CIS 4190/5190: Applied Machine Learning Fall 2025

Campus-Scale Img2GPS:From ResNet Baselines to Swin-B with Controlled Fine-Tuning

Team Members: Xuzhou Wang, Jackie Chen, Tiancheng Pu *Project: [Img2GPS]*

1 Introduction

This project studies the Img2GPS task: predicting the geographic location (latitude and longitude) of a photo using only the image content. We focused on building a complete pipeline that is realistic for a small-team setting, including data collection, label validation, and model iteration under limited data. Our goal was to produce a model that performs well not only on our internal split, but also under the course setting where the hidden test set may include distribution shifts such as different lighting or weather.

We collected a campus-scale dataset of 583 images using a fixed-location protocol. At each location, we captured eight viewing angles and intentionally included multiple environments—day and night, sunny, rainy, and cloudy—to introduce natural variation. GPS labels were extracted from EXIF metadata, converted from DMS to decimal degrees, validated using coordinate range checks, and compiled into a reproducible `metadata.csv`. Before model training, we ran exploratory analysis to verify data integrity and coverage. This included visualizing the train vs. validation coordinate distributions, mapping points on an interactive map to spot outliers, and measuring image brightness distributions to confirm meaningful day/night variation.

On the modeling side, we started with a ResNet-18 baseline and then improved performance through transfer learning and controlled fine-tuning. A frozen ResNet-50 backbone with a larger regression head reduced the average distance error from 99.88 m to 79.77 m. Allowing partial adaptation by unfreezing only the final ResNet stage (layer4) produced a small additional gain to 78.10 m. Our best model replaced CNNs with a Swin-B hierarchical Vision Transformer[1] and used a residual regression head while fine-tuning only the last Swin stage. This final design achieved our best internal result of 70.25 m average distance error. Overall, our results suggest that (i) strong pretrained backbones matter significantly in low-data GPS regression, and (ii) selective fine-tuning plus a transformer backbone can better capture global scene structure that is useful for localization, at the cost of slower inference.

2 Core Components

2.1 Data Collection and Dataset

2.1.1 Collection Protocol

We constructed a Engineering Campus-scale Img2GPS dataset consisting of 583 images. Data were collected using a fixed-location protocol to ensure that each coordinate corresponds to a physically meaningful point on campus. For each location, we captured eight distinct viewing directions to reflect viewpoint variability, since images taken at the same geographic point can exhibit substantial visual differences depending on camera orientation. In addition, we deliberately sampled across diverse environmental conditions, including daytime and nighttime scenes as well as sunny, rainy, and cloudy weather. This design choice was intended to increase intra-location diversity.

2.1.2 GPS Labeling from EXIF Metadata

Ground-truth latitude and longitude labels were obtained from the EXIF GPS metadata embedded in each image. Because GPS coordinates are commonly stored in degrees–minutes–seconds (DMS) format, we converted the EXIF latitude and longitude values from DMS into decimal degrees before training. After conversion, we performed validation checks to ensure coordinates fall within physically valid ranges ($-90 \leq$ latitude ≤ 90 and $-180 \leq$ longitude ≤ 180). We then generated a reproducible `metadata.csv` file

with the schema `file_name`, `latitude`, and `longitude`, which served as the single source of truth for label lookup during training and evaluation.

2.1.3 Data Curation, Integrity Checks, and Dataset Split

To ensure label reliability and prevent silent failures during training, we curated the dataset with several integrity constraints. First, we enforced a strict naming convention and retained only image files matching `IMG_*.jpeg`. Second, we removed corrupted or unreadable images and eliminated metadata entries that did not correspond to an existing file. Third, we enforced a one-to-one correspondence between image files and metadata rows, which is essential for regression tasks where even a small number of mismatched labels can severely degrade learning.

For internal evaluation, we performed a random dataset split into 80% training and 20% validation. We organized each split into a structured directory layout (`train/validation`), where each directory contains the images and a corresponding `metadata.csv`. We did not maintain a local test split; instead, we used the course leaderboard's hidden test set for final evaluation. Finally, we hosted the dataset on Hugging Face to support dataset versioning and to enable consistent access Colab across development environment.

2.1.4 Preprocessing

All images were standardized to a consistent input representation before being fed into the models. Specifically, images were resized to 224×224, converted to tensors, and normalized using ImageNet mean and standard deviation[2]. The preprocessing pipeline used in the final submission is deterministic (i.e., it does not include stochastic data augmentation), which helps ensure that the model receives a consistent input distribution during inference that matches the evaluation pipeline.

2.2 Model Design

2.2.1 Learning Objective and Target Normalization

We formulate Img2GPS as supervised regression from an image x to a two-dimensional target vector $y = (y_{\text{lat}}, y_{\text{lon}})$. Since our data are geographically localized (coordinates occupy a narrow numeric range), we normalized the regression targets to improve numerical stability and optimization dynamics. Let μ and σ denote the mean and standard deviation computed from the training set for each coordinate dimension. We trained the model to predict the normalized targets

$$\tilde{y} = \frac{y - \mu}{\sigma},$$

and converted predictions back to decimal degrees at inference time via

$$\hat{y} = \hat{\tilde{y}}\sigma + \mu.$$

This normalization makes the regression problem better conditioned and reduces sensitivity to small gradient magnitudes caused by the tight coordinate scale.

2.2.2 Design Rationale and Iterative Development

Our model development was guided by two main considerations: (i) maximizing performance under limited labeled data through transfer learning, and (ii) improving robustness by allowing controlled domain adaptation while avoiding overfitting. We therefore explored a sequence of models that progressively increased capacity and task alignment.

2.2.3 Baseline: ResNet-18

We began with a ResNet-18 baseline model[3] trained to regress normalized latitude and longitude, which has a average distance of 99.88 meters. This served as a reference point for evaluating subsequent improvements and confirmed that the dataset pipeline (label extraction, cleaning, and preprocessing) produced a learnable signal.

2.2.4 Model 1: ResNet-50 + Large MLP Head (Frozen Backbone)

The first major improvement replaced the baseline backbone with a pretrained ResNet-50[2][3]. We froze all convolutional layers and treated the backbone as a fixed feature extractor, producing a 2048-dimensional global representation. We then trained a comparatively expressive regression head (a deep MLP with GELU activations, LayerNorm, and dropout) to map these features to GPS coordinates. This configuration isolates learning to the head and provides a strong transfer-learning baseline that leverages general-purpose ImageNet features while limiting the risk of overfitting the backbone on a small dataset.

2.2.5 Model 2: ResNet-50 + Smaller Head + Partial Fine-Tuning (Layer4 Only)

To introduce domain adaptation while preserving training stability, we unfroze only the final ResNet stage (layer4) and kept earlier layers frozen. The intuition is that lower-level filters capture generic visual primitives (edges, corners, textures), whereas deeper layers are more task-specific and can benefit from limited fine-tuning. In parallel, we reduced the head capacity (512→128→2) to reduce overfitting and to encourage the backbone to contribute more to the final prediction. This selective fine-tuning strategy provided a modest but consistent improvement over the fully frozen ResNet-50 configuration.

2.2.6 Model 3 (Best): Swin-B + Residual Regression Head + Partial Fine-Tuning

Our best-performing model transitions from convolutional backbones to a hierarchical Vision Transformer, Swin-B[4]. Swin employs window-based self-attention with hierarchical feature maps, which allows it to capture both local details and longer-range structural cues in the scene. At the attention level, the core computation is scaled dot-product attention:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d}}\right)V,$$

where Q, K, and V are learned query, key, and value projections and d is the feature dimension[5]. Swin constrains attention to windows and shifts the window partition between layers, enabling efficient incorporation of broader spatial context.

For fine-tuning, we froze the full Swin-B backbone and unfroze only the final Swin stage (the last feature block), following the controlled adaptation strategy used in Model 2. We paired this with a residual regression head that decomposes prediction into an initial linear term plus a nonlinear correction:

$$\hat{y} = W_{\text{base}}x + f_\theta(x),$$

where x is the backbone feature vector, $W_{\text{base}}x$ provides a stable first-order estimate, and $f_\theta(x)$ adjusts the output via a lightweight MLP. This design will improve stability and accuracy while limiting the number of trainable parameters.

2.3 Evaluation and Model Performance

2.3.1 Evaluation Metric

We evaluated localization quality using average geographic error measured in meters. For each image, we computed the great-circle distance between the predicted coordinate $(\hat{\phi}, \hat{\lambda})$ and the ground-truth coordinate (ϕ, λ) using the Haversine formula:

$$d = 2R \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi)\cos(\hat{\phi})\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right),$$

where ϕ denotes latitude in radians, λ denotes longitude in radians, $\Delta\phi = \hat{\phi} - \phi$, $\Delta\lambda = \hat{\lambda} - \lambda$, and $R \approx 6{,}371{,}000$ meters is the Earth radius. We report the mean of d across the evaluation set as the primary performance metric. This metric is consistent with the leaderboard evaluation, which also reports mean geographic error in meters.

2.3.2 Internal Protocol and Model Selection

We used the 80/20 train/validation split for internal evaluation. Each model was trained on the training set. We used the validation set to compare model variants (e.g., changing the backbone, the regression head, or which layers to fine-tune) and to decide which configuration to keep. For the final test evaluation, we submitted the selected model to the course leaderboard, which computes performance on a hidden test set that is not accessible during development. Throughout this process, we tried to keep comparisons fair by using the same preprocessing and the same evaluation metric for all models.

Hyperparameters and the number of training epochs for each stage were determined empirically based on validation performance. In particular, for each stage we selected the checkpoint at the epoch with the lowest validation loss to reduce overfitting.

When deciding which model to submit, our primary criterion was the lowest average distance error in meters on held-out data. We also checked whether the improvement was consistent (not caused by unstable training or one unusually good run). In addition, we recorded inference time because the most accurate model can be much slower, which matters if the system is used in a practical setting.

2.3.3 Results

Table 1 summarizes our main internal results. Switching from the ResNet-18 baseline to pretrained backbones led to a large improvement, showing that transfer learning is very helpful when the dataset is small. Freezing a ResNet-50 backbone and training a stronger regression head reduced the error to 79.77 m, and fine-tuning only the last ResNet stage (layer4) provided a small additional improvement to 78.10 m.

Our best result came from Model 3, which uses a Swin-B backbone and fine-tunes only its last stage. This model achieved 70.25 m average distance error, suggesting that the transformer backbone captures broader scene structure that is useful for localization. However, it also had the highest inference time in our environment, so it represents a clear accuracy–speed tradeoff.

Model	Avg. Distance (m)	Avg. Inference Time / Image
Baseline (ResNet-18)	99.88	–
Model 1: ResNet-50 (frozen) + large head	79.77	7.61
Model 2: ResNet-50 (layer4 fine-tuned) + small head	78.10	25.75
Model 3: Swin-B (last stage fine-tuned) + residual head	70.25	96.60

Table 1: Evaluation results on the dataset split. Inference times are measured in our environment and mainly reflect relative speed differences across models.

3 Exploratory Components

3.1 Exploratory Data Analysis for Data Quality and Robustness

Before spending time on larger models, we used exploratory analysis to verify that our dataset and labels were trustworthy, and to understand what kinds of variation the model would need to handle. Since our labels come from EXIF metadata, the main risk is not noise labels in the usual sense, but silent failures such as incorrect parsing, missing files, or a small number of coordinates that are far outside the campus region.

3.1.1 Geographic Coverage and Split Consistency

We first examined the spatial distribution of coordinates for the training and validation splits. We plotted longitude vs. latitude for both splits on the same figure to check whether the random split accidentally created a geographic imbalance. In addition, we visualized the same points using an interactive Folium map(see Appendix A.1). The map view was especially useful because it makes outliers obvious: a single malformed coordinate would appear far away from the main cluster. Overall, the points formed a compact

campus-scale region, and the training and validation splits occupied similar areas, which increased our confidence that the split is reasonable and that the EXIF-to-decimal conversion works well.

3.1.2 Illumination Diversity via Brightness Statistics

Because we collected images across daytime/nighttime and different weather conditions, we wanted a simple quantitative check that this variation is actually present in the dataset. We therefore computed the average grayscale intensity for each image and plotted a histogram of brightness values(see Appendix A.2). This analysis showed that the dataset covers a broad range of illumination levels rather than concentrating around a narrow band. Practically, this supported our decision to focus on model designs that can handle changes in global appearance (e.g., lighting and contrast), instead of assuming that the test images will match a single condition.

3.2 Implementation Choices to Support Efficient Exploration

Our dataset is hosted on Hugging Face, where each row contains an image and its metadata. During EDA, repeatedly decoding images can slow iteration substantially, especially when we only need coordinate columns for plotting geographic coverage. To make exploration faster, we constructed Pandas DataFrames using only the latitude and longitude columns, which avoids triggering image decoding. This allowed us to regenerate scatter plots and maps quickly when validating splits or checking for suspicious coordinates.

For the brightness analysis, image decoding is necessary. We implemented a lightweight loop that converts each image to grayscale and records the mean pixel intensity. Although this does not directly improve model accuracy, it improved development speed and made it easier to run sanity checks multiple times as we updated the dataset and regenerated splits.

3.3 Insights Connected to Model Iteration

The exploratory results influenced how we approached modeling. First, because the dataset covers multiple illumination conditions, we prioritized transfer learning and architectures that can use higher-level scene structure rather than relying on unstable low-level color cues. Second, we tracked inference time alongside accuracy during experimentation. The Swin-B model produced the best average distance error, but it was also noticeably slower than ResNet-based models in our environment. Reporting both accuracy and runtime helped us frame the final result as an accuracy–speed tradeoff, which is relevant if this system is deployed beyond the course setting.

4 Team Contributions

All team members participated in the full project pipeline, including data collection, EXIF metadata extraction and labeling, data cleaning and splitting, exploratory analysis, model development and training, and internal evaluation. The main difference was the emphasis of each member's responsibilities.

Xuzhou Wang primarily led the data collection process and ensured the dataset covered multiple viewpoints and environmental conditions; he also assisted with model implementation and training runs.

Jackie Chen focused mainly on the modeling work, including designing model variants, setting up transfer-learning and partial fine-tuning strategies, running experiments, and comparing model performance across iterations.

Tiancheng Pu primarily conducted the EDA for data validation and robustness checks, consolidated experimental results, and focused on editing and writing parts; he also assisted with modeling and training workflow.

5 Suggestions for future iterations of this project

With additional time, we would expand the dataset in a more structured way. In particular, we would collect more "hard" samples that are likely to appear under distribution shift, such as strong backlight, motion blur, rainy night scenes, and extreme shadows, while also ensuring better coverage across campus regions. It would also be helpful to perform location-aware splitting (e.g., holding out certain locations entirely) to measure true generalization to unseen places rather than only unseen images.

On the modeling side, we would explore lightweight data augmentation that targets realistic variations (small color jitter, random blur, and mild perspective changes) and evaluate whether it improves robustness without changing the true location signal. We would also test multi-task or auxiliary objectives, such as predicting discrete location clusters or camera orientation in addition to GPS regression, which may encourage the backbone to learn more location-relevant features.

Finally, we would further study the accuracy–efficiency tradeoff. Although Swin-B achieved the best accuracy in our experiments, it was significantly slower than the CNN models. A practical next step would be to evaluate smaller transformer variants (e.g., Swin-T) or efficient CNN-transformer hybrids, and to consider model compression methods such as pruning or quantization to reduce inference time while maintaining acceptable localization error.

6 References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*, 2021.

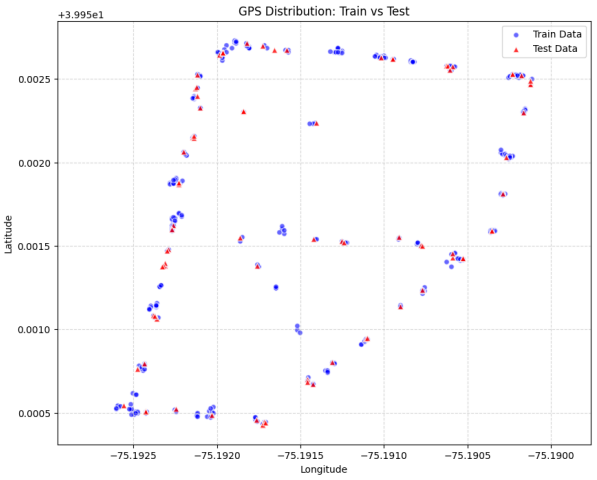
[2] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of CVPR*, 2016.

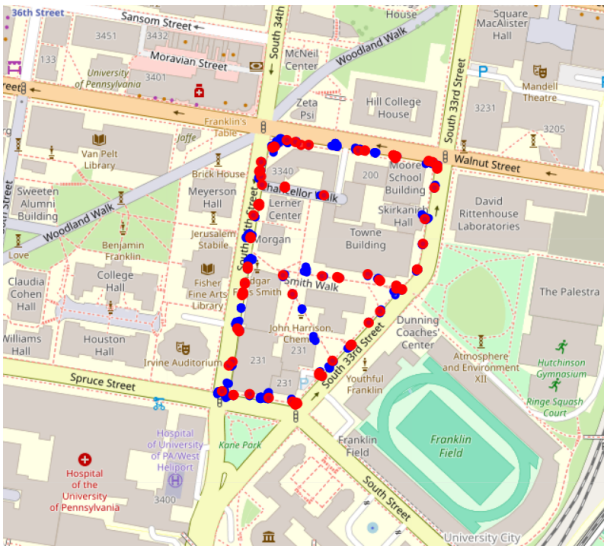
[4] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In *Proceedings of ICCV*, 2021.

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NeurIPS*, 2017.

A.1 Geographic Distribution Visualizations



(a) Longitude vs. latitude scatter plot for training and validation splits.



(b) Interactive Folium map visualization (screenshot) of the same GPS points.

Figure 1: Visual checks for geographic coverage and split consistency (Section 3.1.1).

A.2 Illumination Diversity Visualization

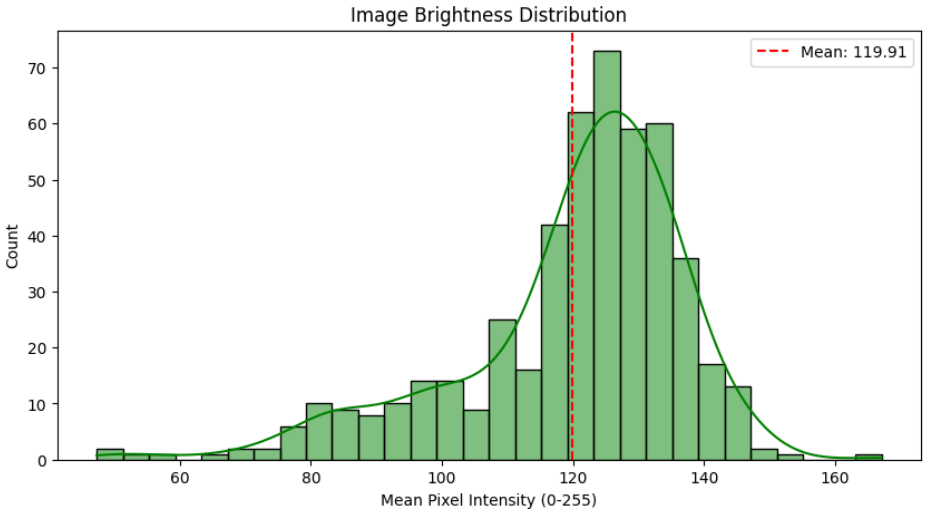


Figure 2: Histogram of mean grayscale intensity across training images (Section 3.1.2).