Beyond the Image: A Study on Segmentation-based Image Augmentation for Reverse Geolocation

Amritpal Singh
Division of Computer Science, Mathematics and Science, St. John's University, NY, USA
amritpal.singh19@stjohns.edu

Abstract

Reverse geolocation can be extremely helpful in many cases, especially those relating to law enforcement and tracking criminal activity. Having a model that can accurately do so would be extremely beneficial to society. However, historically, most models have struggled to do this with meaningful performance with the highest being 72.4% accuracy on the country level for the Im2GPS3k dataset (PIGEOTTO (Haas et al., 2023)). In our project we compare the results of selectively segmented vs. full images being passed into convolutional neural networks to verify that it is a valid approach. We employ Meta's SAM model, ResNet, and Convolutional Neural Networks to do so. These results are compared to Transfer Learning models based on ResNet alongside StreetCLIP (Haas et al. 2023) (a contrastive learning model). Ultimately, our findings display that the approach is not effective and even reduces performance when compared to applying the same methodologies to our base images.

Introduction

Reverse geolocation is the process of using information contained within an image to determine the location where the image was captured. It can be useful in many areas, especially when dealing with the criminal justice system. Despite their being numerous models out there, the highest performance reported on Im2GPS3k (a public dataset sourced from Flicker to test the accuracy of reverse geolocation models) reported at a country level (750km) was PIGEOTTO with an accuracy of 72.4%. We attempt an alternative procedure involving augmented images through segmentation. In our use case of training on street view images, segmentation may prove to be effective at country level identification as different countries contain distinctively different objects. Figure one displays an example where utility poles are compared in four different countries. The same concept applies to other objects such as road markings, architecture, signs, etc. Building on this, we apply segmentation and create augmented images based on whether or not the segmented objects fit into our pre-selected categories and weigh the color intensities by how confident our object detection model is. The hypothesis follows that due to these differences, the models trained on these augmented images may perform better if these features are highlighted. A detailed explanation of our methodology and the resulting data will be provided in the following sections.

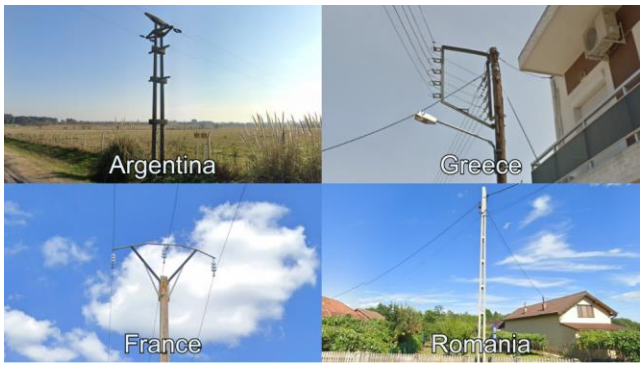

Fig 1. Comparison of Utility Poles in Different Countries (Plonk It Guide to GeoGuessr)

This study builds upon the existing body of research that has employed similar methodological approaches. Yuan implements segmentation in to recognize hotel rooms in different countries belonging to different chains (Yuan, 2022). He uses YoloV3 (trained on COCO) for bed recognition, segments the bed from the background and funnels into a transfer-learning based VGG16 model. The initial accuracy received was suboptimal, so Yuan reorganized the data into 2 classes to achieve an accuracy of 62%. Even though our research is on a different image domain (street view vs indoor

images), we try to expand on Yuan's work through using multi-segmentation on all objects present in an image to augment our dataset along with using different architectures. Banerjee attempts to use CNNs, ResNet, alongside other techniques to correctly identify the country an image is taken from, using one of the two datasets that we aggregate. From his results, Banerjee observed that a traditional CNN performed better than Transfer Learning using ResNet. Given the shared foundation of our base models with Banerjee's approach (CNNs and ResNet for transfer learning), we can expect comparable results.

The domain of reverse geolocation has witnessed the development of a plethora of models, with some demonstrating superior performance compared to others. PlaNet was a popular geolocation model created by employees at Google (Weyand et al., 2016). The approach was innovative in employing a grid approach where Earth gets divided into thousands of geographically defined cells, then through a CNN and training on millions of images, a probability distribution is generated over those cells. The methodology received a groundbreaking 53.6% country level accuracy when it was presented in 2016. There are significant differences between our two approaches. In the context of streetview reverse geolocation, StreetCLIP excels. It performed the best in most categories including in country level identification (until very recently where the model was superseded by PIGEOTTO) with an accuracy of 61.3% through the use of OpenAI's CLIP, which creates an embedding space that maximizes similarity between image embeddings and text where the country is identified. It is also very efficient to train as it enables zero-shot learning. OpenAI's Vi-T model inputs patched image embeddings and text, as such it can identify patterns between patches. In our experiment instead of just focusing on patterns, we plan to specifically segment different elements which in theory may give us better results if certain objects are identified/learned correctly. We also directly compare our results to StreetCLIP by applying it to our datasets as it is publicly available on HuggingFace.

Methods

Initial Preprocessing

To create our initial dataset, we combine two publicly available street view datasets from Kaggle. GeoLocation Geoguessr Images (K., 2021) contained 49,997 images. We maintain the folder structure (one folder for every country and each image in the respective country folder) while cropping and resizing the images to a shape of (640,640,3). The second dataset, Streetview Image Dataset (Kaushal, 2023) consists of 25,228 images. These images were already the correct shape, however we had to convert from coordinates (latitude and longitude) to countries, which we did through the use of the reverse-geocoder library (Thampi, 2016). They were then formatted in their respective country folders. We removed folders with under 75 images to prevent class imbalances, resulting in a total of 73 countries/classes and 73,867 images.

To develop the augmented image, we used Meta's Segment Anything Model (SAM) alongside ResNet. SAM allows us to break an image into individual masks/segments. We take said masks and then apply them into the base ResNet50 model trained on ImageNet (after padding and resizing) to receive the predicted classes of the object. If one of the predicted classes was part of our 128 pre-selected classes (which included objects such as poles, traffic lights, etc) and the probability of said class was over 10% (if the predicted probability is less than 10% then the object likely isn't one of the classes), we save the score into an augmented mask that we create. If the two above situations do not apply then the score for the mask is 0. The mask has a shape of (640, 640, 1), with each location representing the maximum score (0-100) at the respective pixel. We apply this mask later to develop our augmented datasets.

We also wanted to test our performance on Im2GPS3k so we downloaded the images (Vo et al., 2017) and the respective coordinates (Müller-Budack et al., 2018) and applied the same reverse-geocoder library to find the countries. We delete countries not in our dataset and format them according to the folder structure as specified prior, resulting in 2504 images.

Model Development

In order to develop our models, we first had to create our different datasets. A total of three different datasets were made. The first consisted of the original images after the preprocessing steps we described earlier. The second consisted of applying the score value from the masks directly to the original image (we refer to this later as weighted base). For example, if a pixel has a score of 60, then we would multiply all three-color channels by that value/100 (.6) resulting in varying color intensities. The third dataset was a mix of the two, that only muted the colors to a specific limit (referred to as weighted custom). The equation is below:

$$f(x) = \begin{cases} ns \ \ if \ x < 10 \\ \min + (\{x - 10\} / \{threshold - 10\}) \times (1.0 - \min) \ \ if \ 10 \leq x \leq threshold \\ \max \ \ if \ x > threshold \end{cases}$$

Eqn. 1 Custom Color Intensity Weighting Formula

The parameters we use to develop the third dataset are as follows: x:pixel score, ns: the value we receive if the pixel score is nonsignificant – set to .1, min: the minimum return value if x=10 – set to .2, threshold: the threshold at which we give the maximum score of 1.0 (keeps the same intensity) – set to 70, max: the maximum return value (returned

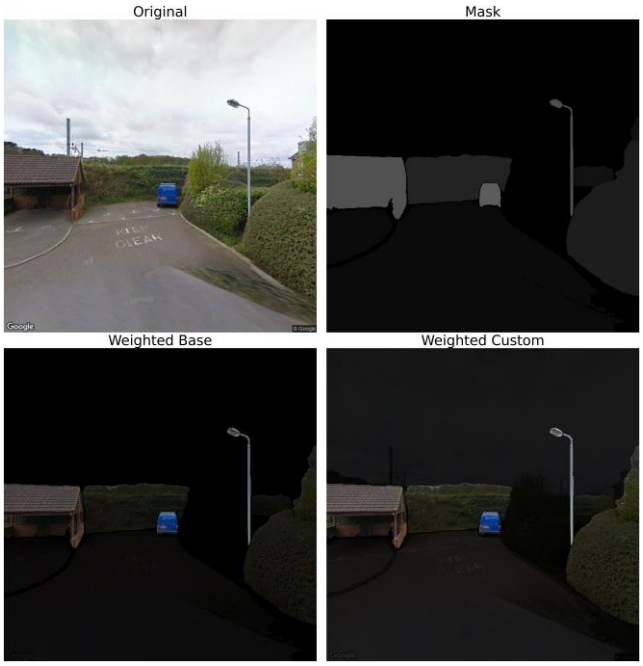

Fig. 2 Dataset Comparison

if above threshold) – set to 1.0. After developing the three datasets, we created a 80/10/10 split for training/validation/testing using the split-folders library enabling us to split our folders identically among the three datasets (ex. the train datasets contained the exact same image files, just the different versions). Figure 2 displays a visual example of the three different datasets alongside the augmented mask.

All of our models were trained on all three datasets (meaning we create three different models for each methodology) with their corresponding training and validation data and were tested on all three test sets as well. We employed keras's ImageDataGenerator and flow from directory to rescale, apply a random shear (shear_range=.2), zoom (zoom_range=.2) and horizontal flipping, introducing more variety and preventing overfitting. All models used a batch size of 64 and were trained for 25 epochs. They were compiled using the adam optimizer and employed earlystopping and reducelronplateau for further performance improvement.

The first model that we trained and tested our developed datasets on consisted of 5 convolutional blocks, 1 dense block followed by a final dense layer with softmax activation. Each Convolutional Block consists of a Conv2D layer with varying nodes and kernel sizes, followed by BatchNormalization, an Activation Layer (relu in our case), MaxPool2D layer with pool_size (2,2) and finally a Dropout layer with rate=.25. In our model, we used 32,64,128,64 and 32 nodes respectively with a kernel size of (3,3) for our convolutional layers. The dense block consisted of a dense layer with 512 nodes, a BatchNormalization layer, an Activation layer (relu) and a dropout layer with rate=.50.

The second model consisted of us comparing our performance to transfer learning. We import the pretrained ResNet50 model with fixed weights, apply GlobalAveragePooling2D, add two of our dense blocks (with 1024 and 512 units) and a dense layer with softmax activation. We apply this structure to all three of our datasets to receive three transfer learning models.

To compare our performance to other models and datasets, we import StreetCLIP from HuggingFace and apply it to all of our testing data. We also implement our models to the Im2GPS3k dataset to differentiate performance further.

Results and Discussion

In terms of our results, we evaluate each of our six trained models and StreetCLIP on our 3 datasets as well as the modified Im2GPS3k dataset. The resulting accuracies can be viewed in Table 1.

	Trained On:	Original Test Accuracy	Weighted Base Test Accuracy	Weighted Custom Test Accuracy	Im2GPS-3k -modified Accuracy
Convolutional Neural Network	Original	40.99%	11.98%	15.18%	12.82%
	Weighted Base	26.75%	26.67%	28.11%	16.13%
	Weighted Custom	24.87%	14.95%	36.10%	10.90%
Transfer Learning	Original	27.66%	3.64%	5.48%	14.30%
	Weighted Base	17.86%	20.99%	21.28%	12.86%
	Weighted Custom	19.76%	20.57%	21.73%	14.58%
StreetCLIP	Default	71.45%	21.63%	47.44%	56.31%

Table 1: Accuracy of Model Evaluation

The results from our experimentation revealed some intriguing patterns. We had initially hypothesized that our augmented dataset based on the predicted ResNet segmentation values would lead to superior performance, but this doesn't seem to be the case. We observe that for convolutional neural networks, our model trained on the original images had an accuracy of 40.99%, which outperformed the other two models trained and evaluated on their respective testing sets. Specifically, the original performed the best, then the weighted custom and finally the weighted base (40.99%>36.10%>26.67%) (we can observe the diagonals to analyze these accuracies in particular). This likely means that the models perform better if there is more information present in the image, not just the specific segments that we deem important. For example, there is ~10% improvement between the weighted base and weighted custom. When we

look back at figure 2, we see that the weighted custom image still contains the sky and other objects that are not part of our selected classes (just at a much lower color intensity), while the weighted base image completely darkens those out. This phenomenon likely contributes to a lower accuracy as there is less information present in the image. Unfortunately, due to compilation errors and time restrictions, we were unable to verify this through the use of saliency maps.

The overall performance of transfer learning was much lower than training our convolutional neural networks from scratch. This is again likely due to how the best models seem to be those that focus on the most information, not just segments/objects within the image (which ResNet identifies). As such, the same phenomenon we discussed earlier applies with original tested on original performing the best, then weighted custom on weighted custom and finally weighted base on weighted base.

StreetCLIP performed extremely well on our base images with a 71.45% accuracy, with weighted base and weighted custom being significantly lower. Since the StreetCLIP model was trained on full color images (not segmented ones), its performance across datasets is to be expected. It has the highest accuracy among all of our tests, which also aligns with our expectations as our models used a much smaller training set (around 73,000 vs 1.1 million images). As to which architecture is superior is yet to be determined as we can't directly compare, yet the contrastive learning approach may be better because of its ability to understand specific patterns, not just recognize objects.

When testing on the modified Im2GPS3k dataset, all of the accuracies are within four percentage points of each other except StreetCLIP. The close proximity of the models' accuracies makes it challenging to definitively identify the best performer on new images.

Conclusion

Our investigation into segmented image performance for reverse geolocation yielded unexpected results. While highlighting specific objects was our initial goal, models trained on original images consistently surpassed those trained on segmented counterparts. This suggests comprehensive image data, potentially including subtle details, might be crucial for accurate location recognition. Segmenting with a ResNet-based approach might have unintentionally removed valuable contextual cues. While ResNet excels in object recognition, its training on ImageNet may not encompass all objects relevant to reverse geolocation (ResNet does not identify roads, foliage, etc).

Similar to the underperformance of transfer learning models, this highlights the potential drawbacks of focusing solely on object recognition in reverse geolocation. StreetCLIP's superior performance using contrastive learning underscores the benefits of larger datasets and techniques beyond simple object identification.

Future research could explore segmentation models pretrained on datasets more relevant to street scenes or specific geographic features. Additionally, alternative segmentation techniques or even a combination of approaches might yield better results. By addressing these limitations and exploring new avenues, we can continue our pursuit of more accurate reverse geolocation models.

References

Banerjee, A. (2023, May 9). *Image Geolocation with Computer Vision*. Arsh Banerjee. https://www.arshbanerjee.com/uploads/paper/3bda9_20230723185809.pdf

Beginner's Guide to Geoguessr. Plonk It. (n.d.). https://www.plonkit.net/beginners-guide

Haas, L., Alberti, S., & Skreta, M. (2023, February 1). *Learning generalized zero-shot learners for open-domain image geolocalization*. arXiv.org. https://arxiv.org/abs/2302.00275

Haas, L., Skreta, M., Alberti, S., & Finn, C. (2023, July 11). Pigeon: Predicting image geolocations. https://arxiv.org/html/2307.05845v4

K., R. (2021, September 1). *Geolocation - Geoguessr images (50k)*. Kaggle. https://www.kaggle.com/datasets/ubitquitin/geolocation-geoguessr-images-50k

Kaushal, A. (2023, June 2). *Streetview Image Dataset*. Kaggle. https://www.kaggle.com/datasets/ayuseless/streetview-image-dataset

Müller-Budack, E., Pustu-Iren, K., & Ewerth, R. (2018). Geolocation estimation of photos using a hierarchical model and scene classification. *Computer Vision – ECCV 2018*, 575–592. https://doi.org/10.1007/978-3-030-01258-8_35

Thampi, A. (2016, September 15). *Reverse-Geocoder: A fast, offline reverse geocoder in Python*. GitHub. https://github.com/thampiman/reverse-geocoder

Vo, N., Jacobs, N., & Hays, J. (2017). Revisiting IM2GPS in the Deep Learning Era. *2017 IEEE International Conference on Computer Vision (ICCV)*. https://doi.org/10.1109/iccv.2017.286

Weyand, T., Kostrikov, I., & Philbin, J. (2016). Planet - photo geolocation with Convolutional Neural Networks. *Computer Vision – ECCV 2016*, 37–55. https://doi.org/10.1007/978-3-319-46484-8_3

Yuan, Y. (2022, March 21). *Geolocation of images taken indoors using convolutional ...* AiLECSlab. https://ailecs.org/wp-content/uploads/2022/03/TN22_05_Geolocation_of_Images.pdf

Code

The code for our research is viewable at:

https://github.com/AmritS7/Segmentation-Based-Geolocation