CIS 4190/5190: Applied Machine Learning — Fall 2024

An Analysis Ensembling Models in Vision

Team Members: Ian Lamont, John Majernik — *Project: Img2GPS*

final model repository, final dataset

1 Summary

The goal of this project was to build a computer vision model that could predict GPS locations from images taken from the engineering quad. To approach this problem, we developed and tested multiple ensemble strategies that combined predictions from distinct models. In doing so, we explored the question: how does ensembling improve GPS prediction accuracy compared to standalone models? By training diverse models, we observed which model architectures contributed most to the best-performing ensemble. These insights helped us determine which model designs were better suited for the task of predicting the GPS location of an image and what problems are suited to ensembling.

Our ensemble model was designed to balance the strengths of each base model. Below is a summary of the models and their design:

1. **Custom ResNet with Attention and FPN:** Combines a ResNet50 backbone with channel and spatial attention modules and a Feature Pyramid Network (FPN), enhancing multi-scale feature extraction and focusing on relevant spatial regions of the image.
2. **ResNet18 with Custom Loss and Oversampling:** A lightweight ResNet18 model trained with a custom geodesic loss function and oversampled edge cases to improve predictions in less populated regions of the dataset.
3. **Custom CNN:** A simple, custom CNN architecture with fewer layers, optimized for fast and generalizable predictions.
4. **Region-Based Ensemble:** A two-stage architecture where a ResNet18-based ensemble classifier first assigns images to one of seven predefined regions. Region-specific ResNet18 models are then used to predict GPS coordinates for each region.

Our ensemble methods included simple averaging, weighted averaging, and a region-based ensemble approach. The best-performing model was the Region-Based ensemble achieved a mean geodesic distance of 45 meters and a median distance of 33 meters from the target location (figure 1). A plot of model accuracy can be seen in Figure 1 of the appendix.

2 Core Components

2.1 Data Collection

The initial data collection phase involved multiple trips to the Engineering Quad on different days to help minimize the effect of differing conditions. Images were collected according to the project description: by spinning in a circle, feet planted, to get multiple images from the same location. We collected about 1600 images using this method. Early in the project, we attempted to increase this base dataset by dividing each image into 9 different images, but this decreased performance, so we stuck with the base images.

For certain base models in the ensemble, images were given the additional information of a "region" by dividing the quad into 7 areas based on GPS coordinates using the `shapely` library (figure 8). All datasets were split into train and test sets using a 75/25 split. Datasets were then cleaned to make images 224x224, normalized, and augmented with color jitter and flipping to improve model robustness.

2.2 Design Considerations and Training Methods for Ensemble Models

2.2.1 Model 1: Custom ResNet50 with Attention and FPN

The first model combines the feature extraction capabilities of a ResNet50 backbone with channel and spatial attention mechanisms, as well as a Feature Pyramid Network (FPN), to improve GPS prediction. The goal was to ensure that the model could effectively use both global and local features, as geographic markers often rely on fine-grained details and broader contextual clues.

Initial iterations with a baseline ResNet50 architecture showed that the model struggled with edge cases, such as subtle or obscured geolocation markers. To address this, channel and spatial attention mechanisms were introduced to dynamically focus on relevant features. A custom loss function penalizing distant guesses, grid search for hyperparameters (dropout rate, learning rate), and other adjustments led to significant improvements.

Performance (figure 3, 4):

- Initial MAE of baseline: 0.000540
- Improved Architecture MAE: 0.000458

The architecture encouraged diverse predictions while improving the mean absolute error (MAE).

2.2.2 Model 2: Fine-tuned ResNet18 with Custom Loss and Oversampling

This model fine-tuned a ResNet18 backbone for GPS coordinate prediction. A custom fully connected layer outputs longitude and latitude, and the model uses a geodesic loss function that exponentially penalizes distant guesses.

Observing that predictions were overly concentrated near the center due to data imbalance, oversampling was applied to underrepresented edge cases (figure 5). Noise and augmentation were added to these samples to improve learning. The geodesic loss function ensured large errors were heavily penalized, focusing the model on precise localization.

Performance (figure 7, 6):

- Optimizations resulted in a ˜7% decrease in MAE.

2.2.3 Model 3: Custom CNN

This model was the first model created as a baseline to start studying the problem. What was initially a custom ResNet for studying the question of "how do residual layers contribute to accuracy" continually got stripped down to a simple CNN of a couple of layers and a new research question. The original model was continually overfitting so the drop in complexity was seen as the solution. As the scope of the project change this was kept and added to the final ensemble.

The model was trained with a mean squared error (MSE) loss function to minimize the difference between predicted and true GPS coordinates.

2.2.4 Model 4: Region Classification into Location Ensemble

This model is composed of 2 different ensembled layers. The first layer classified each image into one of 7 regions (figure 8). Images are then grouped by region to be predicted according to a region specific model. All models in this architecture are based off of ResNet18 because of it's robustness to overfitting and simplicity compared to other models. Other models like ResNet50 and VGG were tested but they did not perform as well and are not included in the analysis.

The base region classifier is made of 5 models ensembled, each with the same ResNet18 architecture and trained with MSE loss (figure 9, 12, 10, 11). Each one has slightly different training parameters. Learning

rate, stopping epoch, and scheduler rate were changed using grid search to find the best performing models. Grid search did help tune the parameters but they are not super exact due to limitations in computing resources and time. A more thorough search would likely improve performance.

Clearly the different models have differing performances per region (e.g. model 0 outperforms model 2 in the "outer quad" region but underperforms in the "top of walk" region (figure 12). This is consistent with the initial hypothesis that we could gleam the best performance per region by creating an ensemble. This is supported by the ensemble outperforming almost every base model.

It is important to note that mode 1 does outperform the ensemble slightly in these graphs. This shows how ensembles are not strictly better than non-ensembled models. Issues like this were common when the base models performed worse and became less of an issue as the base models improved. This is likely due to worse models "pulling down" the performance of the others. It should also be said that these specific results are subject to change when the validation set is changed, so all performance should be taken as exact.

The models were assembled by taking a weighted average of each model output. Each model outputs a 7 dimensional vector representing each region which is then multiplied by the model's accuracy in that region (weighted by frequency). All vectors are then averaged. This is a naive ensembling method. An avenue to keep exploring ensembling's effectiveness would be to use a regression model to determine the weights of each model.

After a region is determined for an image it is passed to 1 of 7 models corresponding to the regions. These models were trained only on data that lies within its region and were stopped early to avoid overfitting. They used cross entropy loss to train. These models directly output predicted coordinates. These models perform extremely well when given the correct region for an image with the average coordinate being 32.9m off (figure 13, 14). Many of these base models are only slightly better than a naive mean of the region coords. The performance of the whole model could be improved by fine tuning these base models to have a more complex understanding of their region beyond an average guess. Additionally, spitting the orange and green regions corresponding to the corners of 33rd and walnut and 34th and spruce respectively into smaller regions to avoid the right angle corner which causes the mean to be far from all values could improve the accuracy. These are the 2 most populous regions and both have many points classified 50m from where they should be.

Overall the performance of the entire pipeline was better than expected. The mean distance from a projected coordinate to an actual coordinate was 45m and the median was 33m (figure 17, 18). This result is to be expected from the underlying architecture. An analysis of the distribution of distance errors shows that a majority of the images are within 50m of their actual position (figure 1). Under perfect region classification almost all images are classified within 50m of their intended position. The increase of images being way beyond their actual position could be caused by an imperfect region classification. Images that are classified in their wrong region are almost guaranteed to perform extremely poorly which could be exemplified by the long tail on the distribution.

This model was evaluated at each level using held out validation data. It was chosen to split the data collected 75-25 as to have a variety of validation data. Unfortunately the model underperforms on the small "released_img" database provided. We attribute some of this error to the fact that images from our collected dataset are taken at eye level aligned with the horizon which is not the case in this dataset. More data collection is necessary to help the model understand these cases. We experimented with artificial data creation early in the project by splitting each image into 9 different ones. We decided to abandon this idea due to poor performance attributed to no identifiable locations in the small images, but maybe this would help with test performance.

3 Exploratory Questions

For each of our exploratory questions, we aimed to thoroughly investigate the impact of ensembling on GPS prediction accuracy and evaluate the architectural components that contribute most to ensemble perfor-

mance.

3.1 Question and Motivation

The primary questions explored in this project were:

- What areas of a model architecture can benefit from ensembling?
- Is ensembling always beneficial?

Our motivation stemmed from the diversity of images collected for this project. We hypothesized that different models may specialize in predicting specific regions (e.g., engineering walk or walnut street) more effectively. By identifying the strengths of individual models and strategically combining them, we believed ensembling could enhance overall prediction accuracy.

3.2 Related Coursework

The course material on ensemble methods directly informed our approach. Concepts such as variance reduction and robustness through ensembling were critical to our methodology. Fixed-weight averaging and region-based ensembling were two techniques explored in this project. As discussed in class, combining multiple models reduces variance, and focusing on specific data regions improves robustness. This project demonstrated the practical application of these theoretical concepts to a real-world prediction problem.

3.3 Initial Ensemble

To begin, we created a simple ensemble using our first three base models. Two approaches were implemented:

1. Averaging predictions from the base models.
2. Weighted averaging based on individual model performance.

As shown in Figures 15 and 16, while the weighted average ensemble reduced variance compared to any single base model, it performed slightly worse than the best-performing model in terms of accuracy. This highlights a key trade-off: ensembling reduces prediction variance but does not always guarantee improved accuracy, especially when weaker models influence the final prediction.

3.4 Region-Based Ensemble

To address the shortcomings of the initial ensemble, we implemented a region-based ensembling architecture. This approach involved two stages:

1. A region classifier based on ensembled ResNet18 models assigns each image to one of 7 regions.
2. An ensemble of region-specific models predict the GPS coordinates for images within their respective regions.

3.5 Results and Updated Beliefs

Through the creation of a 3 model ensemble and a pipelined, ensembled model we became more aware of the parts of model architecture that benefit from ensembling. The 3 model ensemble surprisingly preformed slightly worse than the base models. This could have had many factors. One potential cause is non-independent models since they all predict around the mean coordinate. Another potential factor is the naive averaging ensembling which could be poorly suited for this problem. However even though this model did not preform exceptionally, variance decreased by a non-insignificant amount. This signals that we could have been on the right track and adding more models could increase performance (by shifting the bias of the system). This model help us understand what factors contribute to the ensemble performance like

independence and ensembling method. We also disproved our hypothesis that ensembling is always better. The pipelined region model performed extremely well. We can contribute some of this performance to the region classifier which had a high accuracy ensemble backing it. This ensemble benefited extremely from the independence of the different models; each one was strong at identifying different regions. The second layer of the pipeline also benefited by allowing each model to "solve its own problem" and aggregate their solutions. This model helped us learn that when a problem can be divided into smaller subproblems, and ensemble of models to solve the subproblems may be helpful. Additionally, we can answer our second question "is assembling always beneficial?" with "yes, if the question is well suited for it."

3.6 Limitations

Despite the relative success of our models, our project did face minor limitations.

- Certain parts of the model lacked fine hyperparameter turning that could have improved performance significantly.
- The regions were chosen naively for what a human would classify as different areas, however this may not be the best model for a machine.
- While training there were instances of float errors which is significant as latitude and longitude were stored with relatively low precision for highly specific locations
- Naive average assembling could be significantly improved with a learning model to determine the output.
- Short grid search times could have chosen bad hyperparameters
- Relatively small dataset size (even with 1600)
- Issues with extracting exact GPS metadata from images (and the actual metadata itself)

While these issues probably led to worse performance, they are not extremely significant. Many of them could be improved in the future to continue exploring this problem. Also these issues did not impact our understanding of our exploratory questions.

4 Team Contributions

John contributed by developing two of the base models and implementing the initial ensembling methods. He also collaborated on data collection and writing the report. Ian designed and implemented the region-based ensembling architecture and developing the custom CNN model. He also worked on data collection and contributed to writing the report.

A Appendix: Additional Figures

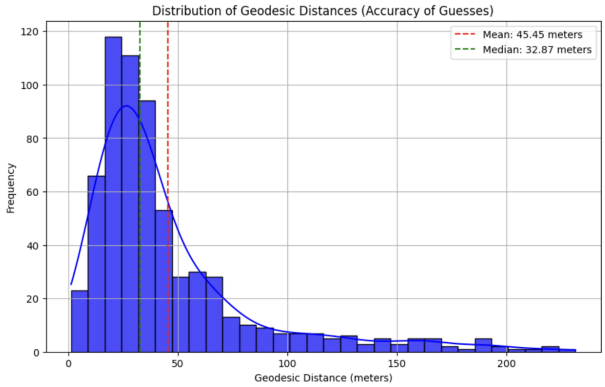

Figure 1: Final Distribution of Prediction Accuracy

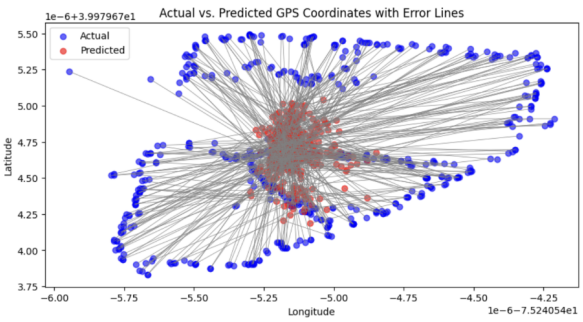

Figure 2: Plot of baseline predictions

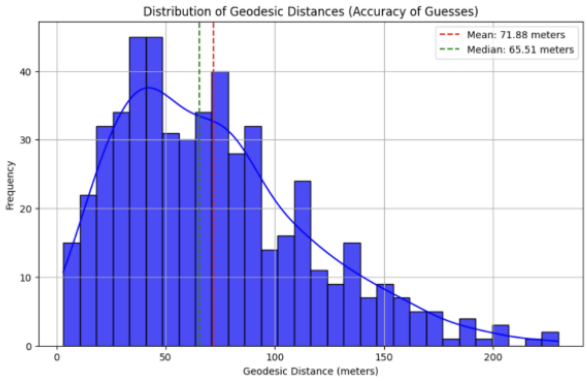

Figure 3: Final accuracy distribution of tuned model 1

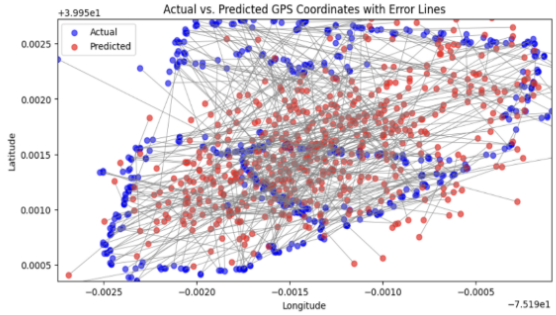

Figure 4: Plot of guesses of final model 1

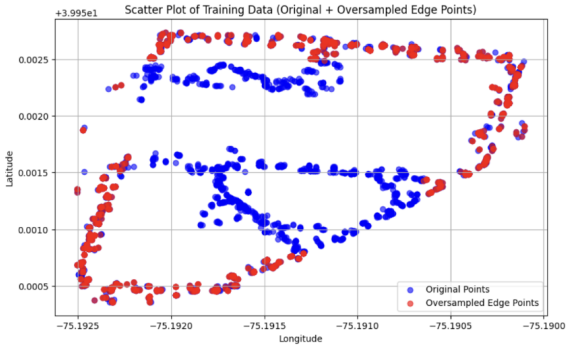

Figure 5: Oversampled edge case points

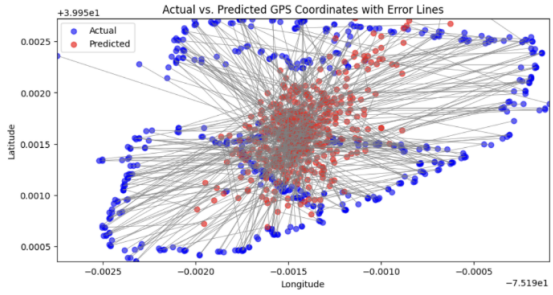

Figure 6: Predictions of model 2

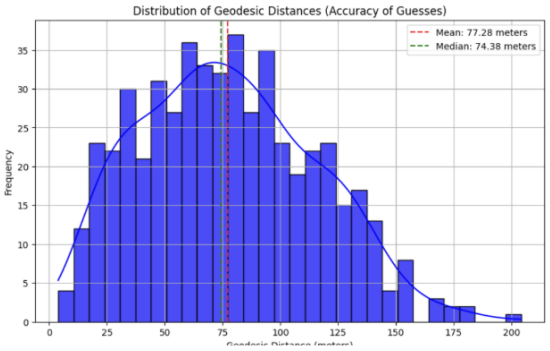

Figure 7: Final accuracy distribution of model 2

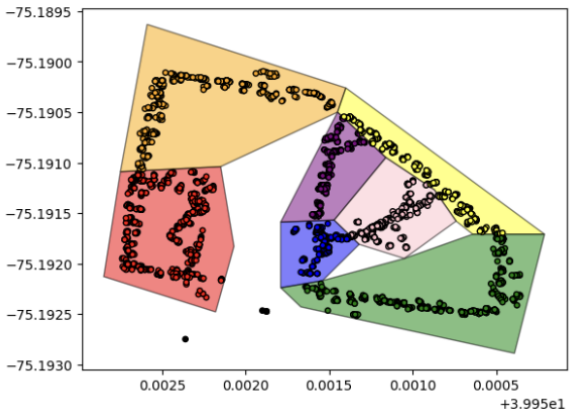

Figure 8: Model 4 region definitions

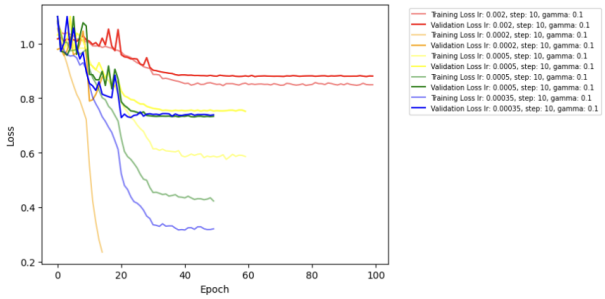

Figure 9: Region-based ensemble training and validation loss

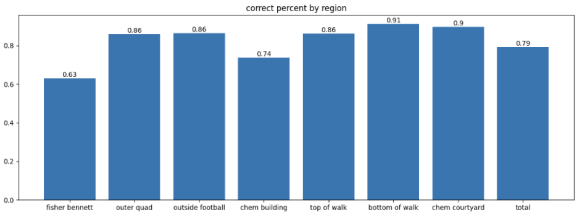

Figure 10: Ensemble correct % by region

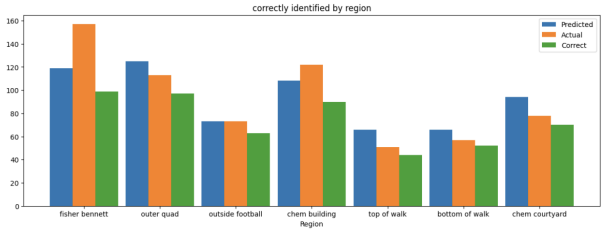

Figure 11: Ensemble correctly identified and misidentified by region

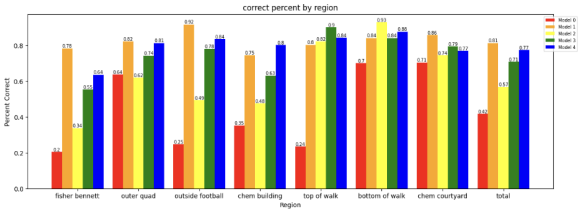

Figure 12: Ensemble base models correct % by region

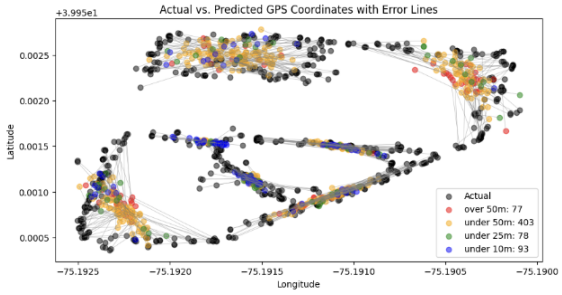

Figure 13: Given region predicted location

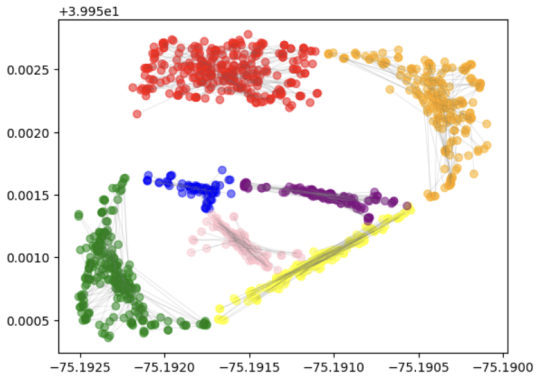

Figure 14: Given region predicted location by region

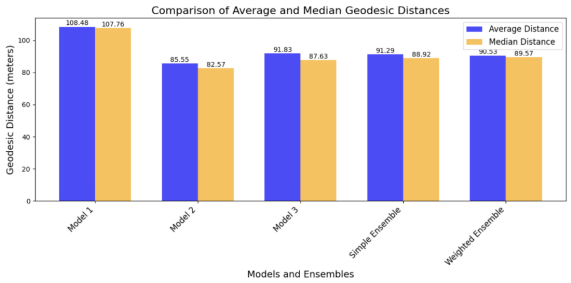

Figure 15: Ensemble comparison to average prediction accuracy

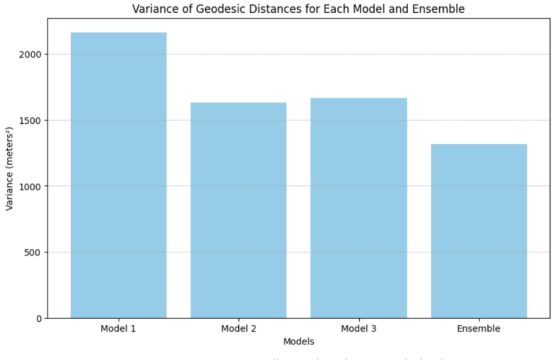

Figure 16: Ensemble variance comparison

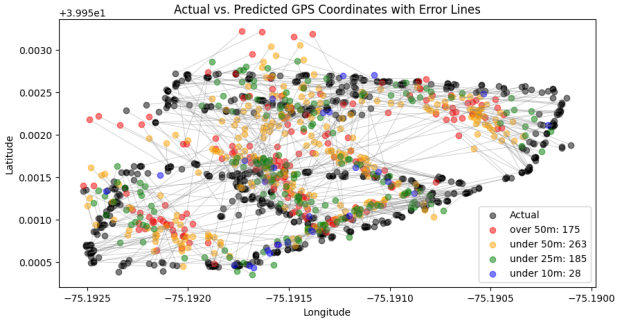

Figure 17: Region model predicted location

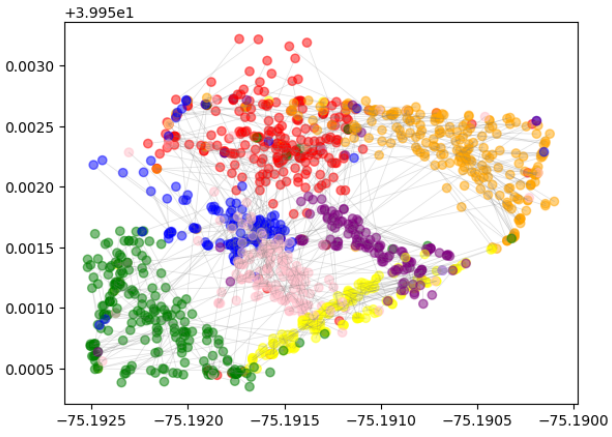

Figure 18: Region model predicted location by predicted region