enriched with geographic context such as country boundaries and centroids. By combining these features, the project aims to improve geolocation accuracy on rural areas, particularly in regions with sparse data, and contribute valuable insights to the ongoing exploration of this fascinating and impactful field. There are already published papers and articles that have taken on this problem and generated results using innovative methodologies, however a trending theme that was apparent in all the works is the great use of computational resources. Many of the approaches already explored required state-of-the-art machines and resources, in comparison the work in this project was to deployed on much smaller scale. Nevertheless the results produced are still promising.

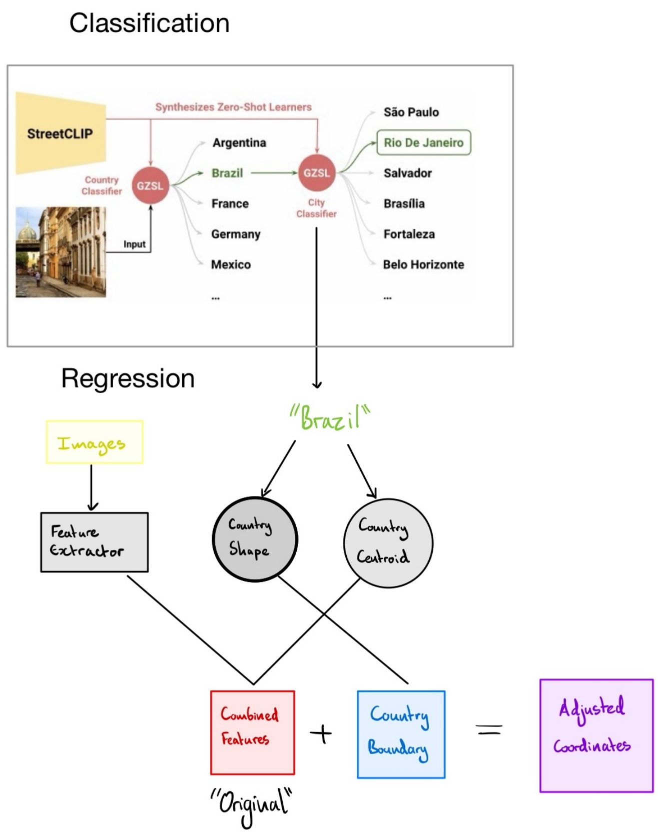


Fig. 1. The Classification and Regression Architecture implemented. An image is provided and the classification model uses Generalized Zero Shot Learning to predict the country. The prediction is then sent to the regression model, where the geoshape and centroid data is pulled. The extracted features and country centroid are merged to have an initial prediction. The country geoshape is then considered and an adjusted prediction is produced. The classification Hierarchical Linear Probing was pulled from StreetCLIP's paper[2]

A. Hybrid vs. Singular

Initially, the approach was to use singular models out-the-box and implement novelties on already successful models. However, after many days of frustration and headaches the perspective shifted away from this idea because even though the singular models, such as PIGEON[4] and VIGOR[5], are robust and highly accurate, computational resources were an issue to replicate and improve such models, as well as complexity of the models created daunting obstacles. Various attempts to try utilize the models failed, the most notable one being through Google Colab's premium subscription that provides 100.0 computational units with strong GPU's such as the NVIDIA A100 Tensor Core GPU. Even using this, processes would take upward of 4 hours to execute and all 100.0 computational units depleted in 3 days of usage.

Following the difficulties faced with using highly complex singular models, the hybrid option of incorporating multiple models started. This option was significantly preferable due to the complexity of the task being much easier to digest, designing was much simpler, and the obstacles created great learning opportunities.

B. Architecture

There are several layers involved in the pipeline of the project. The image dataset contains over 15,000 images that are each tagged with text locations of their corresponding country.[8] The metadata geoshape (stores the boundaries of all countries as polygons to validate and adjust predictions) and centroid data (provides the approximate center point of each country for enhanced regression predictions) was found online at opendatasoft.[9] The features were pulled from the image data, and combined with the centroid data for the predicted country to establish prediction.

Fig. 2. Dataset Image from Spain, coordinates (Latitude: 38.6778, Longitude: -0.19807)

Fig. 3. Dataset Image from New Zealand, coordinates (Latitude: -44.02048, Longitude: 171.42583)

1) **Preprocessing and Data Layer:** First, the Preprocessing and Data layer, which loads the image data, and resizes the images to fit the classifcation model specifications (StreetCLIP builds on the OpenAI's pretrained large version of CLIP ViT, using 14x14 pixel patches and images with a 336 pixel side length.[10]). Additionally, with memory constraints in mind, the dimensions of the feature extractor were reduced, the original iteration that was executed took 6 hours and developed 768 dimensions. The PCA function reduced the 768 dimensions down to just 4. However, the PCA function was utilized because it ensures the most valuable and rich data is stored in the 4 features, this drastically dropped out execution time - Note: performance took a dip, if memory constrains were improved, the PCA function would not be utilized.
2) **Classification Layer:** Second, the Classification layer that is the streetCLIP model, which uses image and text based inputs to classify imagery, down to city level.[10] The methodology of streetCLIP is the model is trained on over 1.1 million street-level urban and rural geo-tagged images, as well as the implementing Generalized Zero-Shot Learning[10]. Generalized zero-shot learning is the task of training a classifier to correctly predict both classes which were seen and unseen during training.[10] The use of generalized zero-shot learning enables StreetCLIP to predict for unseen classes (e.g., countries or cities not present during training). What streetCLIP does differently to normal GZSL is they introduce synthetic caption domain-specific pretraining as an additional pretraining step for CLIP both to learn how to train better zero-shot learners as well as to develop domain-specific zero-shot capabilities.[10]
3) **Regression Layer:** Third, the Regression layer plays a crucial role in the pipeline by predicting the precise geographic coordinates of an image. It operates in conjunction with the classification model, which first identifies the country most likely associated with the image. The regression model then refines the prediction to generate latitude and longitude coordinates.
   In this step the regression model merges the previous features with geographical data to further solidify prediction accuracy. By incorporating country geoshape and country centroid coordinates, the prediction are anchored spatially.
   The input to the regression model consists of the reduced feature vector and the geographic centroid. The model predicts initial latitude and longitude values based on these inputs. If the predicted coordinates fall outside the classified country's geographic boundaries, a boundary adjustment step ensures the predictions are constrained within plausible limits. This step may either refine the regression output or default to the centroid if the model's confidence is low.
   The regression task is handled by a Random Forest Regressor (RFR), an efficient ensemble learning technique which is based on decision trees as seen in Figure 24. It combines the predictions of multiple decision trees to reduce overfitting and improve accuracy.[11] Each tree is trained on a random subset of the training data and features. The final prediction is obtained by averaging the outputs of all trees, which reduces overfitting and variance.
4) **Evaluation Layer:** Fourth, an evaluation script creates different metrics that cover over 3,000 test images, which will be covered later. The evaluation covers both adjusted predictions with original predicitons and tests the model for distance to actual coordinates, split in 5 bins (50 km, 500 km, 1000 km, 5000 km, 5000

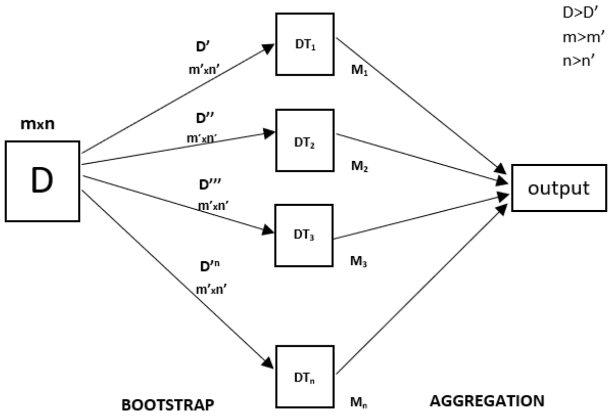


Fig. 4. The most popular implementation of Random Forest Regression is bagging also known as Bootstrap and Aggregation, it works by training multiple base models (typically decision trees) on different subsets of the training data and then combining their predictions to produce a final output. The primary goal of bagging is to reduce overfitting and variance while maintaining model flexibility.[11]

km). The evaluation covers average and total distances computed by the Haversine distance formula5 which calculates the shortest path over the Earth's surface between two geographic points, given their latitude and longitude.[12]

$$a = \sin^2\left(\frac{\Delta\text{lat}}{2}\right) + \cos(\text{lat}_1)\cdot\cos(\text{lat}_2)\cdot\sin^2\left(\frac{\Delta\text{lon}}{2}\right)$$

$$c = 2\cdot\text{atan2}\left(\sqrt{a},\sqrt{1-a}\right)$$

$$d = R\cdot c$$

Fig. 5. Haversine distance formula for calculating the distance between 2 geographical points.

5) **User Interface Layer:** Fifth, a user interface was created[1] to be interacted with, showcasing the model and geolocation results. The simple UI was created with the streamlit library and took inspiration from an article which built a similar interface[3]. The UI prompts the user to dump an image, from there the abstracted page show the predicted country, predicted coordinates, adjusted coordinates, and a map of the geolocation of the image, surrounded by the predicted country's geoshape. On the backend of this, all the scripts originally built for no UI features were shifted to taking in input files directly from the UI. When the user places an image into the prompt box, it immediately is ran through the classification model to predict the country in which the model places the most confidence of the image generating from. In the next phase, the image is then split into 4 high quality features that encompass rich data from the image, these features are then paired with country features that are identified and populated based off of the classification models prediction. Thereafter, the regression model steps in and uses all this information to make an educated guess at where the image is located. Finally, the centroid data is used to validate the location is within the country itself, and if needed regression is ran again.

IV. Evaluation

The test dataset used for evaluation contained 3,000 unique ground images from over 90 countries in the world. The image path, latitude, longitude, and 2 character country code were extracted and placed into metadata files. Using this information, the character country code, geoshape, and centroid values of a predicted country were extracted and these features were merged to create the most accurate prediction. The tests curated on the model consisted of distance comparisons by visualizations of; scatter plots comparing distances from actual coordinates to orginial and adjusted points, categorization of output into 5 bins (50 km, 500 km, 1000 km, 5000 km, 5000 km), and mean distances computed by haversine distance function.

1) **Classification Accuracy:**The evaluation process in this project is designed to assess the performance of the geolocation prediction pipeline. The evaluation focuses on two main aspects: the predicted country classification and the accuracy of predicted geographic coordinates.
The first step in the pipeline involves using the classification model (StreetCLIP) to predict the country where the input image was taken. Evaluation for this step involves calculating the proportion of images for which the predicted country matches the actual country. This is expressed as the classification accuracy:

$$Accuracy = \frac{\text{\# of Correctly Predicted Countries}}{\text{Total \# of Images}} * 100$$

This metric is critical for understanding how well the classification model narrows down the search space for the regression model. A high classification accuracy ensures that the regression predictions are geographically constrained to the correct country. Whereas a low classification indicates inherent issues with either the image data or issues during the classification process.

2) **Regression Accuracy:** The regression model predicts the geographic coordinates (latitude and longitude) of the image. The accuracy of these predictions is evaluated using the Haversine distance formula, which calculates the shortest distance over the Earth's surface between two points given their latitude and longitude. The metrics used are:
   - Haversine Distance (Original): The distance between the actual coordinates and the coordinates predicted by the regression model without adjustment.
   - Haversine Distance (Adjusted): The distance between the actual coordinates and the adjusted coordinates, where the adjustment ensures the predictions fall within the predicted country's boundary.

[1] https://github.com/shyam-185/Computer-Vision

The evaluation includes:

- Mean Haversine Distance: The average distance error for all images in the test set.I

Metric	Value
Mean Distance (Original)	6130.226226983671
Mean Distance (Adjusted)	2617.4095483990905
Closest Point (Original)	111.10917826438488
Closest Point (Adjusted)	1.0884427380291495
Furthest Point (Original)	17659.316144506556
Furthest Point (Adjusted)	19400.992881182407

TABLE I
DISTANCE EVALUATIONS BETWEEN THE ORIGINAL AND PREDICTION.

- Distance Categories: Predictions are grouped into bins based on their Haversine distance from the actual coordinates - (50 km, 500 km, 1000 km, 5000 km, 5000 km).II

Distance Category	Original Predictions	Adjusted Predictions
<=50 km	0	86
<=500 km	11	1187
<=1000 km	55	530
<=5000 km	1422	541
>=5000 km	1558	702

TABLE II
THIS TABLE REPRESENTS THE 5 BIN CATEGORIES THAT PREDICTIONS PLACED IN FOR BOTH ORIGINAL AND ADJUSTED POINTS.

3) **Original vs. Adjusted Accuracy:** To evaluate the impact of the adjustment step, the performance of the original predictions (before adjustment) is compared to the adjusted predictions.10 Metrics such as the percentage of predictions that fall into each distance category are calculated for both sets of predictions, providing insight into how much the adjustment step improves the model's accuracy.9
4) **Visualizations:** Complementing all the data and calculations, various visualization graphs were produced:
   - **Scatter Plots:** Three scatter plots were generated based of the test data. Actual vs. Original Predictions6, Actual vs. Adjusted Predictions7, Original vs. Adjusted Predictions8.
   - **Distance Distribution Plot:** Displays the proportion of predictions falling into each distance category.9
   - **Distance Line Plot:** Line graph showing the variation in Haversine distance across the dataset.10

The results display a wide variance between original and adjusted predictions, with adjusted predictions being more accurate and less clustered than original predictions.8 The actual points are scattered more or less evenly throughout the globe as the visualizations makeout in Figure 6 and Figure 767. The points roughly resemble countries and continents. For example, if you look at Figure 66 or Figure 77 you on the bottom right of the graph, the clustered points are representing Austrailia and New Zealand. In comparison, in Figure 77 the adjusted points are clustered more in the European region (middle top), however there is some deviation (a lot of points in Africa).

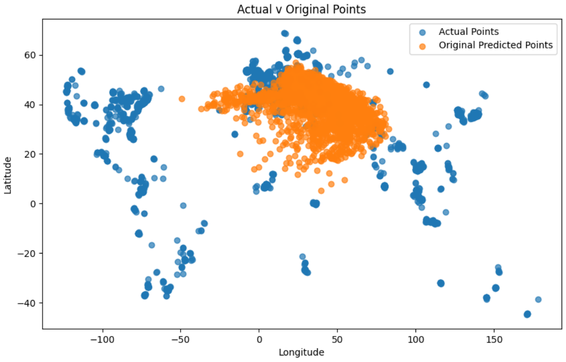


Fig. 6. Scatter Graph displaying the actual points of where the image data is located versus the original predictions.

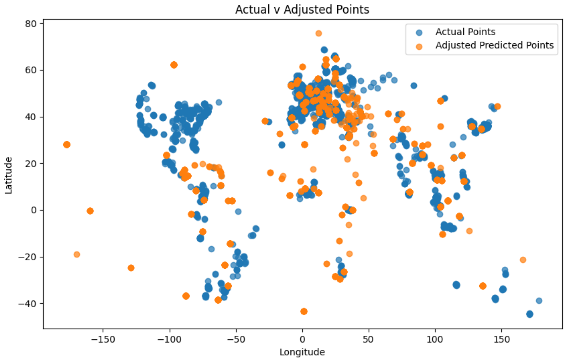


Fig. 7. Scatter Graph displaying the actual points of where the image data is located versus the adjusted predictions.

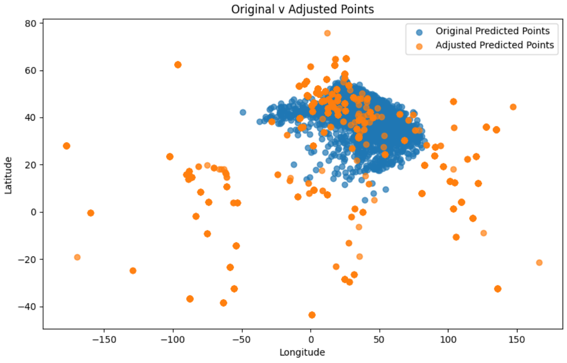


Fig. 8. Scatter Graph representing both original and adjusted predictions.

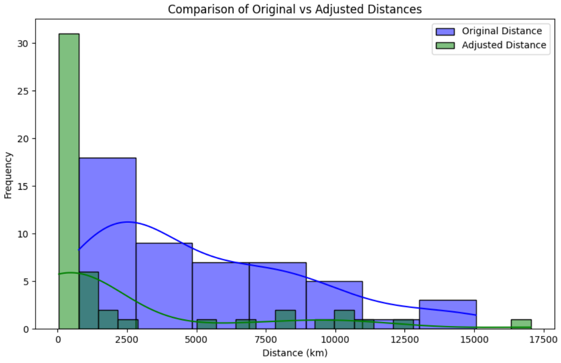


Fig. 9. Graph to visualize the difference between orginial and adjuted points, relative to actual points.

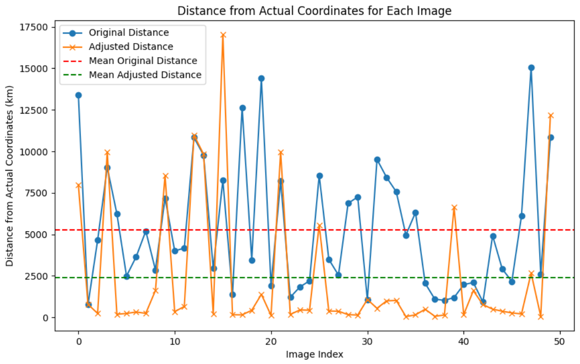


Fig. 10. Line Graph representing the difference between orginial and adjuted points, relative to actual points.

V. Related Works

1) **Image-Based Geolocation Using Large Vision-Language Models:** While this research paper is more directed toward the privacy risks and threat mitigation aspect of image geolocating from bad actors, it compares various techniques and their own 'ETHAN', which is their own model trained as a professional human GeoGuessr, to calculate the accuracy of geolocating images. It is effective in using semantic features from large vision-language models, reducing the need for extensive location-specific datasets.

   However, the paper highlights certain limitations of ETHAN. The model struggles with images that lack distinctive visual or language-based features, such as those captured in sparsely populated rural areas, dense natural environments, or imagery that lacks significant geographic markers. Additionally, its reliance on semantic features extracted from vision-language models may hinder its adaptability to scenarios where these features are not very informative. The model's performance may degrade in cases where language associations or metadata are either unavailable or misleading. Link to paper. [1]

2) **TransGeo: Transformer Is All You Need for Cross-view Image Geo-localization:** This paper introduces TransGeo, a novel transformer-based framework for cross-view image geo-localization. Unlike conventional CNN-based methods, TransGeo leverages the global modeling capabilities of transformers and introduces an innovative attention-guided non-uniform cropping strategy to reduce computational costs while enhancing performance. Their approach achieves efficiency in scenarios with domain gaps between views.

   TransGeo leverages both ground and satellite imagery to further improve accuracy of geolocation models. One of TransGeo's key innovations is the introduction of attention-guided non-uniform cropping. Attention-guided cropping reduces unnecessary computations, reallocating resources to enhance the resolution of critical areas.

   However, The model's reliance on attention-guided mechanisms and additional cropping steps introduce challenges in implementation and deployment of their model, which requires powerful computational resources and large training datasets. Link to paper. [5]

3) **PIGEON: PREDICTING IMAGE GEOLOCATIONS:** This model marked a great advancement in the Computer Vision field. PIGEON is extremely accurate in geolocating, even beating the best of the best at GeoGuessr, which is an online game where the goal is to guess your coordinates, based on the streetview placement you are in. Their approach was to combine geocell creation, multi-task contrastive pretraining, and a novel loss function.

   The main novelty introduced in this paper is the implementation of geocells, which structures the globe into hierarchical, non-uniform cells of varying sizes, depending on the density of visual and geographic features. In previous works, researchers often generalize geocells into one size across the globe, creating grids. However, even though that lead to faster results, the accuracy of doing that dropped greatly.

   The paper introduces a novel loss function designed to optimize predictions across multiple levels of granularity. This loss function ensures that the model not only prioritizes accuracy but also learns the underlying geographic relationships between regions.Additionally, the model demonstrates strong generalization capabilities, achieving state-of-the-art results on multiple benchmark datasets.

   However, PIGEON's performance heavily relies on the quality and diversity of geotagged image datasets used for pretraining. Sparse or biased data may affect its generalization. Similarly, the computational load is very large, they state it requires 4 A100 GPU's to run, with a runtime of 12 hours. Link to paper. [4]

4) **LEARNING GENERALIZED ZERO-SHOT LEARNERS FOR OPEN-DOMAIN IMAGE GEOLOCALIZATION:** StreetCLIP specializes in open-domain image geolocation, leveraging a novel synthetic caption-based pretraining methodology. This method generates domain-specific captions that serve

as training inputs, grounding the model in geographic tasks. The approach enables StreetCLIP to perform generalized zero-shot learning (GZSL), which allows classification for both seen and unseen geographies.[2] StreetCLIP also introduces a hierarchical linear probing strategy during inference, predicting first at the country level before refining predictions to city levels. Trained on a unique dataset of 1.1 million Google Street View images from 101 countries[10], StreetCLIP has demonstrated superior performance on benchmarks like IM2GPS and IM2GPS3K. The model's innovative integration of domain-specific synthetic captions and a hierarchical prediction pipeline sets it apart from prior works that relied heavily on supervised or retrieval-based methods. This allows StreetCLIP to generalize effectively across diverse geographic contexts, providing an important foundation for further exploration in geolocation tasks.

Despite its large dataset of 1.1 million Google Street View images, the model is inherently limited by the geographic coverage and diversity of the dataset. It might perform well in urban or well-documented areas but struggles in rural or less-represented regions where data is sparse or non-existent, due to the model being trained on images from only 101 countries. Link to paper. [2]

VI. Summary and Conclusion

This project aimed to engage with the challenging problem of geolocation in the field of computer vision, focusing on accurately identifying the geographic location of an image based solely on its visual content. Recognizing the complexity and scale of this task, a novel hybrid approach was developed, integrating a classification model (StreetCLIP) with a boundary-based regression model (Random Forest Regression). This fusion significantly enhanced geolocation accuracy, achieving improvements of up to 55%.

Throughout the development of this project, valuable insights were gained, not only about the specific geolocation challenge but also about the broader landscape of computer vision and artificial intelligence research. As a first attempt into conducting independent research, the process involved immense literature review, frequent trial-and-error experimentation to see which approach would work, and thorough documentation of every step, resulting in a steep yet rewarding learning curve. While the model's results may not rival state-of-the-art solutions, building a small-scale model that demonstrates potential and promise was both satisfying and fulfilling.

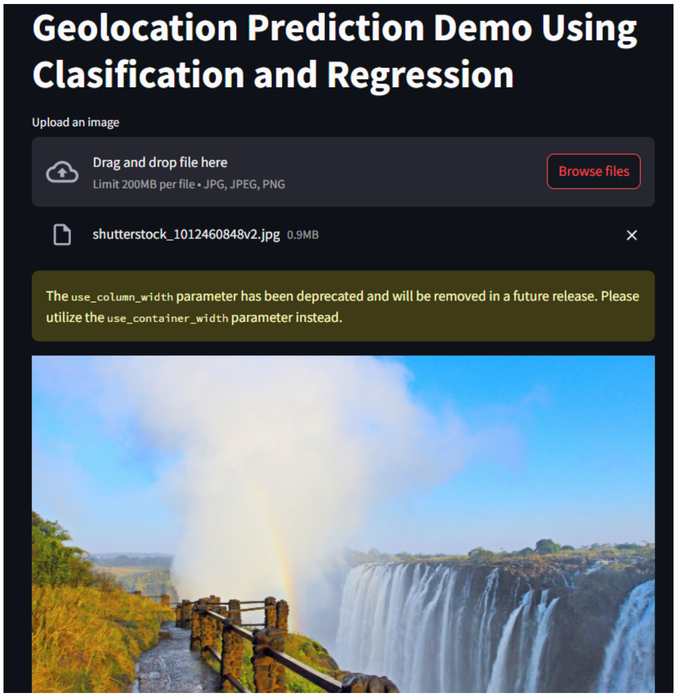
The findings from this research highlight the opportunities for further exploration in hybrid and hierarchical methodologies for geolocation. With access to a larger, more diverse dataset, expanded feature extraction, and detailed parameter tuning—such as moving beyond country-level to city-level classification — the model has the potential to achieve remarkable results. This project serves as a foundation for future advancements in this challenging domain, both on a personal and professional level.

VII. Supporting

A. GitHub Page

A GitHub repository was created for the project that consists of all source code to validate the paper: [13]. The dataset is not included but can be downloaded on Kaggle[14] or Huggingface[8]. The UI file can also be accessed there and can be deployed and tested. [2] [3] [4]

B. User Interface Demo



Geolocation Prediction Demo Using Clasification and Regression

Upload an image

Drag and drop file here

Limit 200MB per file • JPG, JPEG, PNG

shutterstock_1012460848v2.jpg 0.9MB

The use_column_width parameter has been deprecated and will be removed in a future release. Please utilize the use_container_width parameter instead.

Fig. 11. webpage1

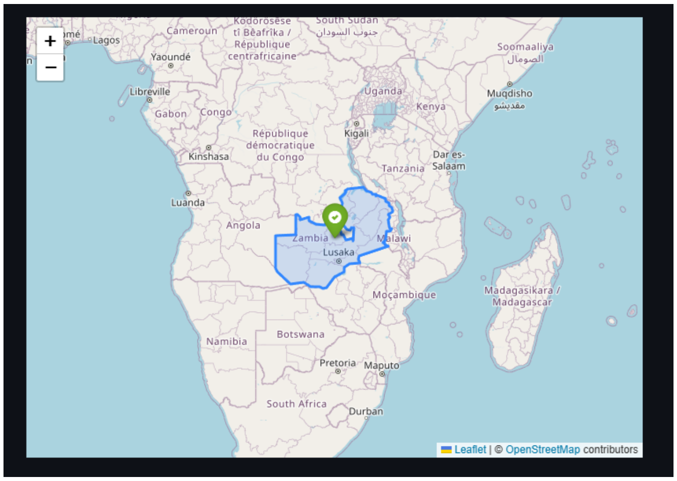


Fig. 12. webpage2

[2] https://github.com/shyam-185/Computer-Vision

[3] https://huggingface.co/datasets/rohanmyer/geotagged-streetview-images

[4] https://www.kaggle.com/datasets/rohanmyer/geotagged-streetview-images-15k

REFERENCES

[1] Y. Liu, J. Ding, G. Deng, Y. Li, Tianwei, Z. Weisong, S. Y. Zheng, J. Ge, and Y. Liu, "Image-based geolocation using large vision-language models," Ph.D. dissertation, Nanyang Technological University Singapore, University of New South Wales Australia, Institute of Information Engineering, Chinese Academy of Sciences China, 2024.

[2] L. Haas, S. Alberti, and M. Skreta, "Learning generalized zero-shot learners for open-domain image geolocalization," 2023.

[3] J. K, "Geolocation and ai with streetclip: introduction, country classification and building a web interface," *Medium*, 2023.

[4] L. Haas, M. Skreta, S. Alberti, and C. Finn, "Pigeon: Predicting image geolocations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2024, pp. 12 893–12 902.

[5] S. Zhu, M. Shah, and C. Chen, "Transgeo: Transformer is all you need for cross-view image geo-localization," Ph.D. dissertation, University of Central Florida, 2022.

[6] FAQs for Google Colab: https://research.google.com/colaboratory/faq.html.

[7] GeoGuessr Pro vs. AI: https://www.youtube.com/watch?v=ts5lPDV.

[8] Image Dataset: https://huggingface.co/datasets/rohanmyer/geotagged-streetview-images.

[9] Https://public.opendatasoft.com/explore/dataset/world-administrative-boundaries/export/.

[10] Https://huggingface.co/geolocal/StreetCLIP.

[11] A. Dutta, "Random forest regression in python," *GeeksforGeeks*, 2024.

[12] J. Louwers, "Calculate geographic distances in python with the haversine method," *Medium*, 2023.

[13] Https://github.com/shyam-185/Computer-Vision.

[14] Https://www.kaggle.com/datasets/rohanmyer/geotagged-streetview-images-15k.