Image Geolocation with Computer Vision

Arsh Banerjee

May 9, 2023

Project Codebase:
`https://github.com/arsh-banerjee/GeoLocation-With-Computer-Vision`

1. Introduction

In the past decade, the rise in computing power and research into neural networks have helped researchers make significant strides in object detection, segmentation, and image generation. With deep learning and high-quality training data, neural networks are often able to achieve similar performance as domain experts in numerous tasks involving image classification. For example, a single CNN has the same performance as 21 board-certified dermatologists in classifying skin lesions and detecting skin cancer [2]. For this paper, we explore the classification task of geo-location, which involves predicting the location of an image using only pixel data. A domain expert in geo-location can utilize features such as road markings, plant species, soil types, weather patterns, and even license plates to identify the country in which an image was taken within seconds [1]. This task becomes significantly harder than traditional classification as it does not suffice to simply detect a tree, it must distinguish between an oak and a palm. A similar level of detail is required for all of the aforementioned features, for instance, license plates must be distinguishable between different regions in order to make an accurate classification. Accordingly, this paper aims to create a system to perform geo-location by implementing and evaluating differing model architectures and model types. We focus on methods that have been proven over the past years to be successful at classification tasks, such as Sift-based methods, training CNNs, and transfer learning. Quantitatively, through these implementations, we aim to create a high-accuracy model that can achieve superhuman performance and outperforms the computational baseline. Qualitatively, we hope to compare the attributes of each model with the results, understanding how the strengths and weaknesses of each particular model apply to this task. Geolocation as a computer vision technology has the potential to redefine augmented reality and navigation systems. Through this project, we hope to understand the nuances of geolocation, ultimately allowing for the creation of more effective models for this task.

2. Related Work

This section will discuss existing work regarding image classification broadly, then discuss specific implementations of geo-location models.

Researchers began working on the task of digital image classification as early as the 1970s, when textural and color features were at the cutting edge of computational tools. While useful in some simple cases, these types of features have limited discriminative power and limited robustness. Subtle changes in illumination or pose would render these simple features useful. Then, as computational power increased, the SIFT (Scale-invariant feature transform) algorithm was heavily utilized for classifying images. SIFT identifies key points in different scale-spaces and then constructs a histogram counting the gradient and direction of pixels in the keypoint region [4]. This algorithm was particularly effective as the SIFT key points were scale and rotation invariant and

robust to illumination changes, unlike prior texture and color-based features. A similar approach that builds off of SIFT is the concept of a visual Bag-of-words (BOW), which has improved performance and flexibility [5]. By clustering similar SIFT descriptors together, a one-hot vector of "visual concepts" related to a classification can be created and then utilized by a model to generate a prediction. These models were relatively successful, however, they still exhibited limited semantic understanding and had limited spatial information. With the rise of computational power and high-quality annotated datasets, CNNs have become the leading approach to image classification due to their state-of-the-art performance and ability to capture low-level features alongside complex spatial patterns [3]. CNNs have proven to be highly capable at broad image classification, achieving a 0.9 accuracy rate on ImageNet, a dataset with over 14 million images and 21,000 categories.

Naturally, then, the leading approach to image geolocation involves training a convolutional neural network on a labeled dataset of images. One of the leading models is called called PlaNet [7]. Weyand et al. describe their implementation in "PlaNet - Photo Geolocation with Convolutional Neural Networks," which involves training a CNN on a dataset of 126M geotagged photos from Flickr. With their trained model, only 30% of the test set could be predicted accurately at the country level.

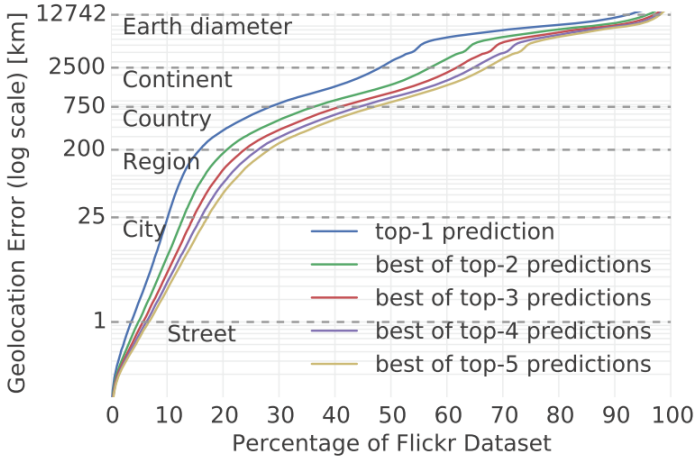

Figure 1: PlaNet's Geolocation Accuracy on a Test Set of 2.6M Images

Weyand et al. recognize that image geolocation "is an extremely challenging task" and note that only a few works have attempted to create such models. Instead of relying on a large dataset from Flickr, which has a high probability of including images not relevant to geolocation, we rely on street view images to create a model. We set PlaNet as a digital baseline to evaluate our models.

3. Approach

In order to create this geo-location system, we intended to implement a set of models that have been proven, in related work, to be successful at broader tasks of image classification. Because of the niche aspect of this task, it is not immediately clear that CNNs would perform better. For instance, a visual bag of words model may be able to capture finite differences needed for golocation (Palm vs. Oak, red road markings vs. blue ones). Specifically, we will implement and evaluate

the following models:

1. Multilayer Perceptron Classifier trained on image histograms
2. Multilayer Perceptron Classifier trained on Bag of Visual Words Concept
3. Custom Convolutional Neural Network trained on raw pixels from images
4. Transfer Learning (ResNet50) with an added Dense layer

From the related work, it appears that the SIFT-based bag of visual words model best resembles the procedure of domain experts. We hypothesize that this model will perform the best as the visual codebook is most likely to be able to differentiate between the differences in trees, road markings, license plates, and other relevant features. For similar reasons, the transfer learning model is also hypothesized to have high performance due to the synsets in ImageNet, ResNet50's source domain. Because ResNet50 was trained on a dataset that annotates the subtle distinctions needed for geolocation, it also resembles the methods of domain experts. Using accuracy, we hope to evaluate these models against one another and the baselines PlaNet and human performance.

4. Implementation

4.1. Dataset

One critical aspect of this project was the dataset which is comprised of 50k images taken from 124 countries through Google Streetview. Figure 1 displays a sample from the dataset.

Figure 2: Sample Street View Image from the US

Unlike the dataset used to train PlaNet, this dataset is much smaller in size, however, it only includes street view images as opposed to images on Flickr, which have a higher probability of not have information useful for geolocation. The link for the dataset can be found below:
`https://www.kaggle.com/datasets/ubitquitin/geolocation-geoguessr-images-50k`

4.2. Structure and Pipeline

All of the trained models and related scripts are included in the project codebase. Accordingly, the codebase contains a series of scripts related to each model implementation. We provide documentation of how each model was implemented below:

`HIST_Model.py`: We create a feature vector based off of the color histogram. Using 32 bins to divide the color space, a histogram is created from every image. We then normalize the color histograms. Then, using scikit-learn, we split the data into a training set and test set, 80% and 20% of the dataset, respectively (this split is utilized across all models). The training set is used as input to a multilayer perceptron classifier. The classifier has two hidden layers of size (256, 128) and one output layer of size 124.

`SIFT_Model.py`: Throughout COS429, SIFT was described as one of the best algorithms to identify keypoints due to its robust nature. We also discussed the viability of using SIFT features for image classification. Accordingly, after loading the images, we identify 128 SIFT features from every image using OpenCV's `'cv2.xfeatures2d.SIFT_create()'`. We then perform k-means clustering on the training data (to avoid data leakage) in order to select 200 features that appear across different images. This creates the visual bag-of-words that can be used for classification. Then, we iterate through the training and test datasets where each feature for each images is assigned to a cluster label based on its nearest centroid, or object in the bag of visual concepts. This count forms a histogram upon which a multilayer perceptron classifier is trained on. The classifier has two hidden layers of size (256, 128) and one output layer of size 124.

`CNN_Model.py`: This file contains the code defining the architecture and code to train the CNN, leveraging the Tensorflow package. The architecture of the CNN follows many of the characteristics and standards explained in the deep learning section of COS429. The sequential model contains the following architecture:

1. Convolutional Layers:
   (a) Layer One: The first convolutional layer consists of 32 filters with a kernel size of (3, 3) and uses the ReLU activation function. It is followed by a max pooling layer with a pool size of (2,2)
   (b) Layer Two: The first convolutional layer consists of 64 filters with a kernel size of (3, 3) and uses the ReLU activation function. It is followed by a max pooling layer with a pool size of (2,2)
   (c) Layer Three: The first convolutional layer consists of 128 filters with a kernel size of (3, 3) and uses the ReLU activation function. It is followed by a max pooling layer with a pool size of (2,2)
   (d) Layer Four: The first convolutional layer consists of 128 filters with a kernel size of (3, 3) and uses the ReLU activation function. It is followed by a max pooling layer with a pool size of (2,2)
   (e) Layer Five: A dropout layer with $p = 0.5$
   (f) A flattening layer then converts the multidimensional feature maps into a one-dimensional vector for a dense layer with 512 neurons and the ReLU activation function. The final dense layer was 124 neurons, which is the number of possible countries to predict. The final layer uses the softmax activation function to create interpretable outputs.

   We utilize a CNN because the Convolutional filters learn local features of the image and are able to capture high-level attributes as well. We utilize many of the design concepts from lecture, namely increasing depth through layers, smaller kernel sizes, and utilizing dropout to prevent

overfitting. In addition to dropout, each layer with weights (Convolutional and Dense) apply L2 regularization with a coefficient of 0.001 to its kernel. This sequential model is trained on the training set, after it is normalized and resized to 512 × 220 pixels due to computational limitations. The network is trained for 30 epochs.

`Transfer_Model.py`: In addition to creating a CNN from scratch, we also evaluate the viability of transfer learning for this task. Since there is limited access to labeled data, we leverage this pre-trained model and its generalization ability for this task. We utilized the pre-trained ResNet-50 model and its associated imagenet weights. All layer weights are frozen,d and two dense layers are appended:

1. Layer One: Hidden Layer with 2014 neurons and ReLu activation

2. Layer Two: Output layer with 124 neurons corresponding to output laters and softmax activation

This model is trained after the input data is normalized and resized to 512 × 220 pixels due to computational limitations. The network is trained for 30 epochs.

5. Evaluation

5.1. Quantitative

In order to evaluate the success of the different models quantitatively, classification accuracy is used (number of correct predictions / total number of predictions). We begin by evaluating the accuracies of the different models in Table 1.

Histogram	SIFT	CNN	Transfer Learning Model
0.2677	0.24	0.375	0.255

Table 1: Classification Accuracies of the Different Models

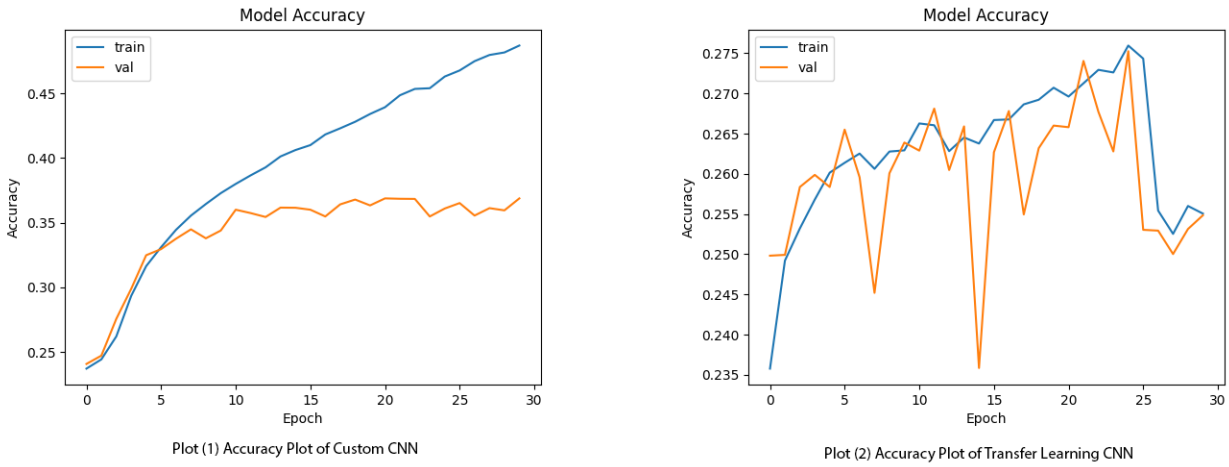

Figure 3: Training and Validation Accuracies of the CNNs

The final accuracies of each of the models are noted in Table 1. We see that the CNN trained on the images performed the best, followed by the histogram model. From Figure 3, we can see that the CNN trained from scratch has evidence of overfitting as the training and test set accuracies

diverge implying that a higher overall accuracy may be possible. In regards to the transfer learning model, also shown in Figure 3, the curves indicate that the model is not overfitting and just simply cannot generalize the dataset further with its current architecture. Still, given the relatively high accuracy of the "custom" CNN on the training set the model fulfills the initial goal of creating a viable classifier for this task.

Earlier, PlaNet was set as a baseline model for the geolocation task, however, since the models have different training and test sets, a comparison of model accuracies would not be significant. Instead, we leverage the platform GeoGuessr (`https://www.geoguessr.com/`) to conduct this analysis. Within GeoGuessr, players are given a Google Streetview image and then asked to determine the location the image was taken. This directly matches the machine learning task we approached in this project. When PlaNet was applied to the Geogussr task, it had an accuracy rate of 34% compared to this paper's CNN, having a 37.5% accuracy [7]. Weyand et al. also found that humans had an accuracy of 22%, thus, both PlaNet and this CNN achieved superhuman performance. Still, neither model comes close to achieving the accuracy of domain experts. Some players of Geoguesser routinely predict 30 countries in a row, with the world record being over 250. Given this model's accuracy of 37.5%, the chance of reaching even the average level of domain experts is $(0.375)^30 = 1.663e - 13$, which clearly illustrates how there is room for improvement for geolocation models in future work.

5.2. Qualitative

Through a qualitative analysis, we hope to discuss how the known qualities of the different models correlate to their displayed performance. One surprising finding is that the model trained on histograms outperformed that using a visual bag of words. A visual bag of words model is often expected to perform better due to its ability to capture spatial information and be more robust to variability due to its more informative representation. To understand this discrepancy, we plot the average image for two classes, the United States and Brazil, shown in Figure 4.

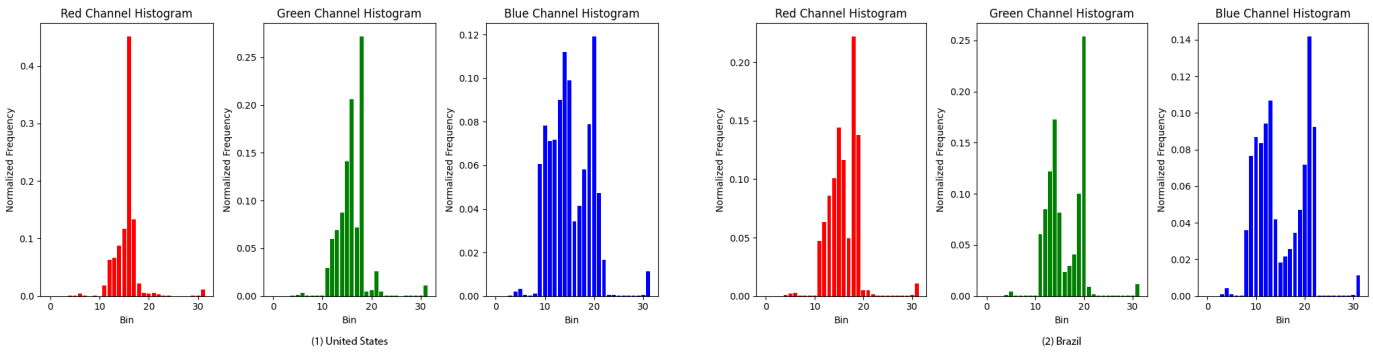

Figure 4: RGB Histogram of the United States and Brazil

While this figure only shows the average RGB histograms for two classes out of 124, it still shows a clear difference, as Brazil has noticeably more green and red hues. The differing geographies and landscapes of countries lead to distinctive color characteristics. For instance, Brazil has highly oxidised soil which leads to a distinctive red color while also having more vegetation. The histogram feature likely outperforms SIFT because these broad differences in classes can be represented via

a histogram, whereas SIFT may be finding the "visual concepts" in all image classes. Another interesting result is the poor performance of transfer learning for geolocation. In many cases, transfer learning is also capable of creating a model that outperforms other CNN and hybrid learning approaches [6]. Here, however, that is not the case, as the transfer learning model performed less effectively than a histogram model. Fundamentally, the strength of transfer learning is to leverage data from related domains and apply it to a target domain that may not have as large of a sample space. When the source domain of a model like ResNet50 differs from the target domain (Google Street View), then transfer learning may not be as successful. ultimately, this analysis reveals for a specialized classification task like geolocation, training a CNN from scratch will lead to the best results.

6. Conclusion

6.1. Effectiveness and Insights

As artificial intelligence becomes more integrated with cameras and smartphones, the ability to geo-locate an image can produce significant benefits for emergency services, travelers, navigation, and security. This research project stemmed from the idea of creating a CNN to replicate domain-expert-level performance at the image geolocation task. While the hypotheses regarding the models did not prove to be true, the CNN trained did outperform both the baseline model PlaNet. Moreover, while it also did not surpass expert-level performance, it did surpass the average human. The qualitative analysis revealed that replicating the manner in which experts perform the task may not be the most beneficial in terms of performance. Instead, simple features like the color histogram are more apt to capture the broad differences between locations. The quantitative and qualitative analysis also reaffirmed the effectiveness of convolutional neural networks. While being a black box, its ability to capture both low-level details and high-level patterns gave it the highest predictive accuracy. While this CNN may not be able to be deployed broadly due its limited dataset and low accuracy, this project made concrete steps toward illustrating how a geolocation model can be constructed and what features may be successful.

6.2. Limitations and Future Work

One key crucial aspect to increasing the accuracy and generalizability of the model is to retrain on an expanded dataset. The model PlaNet was trained on a dataset of 126M geotagged images while this project only utilized a dataset with 50k images. Google street view essentially has limitless training data with geotagged locations which could be leveraged for the model. Another potential extension would be to create an ensemble model which utilizes broad features like histograms as well as CNNs for prediction. Ensembles utilize different models, which capture and generalize different attributes. By combining these approaches, it may be possible to create a model with better performance.

7. Honor Code

I pledge my honor that this paper represents my own work in accordance with University regulations
/s/ Arsh Banerjee

8. References

[1] Kellen Browning. *Siberia or Japan? expert google maps players can tell at a glimpse.* July 2022. URL: https://www.nytimes.com/2022/07/07/business/geoguessr-google-maps.html#commentsContainer.

[2] Andre Esteva et al. "Dermatologist-level classification of skin cancer with deep neural networks". In: *nature* 542.7639 (2017), pp. 115–118.

[3] Salman Khan et al. "A guide to convolutional neural networks for computer vision". In: *Synthesis lectures on computer vision* 8.1 (2018), pp. 1–207.

[4] Qilong Li and Xiaohong Wang. "Image classification based on SIFT and SVM". In: *2018 IEEE/ACIS 17th International Conference on Computer and Information Science (ICIS)*. IEEE. 2018, pp. 762–765.

[5] Teng Li et al. "Contextual bag-of-words for visual categorization". In: *IEEE Transactions on Circuits and Systems for Video Technology* 21.4 (2010), pp. 381–392.

[6] Manali Shaha and Meenakshi Pawar. "Transfer Learning for Image Classification". In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*. 2018, pp. 656–660. DOI: 10.1109/ICECA.2018.8474802.

[7] Tobias Weyand, Ilya Kostrikov, and James Philbin. "Planet-photo geolocation with convolutional neural networks". In: *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*. Springer. 2016, pp. 37–55.