A Comparison Between Image Geolocalization Approaches

Andrea Manocchio
andrea.manocchio@studenti.unipd.it

Giovanni Passaro
giovanni.passaro@studenti.unipd.it

Abstract

This paper presents a comprehensive comparison of image-based geolocalization approaches. Our first approach was hierarchical, coarse to fine grid based, trained with a ResNet50 backbone on part of the YFCC100M database, cleaned up and preprocessed to ensure a more informative dataset. The second approach was starting from a different CSV-based dataset from Mapillary, metadata processing and progressing to a PlaNet-style convolutional neural network for city-level classification. This simpler method treated geolocalization as a multi-class classification problem instead of a regression or area based approach. We later also implemented a regression with the same base architecture to compare it with the results previously obtained.

1. Introduction

Image-based geolocalization represents one of the most challenging problems in computer vision, requiring systems to determine the geographic location of photographs using only visual content. The significance of this problem extends beyond academic interest, with practical applications in digital forensics, social media analysis, location-aware services, and autonomous navigation systems.

Traditional approaches to geolocalization have relied heavily on GPS metadata, manual annotation, or landmark recognition systems. However, these methods face significant limitations: GPS data is often unavailable or stripped from images, manual annotation is labor-intensive and doesn't scale, and landmark-based systems fail with generic scenes lacking distinctive architectural features.

The advent of deep learning has revolutionized computer vision, and geolocalization is no exception. Google's PlaNet system demonstrated that convolutional neural networks could learn geographic patterns directly from visual content, achieving remarkable accuracy across global locations. This breakthrough opened new possibilities for automated geolocalization systems.

The contributions of this work include: a detailed comparison of different approaches to properly tackle this challenge, documentation of practical challenges in dataset preparation and system architecture evolution, implementation of a PlaNet-inspired CNN for city-level classification, and comprehensive evaluation methodology with stratified data splitting, the experiments we have done to ensure better resluts and overcome obstacles and finally the results of our work.

Our two approaches yields different results for the difference in difficulty. The multiclass classification model produces very good results as it is a fairly simple task. On the other hand the coordinate regression still yield some good results being a much more intricate task.

2. Related Work

The field of image-based geolocalization has undergone significant evolution over the past two decades. Early approaches focused on feature-based matching and database retrieval methods.

2.1. Traditional Approaches

Hays and Efros introduced the seminal Im2GPS system, which attempted to geolocalize images by matching visual features against a large database of geotagged photographs. Their approach used global image features and nearest neighbor matching, achieving modest success on landmark-rich datasets but struggling with generic scenes. The biggest drawback in trying to replicate this architecture is the memory consumption of the knn algorithm.

Subsequent work explored various feature descriptors and matching strategies. SIFT-based approaches dominated early research, with systems attempting to match local features against large-scale databases of geotagged images. While computationally expensive, these methods showed promise for landmark recognition tasks.

Street-level geolocalization emerged as a specific subproblem, with researchers like Zamir and Shah developing systems for matching query images against Google Street View databases. These approaches achieved high accuracy in urban environments but required extensive preprocessing and were limited to areas with street-level coverage.

2.2. Deep Learning Revolution

The breakthrough came with Google's PlaNet system, which demonstrated that deep convolutional networks could learn geographic patterns directly from image content. PlaNet used an Inception architecture trained on millions of geotagged images, treating geolocalization as a classification problem over discrete geographic cells.

Following PlaNet, numerous variations emerged. Vo et al. revisited the Im2GPS problem using deep learning, showing significant improvements over traditional feature-based methods. Mall et al. introduced large-scale datasets for ur-

ban geolocalization, enabling more robust training and evaluation.

Recent work has explored attention mechanisms, multimodal fusion, and specialized architectures for specific geographic regions. However, most approaches still rely on massive datasets and computational resources, limiting practical deployment.

2.3. City-Level Classification

Our work focuses on city-level classification, a constrained but practically important variant of the general geolocalization problem. Unlike global systems that must handle worldwide geographic diversity, city-level approaches can leverage distinctive urban characteristics while maintaining reasonable computational requirements.

2.4. Combinatorial Partitioning

One of the most successful new approaches, combinatorial partitioning, follows a simple idea, starts with multiple coarse grained class sets which are overlapped to create a fine grained partitioning by a combination of the class sets. Each resulting fine-grained class is represented by a tuple constructed merging the sets.

3. Datasets

Our approach to the geolocalization problem underwent significant evolution, driven by practical considerations and performance requirements. This section documents our complete dataset journey, from the initial raw data to the postprocessed final result.

3.1. YFCC100M

Our initial dataset was composed of roughly 240 thousand images, scraped from Flickr geotagged posts. This already posed a significant issue about data quality, since those pictures had great variance, both in subjects and quality. Upon further inspection we realized that many of the images were just not informative, for example representing indoor scenery, sky/moon pictures or closeup selfies.

3.1.1 Filtering and Pre-processing

We removed every image not suitable for the task, based on conveyed information or feasibility of the identification. This method lead us to identify image scenery using a pretrained Places365(based on resNet18) model. After this step we now had a scene label for every image, and proceeded to remove all indoor pictures (manually selecting the unwanted categories) or images with a confidence score lower than 0.5, the reason being that those were probably ”bad” images. Examples of non-informative classes are ”hospital”, ”supermarket”, ”kitchen”, ”hotel room”, ”library” etc. On the other hand, the some of the selected classes were ”tundra”, ”beach”, ”plaza”, ”windmill”, ”harbor”. This process left us with around 41 thousand more qualitative images.

3.2. Mapillary

3.2.1 Multi Class Classification

Our first simpler approach with this dataset was limited to a multiclass classification problem, so we just used the folder name as a label (every city had a different folder of images) and obtained 462 thousand labeled images from 21 different cities such as London, Bangkok, Ottawa, Nairobi, SaoPaulo, Melbourne.

3.2.2 Coordinate Regression Variant

This structure included CSV files containing metadata such as image keys, GPS coordinates (both in UTM easting/northing and latitude/longitude), date of the picture. Initially, we developed a system that read CSV files to extract image metadata and geographic coordinates. The approach involved:

- **Metadata Extraction**: Reading raw.csv files containing image keys and coordinates
- **Image Matching**: Matching CSV entries with corresponding image files

However, we faced several challenges:

- **Data Inconsistency**: Mismatched or missing CSV entries and image files
- **Coordinate Complexity**: Difficulty in learning continuous coordinate regression
- **Scale Sensitivity**: Coordinate systems varying across different cities

Scale sensitivity was solved normalizing the coordinate system in (0,1) to ensure stable learning, while mismatched images were simply discarded.

The final distribution used in both approaches was:

- Training: 64% of total data
- Validation: 16% of total data
- Testing: 20% of total data

4. Methods

Our methodological approach underwent several iterations, each addressing limitations of previous versions while incorporating new insights and requirements, finally implementing a PlaNet-inspired architecture.

4.1. Model Selection

After evaluating various CNN architectures (ResNet, EfficientNet), we selected a PlaNet architecture with ResNet50 as backbone for its balance of performance, computational efficiency and provides strong spatial feature extraction through residual connections.

Transfer Learning: Leveraged ImageNet pre-trained weights, fine-tuning the network for our specific geolocalization task.

Classification Framework: Modified Classification Head transforming the problem from coordinate regression to discrete city classification, going from the 1000 original ImageNet classes to the 21 cities, simplifying the learning objective while maintaining practical utility.

4.2. Network Architecture Details

Several components are shared for both implementation (classification and regression). Our classification architecture defines a modified ResNet50 with the following components:

- **Backbone**: Uses ResNet50 pretrained on ImageNet
- **Classification Head**:
  - Global average pooling layer
  - Dropout layer (p=0.5) for regularization
  - Linear layer mapping to number of cities
  - Softmax activation for probability distribution

4.2.1 Regression Architecture Structure

The regression task needed some additional tweaks in the architecture while still keeping the same backbone

Regression Head:

- Dropout Layer (0.5): Regularization to prevent overfitting
- Linear Layer: 2048 $\rightarrow$ 512 features (ResNet50 final layer has 2048 features)
- ReLU Activation: Non-linear activation function
- Dropout Layer (0.3): Additional regularization
- Final Linear Layer: 512 $\rightarrow$ 2 outputs (latitude, longitude)

Design specifics:

- **Input Resolution**: Standard 224×224 pixels to match ImageNet pre-training
- **Normalization**: ImageNet statistics for consistency with pre-training

4.3. Training Methodology

During both our trainings we incorporated several practices to avoid overfitting and evolved through multiple iterations

Data Augmentation Strategy:

- Resize(256, 256) $\rightarrow$ RandomCrop(224): Standard ImageNet preprocessing
- Geometric: Random horizontal flips, rotations (±15°)
- Photometric: Color jittering (brightness, contrast, saturation, hue) for better lighting/weather variance resistance

Optimization Details:

- **Loss Function**: Cross-entropy loss for multi-class classification, Haversine Distance Loss for the regression task, used to compute the actual distance between the prediction and the true value
- **Optimizer**: Adam with learning rate 0.0001 for classification and 0.001 for the regression
- **Batch Size**: 32 (adjusted for GPU memory constraints)
- **Training Epochs**: 5 (Usually enough for pre-trained models, also limited by our hardware)

Regularization Techniques:

- Dropout in classification head
- Early stopping with patience, not really exploited due to the low number of epochs
- Data augmentation as implicit regularization

Training Phases:

1. **Feature Extraction**: Frozen backbone, train classification head
2. **Fine-tuning**: Unfreeze backbone, train end-to-end with lower learning rate
3. **Final Optimization**: Learning rate scheduling and early stopping

5. Experiments

Our experimental evaluation covers both traditional and deep learning approaches, providing comprehensive comparison and analysis.

Evaluation Metrics:

- Classification accuracy (primary metric)

- Per-class precision, recall, and F1-scores
- Confusion matrices for error analysis
- Training convergence curves

5.1. Experimental Setup

Hardware Configuration:

- NVIDIA 3070ti GPU with CUDA support
- 8GB RAM for data loading and preprocessing

Software Framework:

- PyTorch 1.9+ for deep learning implementation
- Scikit-learn for traditional ML methods and evaluation
- OpenCV and PIL for image processing
- Matplotlib and Seaborn for visualization

5.2. Multi Class Classification

Our multiclass classification model performs really well, also because it is a simpler task

	Accuracy	F1-Score
Training	0.982	0.951
Validation	0.973	0.946
Test	0.929	0.910

Table 1: Performance of the classification

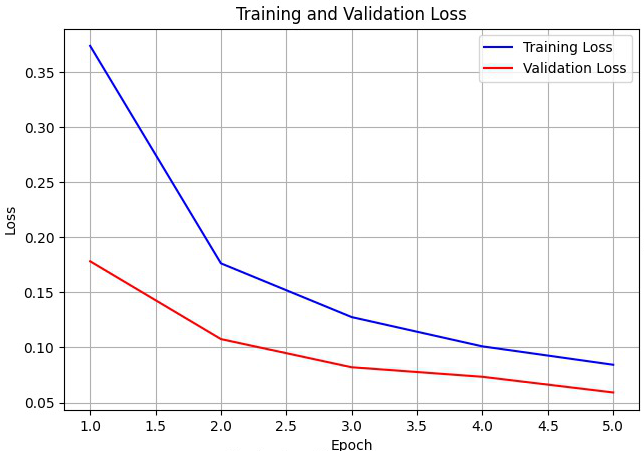
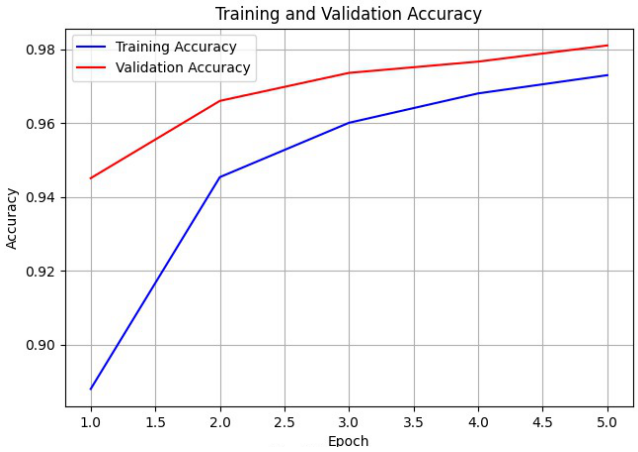


Figure 1: Loss Curve



5.3. Coordinate Regression and Grid Based Results

The first coarse/fine hierarchical grid model performed very poorly, with an average error of 4700 km. Our city reduced CNN achieved significantly better performance: **Performance Analysis**:

- **Training Accuracy**: The mean of our error distance is roughly 2054 km, however if we see the median, which is not influenced by outlier predictions, we can see that it is actually around 380 km
- **Test Performance**: The test scores show that we have a 15% of predictions were in a 100 km radius of the actual place and 66% were within 1000km

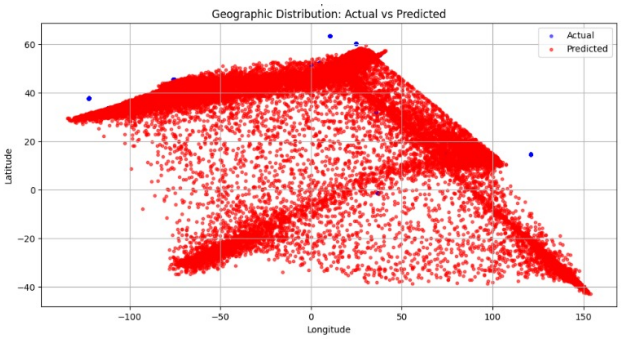


Figure 2: Prediction pattern

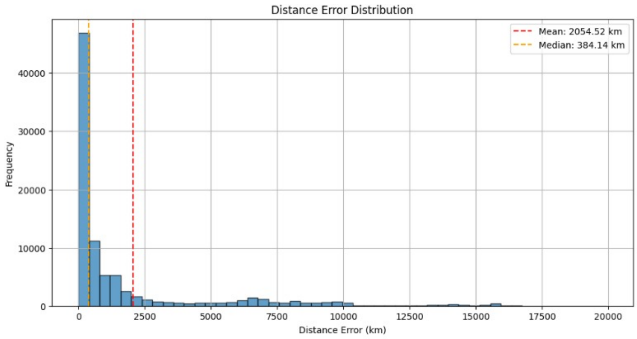


Figure 3: Error distribution

6. Conclusion

Overall this report is aimed to make a comparison between different ways to solve a geolocation problem, using three approaches:

1. **Grid Based** : Good compromise for accuracy and generalization. As explored, there are different approaches to fully exploit the approach, like combinatorial partitioning
2. **Coordinate Regression** : Theoretically the most accurate but also the most difficult to obtain good results during training.
3. **Multi Class Classification** : Performs really good, unfortunately limited by the class system, will perform poorly with every picture that was taken outside the mapped classes, raising a different problem, properly mapping the entire globe into subclasses, but doing so could make the performance plummet.

6.1. Future work

Architecture Enhancements:

- Attention mechanisms to focus on discriminative regions
- Multi-scale feature extraction for different urban elements
- Ensemble methods combining multiple architectures

Data and Training:

- Larger datasets covering more cities and countries
- Self-supervised pre-training on unlabeled geographic images
- Active learning for efficient annotation of new cities
- Longer training time and more epochs

References

1. T. Weyand, I. Kostrikov, and J. Philbin, "Planet-photo geolocation with convolutional neural networks," in *European Conference on Computer Vision*, pp. 37-55, Springer, 2016.
2. J. Hays and A. A. Efros, "im2gps: estimating geographic information from a single image," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-8, IEEE, 2008.
3. N. Vo, N. Jacobs, and J. Hays, "Revisiting im2gps in the deep learning era," in *IEEE International Conference on Computer Vision*, pp. 2621-2630, IEEE, 2017.
4. A. R. Zamir and M. Shah, "Accurate image localization based on google maps street view," in *European Conference on Computer Vision*, pp. 255-268, Springer, 2010.
5. U. Mall, G. Mattyus, E. Tryon, E. Homayounfar, L. Lakshmikanth, and R. Urtasun, "Large scale photo-realistic image dataset for geolocation," *arXiv preprint arXiv:1906.05272*, 2019.
6. D. M. Chen et al., "City-scale landmark identification on mobile devices," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 737-744, IEEE, 2011.
7. E. Kalogerakis, O. Vesselova, J. Hays, A. A. Efros, and A. Hertzmann, "Image sequence geolocation with human travel priors," in *IEEE International Conference on Computer Vision*, pp. 253-260, IEEE, 2009.
8. P. Serdyukov, V. Murdock, and R. Van Zwol, "Placing flickr photos on a map," in *ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 484-491, ACM, 2009.
9. Paul Hongsuck Seo, Tobias Weyand, Jack Sim and Bohyung Han, "CPlaNet: Enhancing Image Geolocalization by Combinatorial Partitioning of Maps"