CIS 4190/5190: Applied Machine Learning | Fall 2024

Image2GPS

Team Members: Kaily Liu, Anant Aggarwal, Siyuan Yao | *Project: Img2GPS*

1 Summary

The project achieved significant progress in predicting GPS coordinates from input images using a deep learning-based approach. A custom model built on the ResNet101 architecture was fine-tuned for the task, with modifications to the fully connected layer for predicting latitude and longitude. Dropout was incorporated to reduce overfitting, and extensive data augmentation, including random rotations, horizontal flips, and color jitter, improved generalization.

The final model employed a custom loss function combining mean squared error (MSE) and geodesic distance, optimizing for both coordinate-level accuracy and spatial error. Mixed precision training and efficient data loading significantly reduced GPU memory usage and accelerated training, enabling completion within 1 hour. A dynamic learning rate schedule (OneCycleLR) facilitated faster convergence and improved stability during training.

Key results included:

- A mean geodesic distance error of 57.38 meters, a substantial improvement over earlier iterations.
- Effective utilization of VRAM through larger batch sizes and a deeper backbone (ResNet101), balancing training efficiency and model complexity.
- Visualization of actual vs. predicted coordinates highlighted closer alignment, with reduced deviations and fewer outliers.

The combination of architectural enhancements, targeted loss functions, and efficient training pipelines was effective at creating a predictive model. Remaining challenges for the team include further reducing geodesic errors and addressing outliers which contribute to higher errors.

2 Core Components

2.1 Data Collection and Dataset

Procedure and Protocol for Data Collection:
Data was collected from the designated region on Penn's campus, following a systematic approach to ensure even spatial coverage. The images were captured from over 280 distinct locations, covering a variety of angles (straight-on and upward) and perspectives (N, NE, E, SE, S, SW, W, NW). This approach ensures that the dataset reflects the diverse viewpoints the model is likely to encounter during inference. The EXIF metadata of the images was extracted to retrieve GPS coordinates. Each image was labeled with its corresponding latitude and longitude. Care was taken to ensure GPS tags were consistently recorded.

Data Curation and Cleaning:
The raw dataset was examined for inconsistencies, such as missing GPS tags, duplicate entries, and corrupted images. Images without valid GPS metadata were discarded. Data augmentation techniques, including horizontal flips, rotations, and color jitter, were applied to increase dataset variability and improve generalization. These transformations also addressed the potential bias caused by lighting and weather variations during data collection.

Dataset Splitting:
To evaluate model performance, the dataset was divided into a training set (2303 images) and a test set (301 images), with no overlap between the two. The test set was randomly collected from within the same region

to simulate realistic evaluation conditions.

Final Dataset:

- **Training:** 2303 images with corresponding GPS coordinates.
- **Test:** 301 images with unseen spatial distribution.

2.2 Model Design

Initial Model Design:
The baseline model used ResNet18 as the backbone with modifications to the final layer to predict two outputs: latitude and longitude. While this served as a starting point, its performance was suboptimal, with a high geodesic distance error and longer training times due to limited capacity for capturing spatial features.

Iterative Improvements:

- **Upgraded Backbone:** The model was upgraded to ResNet101 for greater capacity to learn complex spatial features.
- **Fully Connected Layer:** The final layer was redesigned with dropout to reduce overfitting and improve regularization. It outputs two continuous values representing latitude and longitude.
- **Loss Function:** A custom loss function combining mean squared error (MSE) and geodesic distance was implemented to directly optimize for spatial accuracy.
- **Training Optimizations:**
  - **Mixed Precision Training:** Reduced memory usage and increased computational efficiency.
  - **OneCycleLR Scheduler:** Dynamically adjusted the learning rate to accelerate convergence.
  - **Batch Size Increase:** Leveraged available VRAM to process larger batches, improving gradient estimation and training efficiency.

Final Model Design:
The final model was a ResNet101-based architecture with a modified, fully-connected layer and a custom loss function. Training involved aggressive data augmentation, mixed precision training, and efficient learning-rate scheduling.

2.3 Evaluation and Model Performance

Evaluation Protocols:

- **Internal Validation:** The model was evaluated on the test set using the Mean Geodesic Distance Error and Mean Squared Error metrics. Validation was conducted after every epoch to monitor progress and prevent overfitting.
- **Leaderboard Submission:** The model was fine-tuned based on internal validation results and submitted for leaderboard evaluation. Submissions were selected based on their ability to minimize geodesic distance error.

Performance Metrics:

- **Mean Geodesic Distance Error:** Chosen as the primary metric for its relevance to spatial accuracy.
- **Mean Squared Error (MSE):** Used as a secondary metric during training for monitoring coordinate-level prediction accuracy.

The final model achieved a mean geodesic distance error of 57.38 meters, a significant improvement over earlier iterations. Visualization of predictions highlighted better alignment between actual and predicted GPS coordinates, with fewer outliers.

3 Exploratory Questions

3.1 Exploratory Question 1: How does batch size affect model performance and training time?

Question and Motivation:
This question investigates the relationship between batch size and both training efficiency (time) and model performance (geodesic distance error). Optimizing batch size is critical to utilizing GPU memory efficiently and minimizing training time while ensuring robust model performance. Larger batch sizes allow for better gradient estimation but can lead to generalization issues if too large.

Prior Work or Course Material:
Batch size tuning is well-documented in machine learning literature, with works indicating that smaller batch sizes may generalize better due to noisier gradient estimates, while larger batch sizes stabilize learning but may overfit. From the course material, we expected larger batch sizes to improve computational efficiency but potentially degrade generalization.

Methods for Investigation:
The experiments tested batch sizes of 32, 64, 128, and 256. Training time, memory usage, and geodesic distance error were monitored across these configurations. The same learning rate schedule and augmentation strategy were used to isolate the effect of batch size.

Results and Updated Beliefs:

- **Batch size 32:** High geodesic error (approx. 62 meters), slow training time.
- **Batch size 64:** Improved training time and performance (approx. 57.38 meters).
- **Batch size 128:** Minimal improvement in training time but slight degradation in performance (approx. 58 meters).
- **Batch size 256:** Increased geodesic error (approx. 60 meters) and minor instability in learning.

The results suggest that batch size 64 provided the best trade-off between training efficiency and generalization. Larger batch sizes suffered from slight overfitting, confirming prior expectations that too-large batch sizes could hinder generalization.

Limitations:
The analysis was limited to the GPU constraints and model architecture. Further experiments could explore dynamic batch sizing or gradient accumulation to test larger effective batch sizes without memory restrictions.

3.2 Exploratory Question 2: Does adding geodesic distance to the loss function improve GPS prediction accuracy?

Question and Motivation:
The goal of this exploration was to assess whether incorporating geodesic distance into the loss function improves GPS prediction accuracy compared to using MSE alone. Geodesic distance directly measures spatial error, which is more relevant to the task than MSE.

Prior Work or Course Material:
Existing research highlights the importance of task-specific loss functions. For example, in tasks like object

detection or pose estimation, domain-specific losses outperform generic ones. From the course, we anticipated that adding geodesic distance would improve predictions aligned with the evaluation metric.

Methods for Investigation:
Two models were trained: one using only MSE loss and another using a combined loss function (Loss = MSE + $\alpha \times$ Geodesic Distance, where $\alpha = 0.1$). Both models used the same architecture (ResNet101) and data augmentation pipeline to ensure comparability.

Results and Updated Beliefs:

- **MSE Only:** Geodesic distance error of 62 meters.
- **MSE + Geodesic Distance:** Reduced geodesic error to 57.38 meters.

The combined loss function significantly reduced spatial error, confirming that incorporating geodesic distance aligns training objectives with the evaluation metric. However, tuning the weight α further might yield better results.

Limitations:
The fixed value of $\alpha = 0.1$ may not be optimal. A more thorough hyperparameter search could refine this. Additionally, computing geodesic distance for every batch slightly increased training time.

3.3 Exploratory Question 3: How do deeper models (ResNet18 vs. ResNet101) affect GPS prediction performance?

Question and Motivation:
This question investigates whether a deeper backbone (ResNet101) improves GPS prediction performance compared to a shallower backbone (ResNet18). Deeper models capture more complex features, but their computational cost is higher.

Prior Work or Course Material:
From the course, we learned that deeper networks generally outperform shallower ones in complex tasks due to their greater capacity for feature extraction. However, research warns about diminishing returns in performance as depth increases.

Methods for Investigation:
Two models were trained:

- **ResNet18:** The baseline model with standard training settings.
- **ResNet101:** A deeper model with identical hyperparameters.

Both models were evaluated on the test set using the same loss function and performance metrics (geodesic distance and MSE).

Results and Updated Beliefs:

- **ResNet18:** Mean geodesic distance error of 64 meters.
- **ResNet101:** Reduced geodesic distance error to 57.38 meters.

The deeper model showed significant improvement in capturing spatial features, particularly in regions with high variability in image perspectives. This reinforced our belief that deeper architectures are beneficial for tasks requiring high spatial precision.

Limitations:
While ResNet101 improved performance, the trade-off in training time and memory usage was non-negligible.

Testing more recent architectures like EfficientNet might yield similar results with lower computational costs.

3.4 Exploratory Question 4: How does freezing layers in ResNet affect GPS prediction performance?

Question and Motivation:
This question investigates the impact of freezing layers in ResNet on GPS prediction performance. Freezing early layers can reduce training time and prevent overfitting by leveraging pre-trained features, but it might limit the model's ability to adapt to the specific task.

Prior Work or Course Material:
We learned in the course that transfer learning with frozen layers is often effective for tasks with limited data. Research suggests that while freezing initial layers preserves generic features, fine-tuning all layers might improve performance for tasks requiring domain-specific representations.

Methods for Investigation:
We compared three training strategies using ResNet18:

- **Final Layer Frozen:** All convolutional layers except the final layer were frozen during training.
- **Two Frozen Layers:** Freezing the final layer, as well as layer 4.
- **Three Frozen Layers:** Freezing layer 3, layer 4, and the final layer.
- **Fully Trainable:** All layers of the model were fine-tuned during training.

Models were evaluated on the test set using the same loss function and performance metrics (geodesic distance and MSE).

Results and Updated Beliefs:

- **Final Layer Frozen:** Validation RMSE: Validation RMSE: 91.28
- **Two Frozen Layers:** Validation RMSE: 84.14
- **Three Frozen Layers:** Validation RMSE: 77.09
- **Fully Trainable:** Validation RMSE: 102.43

The results indicate that partially unfreezing layers improves performance, as the model can adapt to the GPS prediction task. However, fully unfreezing all layers caused overfitting and degraded performance, confirming that pre-trained features in the early layers remain valuable for general feature extraction.

Limitations:
Fully unfreezing all layers not only performed worse but also significantly increased training time and computational cost. Future investigations could explore hybrid approaches, such as selectively freezing or dynamically unfreezing layers during training, to optimize both performance and efficiency.

4 Team Contributions

Each member contributed equally to data collection, model design, exploratory direction, and the project report.

Dataset on Hugging Face here. Model uploaded to Hugging Face here.