RecogNYCe - PREDICTING IMAGE GEOLOCATIONS IN NEW YORK CITY

Vasileios Panousopoulos, Flor Sanders

Columbia University

ABSTRACT

This study delves into the realm of image geolocation, focusing on New York City, and explores the application of transfer learning. The RecogNYCe dataset is introduced, featuring street-level images from different boroughs. Various pre-trained models, including ResNet18, ResNet50 and RegNet Y 1.6GF, are scrutinized alongside a benchmark custom CNN. Human performance is evaluated through a web-based guessing game. Despite challenges posed by an unfiltered dataset, pre-trained models exhibit notable success, surpassing even the best human players. The impact of pre-training on ImageNet or Places365 on model accuracy is found to be minimal. These findings underscore the prowess of deep learning models in this geolocation task and suggest potential enhancements through dataset refinement.

***Index Terms*—** Deep Learning, Geolocation, Transfer Learning

1. INTRODUCTION

Geoguessr and similar geolocation guessing games have gained massive popularity since their inception in 2013 [1]. The concept of these games is a relatively simple one. Based on a panoramic street-level view, players need to guess where around the globe the imagery was captured. While attempts have been made to beat the game using computer algorithms, human performance long proved superior.

Although a lot of previous works have tried to address the task of location recognition with computer vision feature-based methods, the recent shift towards deep learning has motivated researchers to implement such models to tackle the geolocation challenge. An early attempt to use Convolutional Neural Networks (CNN) for place recognition was proposed in [2]. In this work, a custom CNN architecture was developed and used to obtain descriptors of images, which are known to achieve good performance in place recognition. Although this model proved to outperform state-of-the-art models of the time, it was still feature-based and was focused on image retrieval.

The problem of planet-scale image geolocation has proven to be a difficult one. The first actual Deep Learning model developed for global geolocation was PlaNet [3], which was based on the Inception architecture. More specifically, in this work the earth was split into a set of geographical cells that defined the classification categories. The CNN was trained on a very large dataset of 91M training samples and was able to outperform humans in 28 out of 50 rounds, each consisting of 5 different panoramas.

DeepGeo [4] was developed in an attempt to perform location prediction in the United States, namely to classify images into 50 different states, without the need for large datasets. The model was based on the 50-layer variant of Residual Network (Resnet) [5] and was trained with panoramic samples, consisting of 4 images taken at the same location and oriented in the cardinal directions and taken from the custom 50States10K dataset. DeepGeo was able to beat human players in 4 out of 5 rounds, each consisting of 10 different locations.

PIGEON [6] was recently proposed and consists the state-of-the-art in planet-scale image geolocation. It is based on the CLIP vision transformer and it is trained on a newly created dataset consisting of 400,000 images. It achieves outstanding results, as it can correctly guess the country in which an image was taken with accuracy of 91.96% and places over 40 % of guesses within 25 km of the true location. It has been ranked in the top 0.01% of all GeoGuessr players.

Motivated by related research, the objective of this work was to investigate for the first time the performance that can be achieved with transfer learning in such a challenging task. More specifically, the problem of image geolocation was scaled down in the city of New York and a custom dataset, consisting of street-level images captured within the borders of the city was constructed. Several famous pre-trained models were fine-tuned on this dataset and their ability to correctly guess the borough in which the images were taken was tested against humans. Our code is made available for reference on Github[1].

The rest of this work is structured as follows. In Section 2, the construction of the RecogNYCe dataset is discussed. Section 3 handles the guessing style game web application developed to obtain a human performance benchmark. In Section 4 the neural network models used for the classification problem are introduced. Section 5 outlines the performance results of the classification models and compares these to several benchmarks, including human performance. Finally, in Section 6 conclusions are drawn.

[1] https://github.com/FlorSanders/RecogNYCe

2. DATASET

To make image localization in New York City possible, a custom dataset had to be created, as, to the best of our knowledge, there is not such dataset currently available online.

2.1. Location Sampling

This work studies the problem of predicting in which of the five boroughs of New York City (Bronx, Brooklyn, Manhattan, Queens and Staten Island) an image was taken. We started from a geojson-description of the boundaries of New York City's boroughs, which was sampled using a Monte-Carlo sampling technique. As such, we created a collection of coordinates (latitude, longitude) which are uniformly distributed across the areas of interest, making the number of coordinates for each of the borough is directly proportional to the area of that borough. One such sampling outcome is presented in Figure 1.

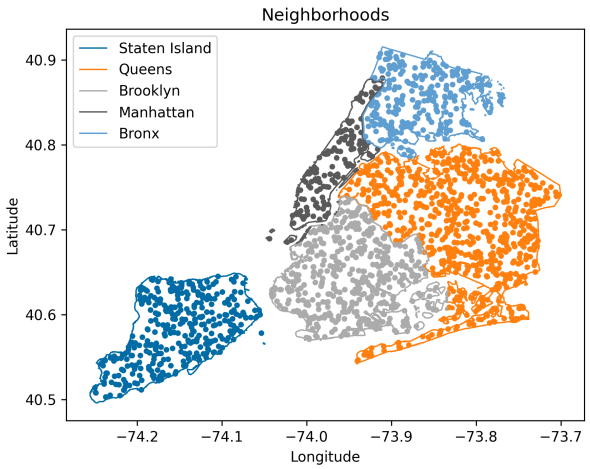

Fig. 1: Map of coordinates sampled within the NYC boroughs.

2.2. Data Scraping

Next, the sample coordinates were used to scrape two different street-level image sources, Google Street View, a commercial platform, and Mapillary, a crowd-sourced alternative. Images were sourced from these platforms by making requests to their respective APIs [7, 8]. If no image is available within a bounding box of 50 meters around the requested point, the coordinate was dropped. Finally, all Mapillary images were processed (cropped and scaled) to a 640×640 resolution, which is the maximum that can be requested through the Google Street View API.

2.3. Dataset Description

In total, we collected 38,721 street-level images captured within New York City. Among these, 17,258 originate from Google Street View and the remaining 21,463 from Mapillary. Table 1 shows the distribution of samples in the five boroughs, which is determined by their respective size.

Table 1: Sample distribution in different boroughs.

	Bronx	**Brooklyn**	**Manhattan**	**Queens**	**Staten Island**
Samples	5,256	9,245	5,217	14,693	5,122

In Figure 2 a set of random samples is displayed. Note that apart from cropping and scaling, due to the large number of samples obtained, no further pre-processing to discard bad images (i.e. first row and fourth column in 2) has been performed, nor was a data augmentation strategy implemented in the current project.

Fig. 2: RecogNYCe dataset samples.

3. GAME

One of the main objectives of this work is to compare the models' performance against humans in the classification problem. To facilitate the collection of this data, a simple web application was developed that implements a guessing style game. In each round of this game, the user is presented with one of the images from the test set used for model training (see Section 4), drawn at random. The player is then expected to guess what borough it was taken in by pressing

the corresponding button. Once this action is performed, the application logs the result, consisting of the image ID, the guess, the correct answer and a unique player identifier[2], to a CSV document for later processing. Simultaneously, the cumulative score of the current playing session is shown in real time to the user. In order to collect as many data points as possible, we allow for playing sessions of unbounded duration.

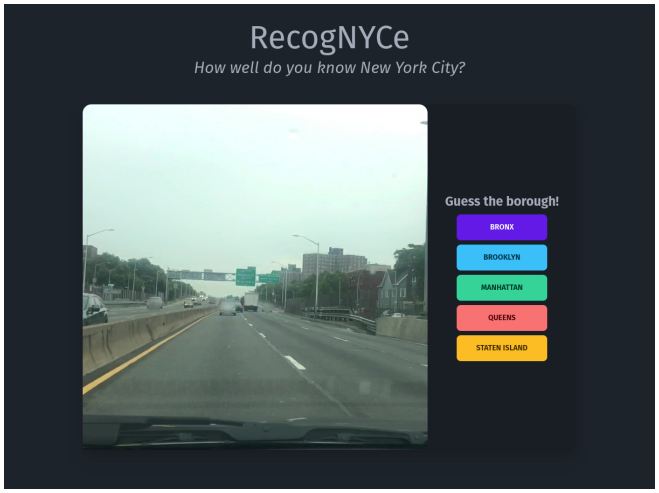

Fig. 3: UI of the RecogNYCe guessing game.

4. MODELS

In this work, we explore a variety of models and their performance on the image geolocation problem.

4.1. Benchmark model

First, a shallow custom CNN is constructed and used as a baseline case to understand the complexity of the problem. In particular, this architecture consists of 2 convolutional layers deploying 5x5 kernels and producing 6 output channels, each followed by a ReLU activation and a max pooling layer which downsamples the frames by 4, and 2 linear fully connected layers. An overview of this architecture is given in Figure 4.

4.2. Transfer Learning Models

To fully comprehend the performance that can be achieved with transfer learning in this application, a collection of models pre-trained on two different datasets are implemented.

4.2.1. ImageNet

First, two variants of the famous ResNet architecture proposed in [5], specifically ResNet18 and ResNet50 which consist of 18 and 50 layers respectively, were deployed. ResNet was chosen as it is considered one of the most successful CNN in image classification. To foster low training times, we adopted one variant of the regular networks described in [9], RegNetY 1.6GF. This architecture features a relatively small number of parameters and typically displays good performance on ImageNet. All three models contain only one fully connected layer, which was replaced by an appropriate linear layer with five outputs and random weights.

[2]Though the player ID is randomly generated to ensure user privacy, this identifier is stored in the browser's local storage to enable tracking the same user over multiple playing sessions.

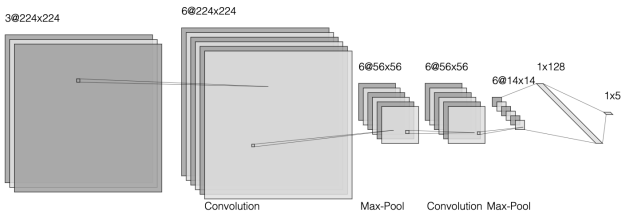

Fig. 4: Custom baseline architecture.

4.2.2. Places365

In order to explore how the dataset used for pre-training a model affects its performance in another domain, we also used the ResNet18 architecture pre-trained on the Places365 dataset. This dataset consists of 1.8M images classified in 365 scene categories and is the first scene-centric dataset, as opposed to ImageNet which is object-centric. Previous works [10, 11] have shown that the use of this dataset in scene recognition applications can lead to considerable improvement in accuracy. Considering that the problem studied in this work requires to classify not an object, but a more generic scene, using a model pre-trained on Places365 could result in better performance.

4.3. Training Process

For training, the dataset was split into training (80%), validation (10%) and test sets (10%) and batches of 100 samples were used. Also, to facilitate loading the samples to memory, the images with original resolution of 640×640 are resized and centrally cropped, resulting in images of 224×224 pixels, which is the default input resolution for models we considered.

As explained, in order to transfer each pre-trained model to our classification problem, their output fully connected layers were replaced with corresponding layers with appropriate size. Furthermore, two different training methods were employed. On the one hand, the main body of each network was initially kept frozen to train only the newly added linear layer and then, the models were fine-tuned by training all the parameters. On the other hand, the method of training directly all of the parameters was also examined. Overall, we observed that regardless of the training method, the models were

able to converge and achieve the same accuracy. It should be noted that keeping the parameters of the convolutional layers frozen made training considerably slow and the models were able to converge to optimal performance only after unfreezing these parameters. This behavior is explained by the fact that our classication problem and corresponding dataset are significantly different to the ones on which the models were pre-trained, and as result, the entire architecture should be trained on it.

Moreover, two different optimization algorithms are compared for the training process, the classic minibatch Stochastic Gradient Descent (SGD) and the Adam algorithm. As Adam uses a combination of gradient scaling and momentum, it is considered more robust than SGD for many training problems. Unfortunately, in this case no difference in either convergence or final performance was observed. Rather, the increased computational complexity of the Adam algorithm seemed to slow the training process considerably. The effects of the learning rate for SGD were also explored. For our models, a rather high training rate of 0.1 displayed considerably better results than when running with more conservative values of 0.01 or 0.001. Lastly, we observed that the models reach their maximum performance after 5 epochs of training with a learning rate of 0.1 and that they suffer from overfitting after that. To mitigate overfitting we used early stopping.

In conclusion, the models were trained for 5 epochs on batches of 100 samples and and the weights were updated with the SGD optimizer and a learning rate of 0.1.

5. EXPERIMENTAL RESULTS

5.1. Accuracy Results

After training, the models were evaluated on the test set which was also used in the RecogNYCe guessing game, where each human player encountered randomly drawn samples. The performance of the trained models compared to the human players is displayed in Figure 5.

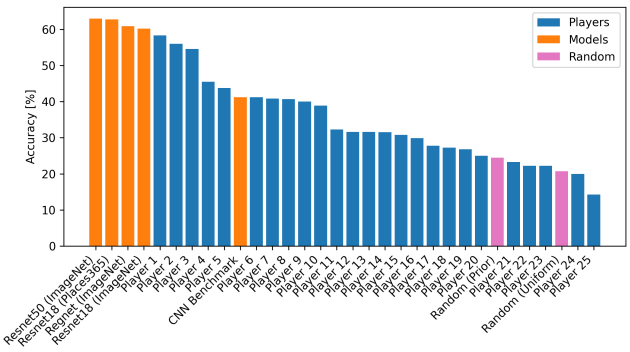

Fig. 5: Accuracy score of the models and game players.

It is clear that all of the pre-trained networks were able to achieve similar maximum performance, which saturated at around 60%, exceeding random performance by a factor of 3 and outperforming even the highest-scoring human player. On the contrary, the simple custom CNN architecture is not able to achieve these levels of accuracy as it converged to a maximum performanc around 40%.

A number of conclusions can be drawn when comparing the performance of the transfer learning models. For one, when trained on the same dataset, the larger ResNet50 model indeed scores better than its smaller ResNet18 counterpart, albeit by a slim margin, indicating a hard cap on achievable performance. Secondly, the version of ResNet18 that was pre-trained on the Places365 dataset indeed scores some percentage points better than the ImageNet pre-trained counterpart. While this difference is very small, it exemplifies that a good choice of pre-training can have a positive influence. Finally, considering its small number of parameters, the comparable performance of RegNet Y 1.6GF is notable. This enforces our view that training larger models is not the key to obtaining better performance on this classification task.

Fig. 6: Resnet-18 model misclassified samples.

In Figure 6 a set of 10 images misclassified by ResNet18, pre-trained on Places365, is displayed. Considering that several of these misclassified images are badly captured shots (i.e. nightly images or pitures of bushes and trees) we conclude that the performance of the examined models is probably affected by the unfiltered dataset. We believe if the dataset is carefully reviewed and bad images are discarded, the accuracy could be even higher. Furthermore, we observe that the remainder of the presented images correspond to samples that are objectively hard to classify.

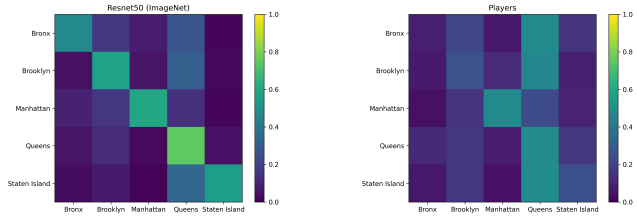

Fig. 7: Classification confusion matrices.

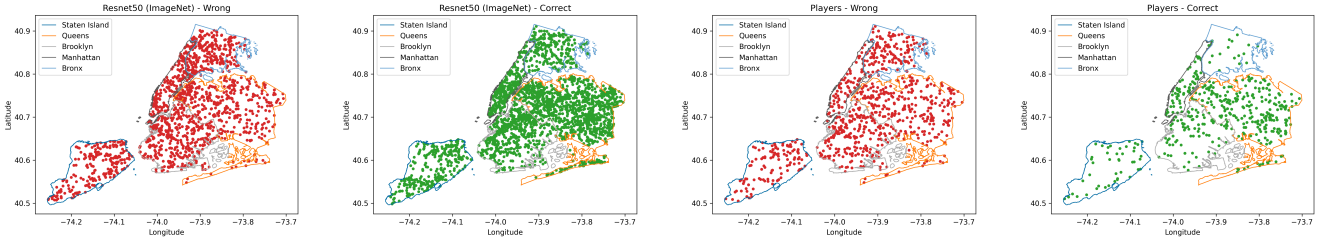

Fig. 8: Maps of sample classifications.

Figure 7 presents the confusion matrices that correspond to classification performance of ResNet50 and human players. The Resnet50 model displays the best performance when classifying Queens, with all other boroughs trailing behind with similar performance. Notably, when the model makes mistakes for these remaining classes, the borough it is most often confused with is Queens. This is a result of the large area of Queens with respect to the other boroughs, causing it to have a large prior probability. In the human player confusion matrix this tendency seems even stronger, where Queens is consistently classified correctly and the other boroughs are equally often mistaken for it. One exception is Manhattan, which shows similar classification accuracy as Queens. The probable reason for this is the familiarity of the players with this particular borough and the recognizability of its downtown streetscapes.

Lastly, Figure 8 provides the exact locations on the NYC map of correctly and wrongly classified samples by humans and ResNet50. From the model's maps two interesting patterns can be extracted. On the one hand, a lot of misclassifications are located close to the borders of the boroughs, because these boundaries are not as fixed as they would seem to be when looking at a map, but rather flow over into each other gradually. On the other hand, in areas like downtown Manhattan and Long Island which feature unique characteristics, such as skyscrapers or the beach respectively, the classification accuracy seems to be higher. As confirmed previously by the confusion matrices, the players tend to make different kinds of mistakes. The good performance for Downtown Manhattan and the tendency to guess Queens when one is unsure of the borough is clearly identifiable on these maps.

6. CONCLUSION

In conclusion, this investigation into transfer learning for image geolocation, particularly within the urban landscape of New York City, illuminates several key findings. The introduction of the RecogNYCe dataset, encompassing street-level images across diverse boroughs, provided a robust platform for evaluating pre-trained models such as ResNet18, ResNet50, and a custom CNN benchmark. Human performance, gauged through an interactive guessing game, offered a comparative perspective.

Noteworthy is the consistent success of pre-trained models, transcending human capabilities in the geolocation task. Intriguingly, the choice between ImageNet or Places365 for pre-training demonstrated minimal impact on model accuracy, indicating the adaptability of these models to the specific challenges posed by city-scale geolocation. This suggests that the unique characteristics of city environments introduce challenges distinct from planet- or country-scale geolocation.

Looking forward, the study highlights the potential for enhanced model accuracy through meticulous dataset curation. Spatial analysis of classification outcomes revealed nuanced patterns near borough borders and heightened accuracy in specific urban features, adding depth to our understanding of city-scale geolocation challenges.

In essence, this research contributes valuable insights into the dynamics of transfer learning for image geolocation, emphasizing the adaptability of deep learning models and paving the way for future advancements tailored to the intricacies of city-level geospatial tasks.

7. AUTHOR CONTRIBUTION STATEMENT

V.P. researched model architectures, implemented transfer learning models, performed model training and visualized learning mistakes. F.S. constructed the dataset, developed the game, developed the benchmark CNN model, processed data and made visualizations. Both authors contributed toward the report.

8. REFERENCES

[1] "Geoguessr," *Wikipedia*, Oct 2023.

[2] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic, "Netvlad: CNN architecture for weakly supervised place recognition," *CoRR*, vol. abs/1511.07247, 2015.

[3] Tobias Weyand, Ilya Kostrikov, and James Philbin, "Planet - photo geolocation with convolutional neural networks," *CoRR*, vol. abs/1602.05314, 2016.

[4] Sudharshan Suresh, Nathaniel Chodosh, and Montiel Abello, "Deepgeo: Photo localization with deep neural network," *CoRR*, vol. abs/1810.03077, 2018.

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[6] Lukas Haas, Michal Skreta, and Silas Alberti, "Pigeon: Predicting image geolocations," 2023.

[7] Google, "Street view api service," `https://developers.google.com/maps/documentation/javascript/streetview`, 2023.

[8] Mapillary, "Mapillary api documentation," `https://www.mapillary.com/developer/api-documentation`, 2023.

[9] Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár, "Designing network design spaces," *CoRR*, vol. abs/2003.13678, 2020.

[10] Priyal Sobti, Anand Nayyar, Niharika, and Preeti Nagrath, "Scene detection and recognition by analysing deep features using placescnn," in *Proceedings of the 2nd International Conference on Intelligent and Innovative Computing Applications*, New York, NY, USA, 2020, ICONIC '20, Association for Computing Machinery.

[11] Bavin Ondieki, "Convolutional neural networks for scene recognition," `http://cs231n.stanford.edu/reports/2015/pdfs/ondieki_final_paper.pdf`.