ResNetBolt: Computer Vision and Geoguessr

Akhil Arularasu
akhil.arularasu@emory.edu
Emory

Max Roberts
max.roberts@emory.edu
Emory

Michi Okahata
michi.okahata@emory.edu
Emory

May 5, 2025

Abstract

Deep learning models are able to generalize over specific tasks after being trained on the order of hundreds of thousands to millions of training samples. Humans on the other hand seem to intuitively, efficiently, and broadly generalize with comparatively fewer training samples. Our project explores the task of image geolocation in games like GeoGuessr which exemplifies this dynamic. We built upon the existing literature to train variations on ResNet50 that outperform humans. Despite recent advancements, machine learning models frequently lack interpretability. Our novel contribution lies in exploring interpretability within the model and in comparison with humans.

Keywords neural networks · computer vision · geolocation · explainable AI

1 Introduction

Recent advancements in deep learning and computer vision have been spurred by innovative neural network architectures, e.g., convolutional and residual neural networks, yet model training remains computationally expensive and dependent on hundreds of thousands to millions of training samples. Furthermore, despite increasing accuracy across a variety of specific tasks, models remain a black box where explanation and interpretation are lacking (Zhao et al. 2024). Human learning on the other hand seems to intuitively, efficiently, and broadly generalize with comparatively fewer training samples. The task of image geolocation in games like GeoGuessr exemplifies this dynamic. Humans without extensive knowledge of geography do quite well and significantly improve over a handful of games which total only a couple dozen training samples. Deep learning models on the other hand would utterly fail without the requisite training data.

Ironically, the best GeoGuessr players are considered robotic in their ability to develop encyclopedic memories of different features that correspond to different regions, with some going as far as developing an intuition for when grass is Mongolian (Weber 2024). Other features include distinct street lamps, types of pavement, and native plants. For the best players, these features seem to generate a decision tree of possible locations. This relationship between human intuition and machine learning has drawn significant attention with many existing implementations of varying complexity (Hays and Efros 2008; Weyand, Kostrikov, and Philbin 2016; Suresh, Chodosh, and Abello 2018; Haas et al. 2024). Our project implements two simple variations on ResNet50 and explores various approaches to explain and interpret the resulting models.

2 Background

Image geolocation has a broad range of possible applications spanning defense, autonomous driving, and disaster response. The potential risks at the most high stakes end of the spectrum are so large that the weights of most complex and accurate models are not publicly released (Haas et al. 2024). We decided that pursuing state-of-the-art models and results exceeded the scope of the project given limited time and computational resources. Instead, we built two simple variations of ResNet50 as established in *DeepGeo* (Suresh, Chodosh, and Abello 2018).

GeoGuessr prompts players to guess the precise coordinates of a location based off of Google Street View images, akin to a regression problem. Instead, *DeepGeo* simplifies the task as a classification problem over 50 states. Unlike earlier efforts like *IM2GPS* and *PlaNet* which used datasets skewed towards famous geographic landmarks around the world, *DeepGeo* takes population density aware samples across all states in the United States.

DeepGeo open sourced the 50States10K and 50States2K datasets as the train and test datasets respectively for image geolocation and state classification. Both datasets are comprised of a 360 degree view of each location broken down into four images, one in each cardinal direction scraped from the Google Street View API. *DeepGeo* produced variations on ResNet50 in TensorFlow with their best-performing model achieving a top 1 accuracy of 38.32%, top 5 accuracy of 71.87%, and outperforming humans. Building on *DeepGeo*, we implemented the models in PyTorch, focusing on some hyper-parameter tuning and model interpretability.

While explainable AI has grown as an important research area in healthcare applications, there is no corresponding focus in the literature on image geolocation (Saraswat et al. 2022; Maheswari et al. 2024; Musthafa et al. 2024). The myopic focus on improving accuracy without concern for interpretability is a glaring gap given the potential risks established above. Our research address this gap by exploring explainable AI in image geolocation models through PCA and integrated gradients.

3 Dataset

We trained and tested our models on the 50States10K and 50States2K datasets described above. The training dataset is comprised of approximately 500,000 samples across all states. The dataset is perfectly balanced between classes with 10,000 samples per state. The testing dataset is the same with 2,000 samples per state. The preprocessing step involved cleaning the datasets for incomplete samples, e.g., samples with only 3 images at a certain location, whether due to a corrupted image or API limits, but the number of flagged samples is a rounding error relative to the size of the dataset.

The size of the training dataset also allowed us to comfortably split the 50States10K dataset for a 90:10 train and evaluation split. This allowed us to monitor training to tune hyper-parameters, prevent over-fitting, and importantly allowed us to entirely isolate the 50States2K dataset from the training process.

The datasets were normalized based on the mean and standard deviation of the 50States10K dataset to improve learning by accelerating the convergence of gradient descent (LeCun et al. 1998). We decided to train ResNet50 from scratch as the canonical classification task had little to do with the task of image geolocation. This also allowed us to maintain the 256×256 resolution of the dataset without resizing to 224×224 and losing information. Future work

could explore whether the pre-trained weights of ResNet50 could yield improved convergence, suggesting some overlap between ImageNet and image geolocation classification tasks. This comparison could shed light on the trade-off between faster training and accuracy when using or discarding the pre-trained weights.

4 Methods

4.1 ResNet

Our models are variations on ResNet50. ResNet solves vanishing gradients and the degradation problem at high network depth by introducing residual shortcut connections across residual blocks (He et al. 2016). ResNet models act as a classifier over any number of classes by modifying the final output layer.

$$y = F(x, \{W_i\}) + W_s x$$

Figure 1: Residual block

4.2 Hyper-parameters

Hyper-parameter	**Value**
learning rate	0.00001
betas	(0.9, 0.999)
epsilon	1e-8
weight decay	0.0001
num epochs	10
batch size	28
num workers	2

Table 1: Hyper-parameters for model training.

It was outside the scope of this paper to conduct a complete and exhaustive search of hyper-parameters since they are internally dependent (Zhao et al. 2024) and it takes around ten hours for the model to converge with even the most capable computational resources available on Google Colab. We got the best results from the hyper-parameters suggested by PyTorch's documentation and some minimal tuning. For example, decreasing the learning rate by an order of magnitude led to our fastest convergence and most accurate model. This improvement suggests that the initial learning rate and schedule was overshooting around and unable to get to the optimal minima.

Given greater computational resources and time, future work could more rigorously explore possible hyper-parameters using grid or random search over a range. We decided on ResNet50 given the existing literature and computational constraints so future work could also compare the number of layers in ResNet (e.g., 18, 34, 50, 101, 152) to explore the trade-off between computational cost and accuracy.

4.3 Early Integration

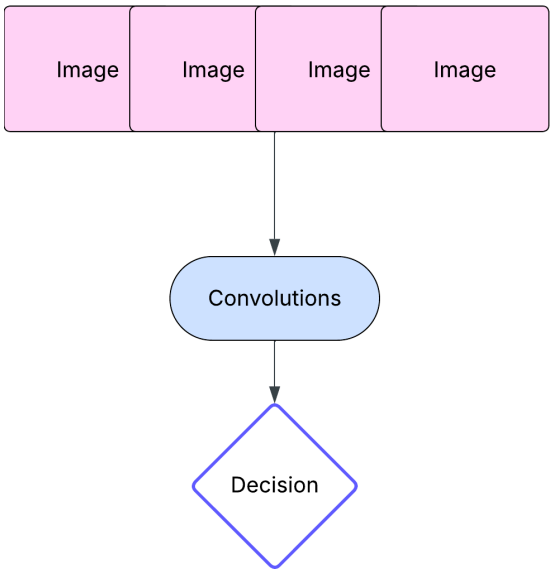

Figure 2: Early integration

The early integration pipeline concatenates the 4 images in each cardinal direction at the level of the data loader into one 256×1024 image. The early integration model has a corresponding input layer that is significantly larger than the canonical ResNet models but through repeated convolutions producers a final output layer for 50 state classification.

4.4 Medium Integration

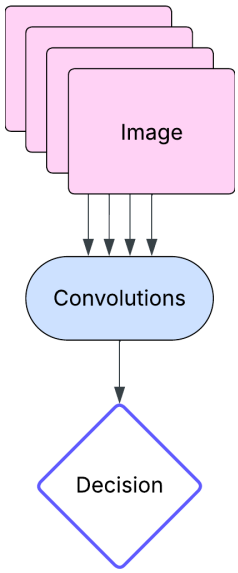

Figure 3: Medium integration

The medium integration pipeline shares the convolutional layers for a 256×256 input layer across the 4 images. These convolutional layers functionally aggregate the features across the 4 images. This approach is agnostic to the specific image a feature is found in which improves generalization. This flexibility contributes to the improved performance of the medium integration model compared to the early integration model.

5 Results

5.1 Training Metrics

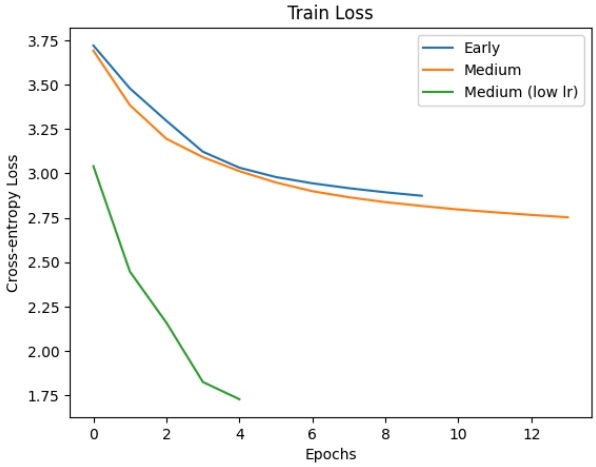

Figure 4: Models' train loss

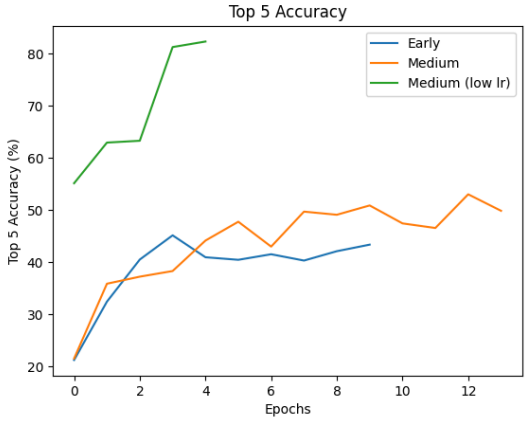

Figure 5: Models' top 5 accuracy

The models were trained with the above architectures and a cross-entropy loss function. The one hyper-parameter we explored with a significant impact was the learning rate (0.0001 and 0.00001) which sped up convergence in terms of increasing decreasing training loss and increasing evaluation accuracy. Model accuracy was assessed by top 1 and top 5 classification accuracy. Classification over 50 tasks is incredibly complex given the visual overlap among certain states in similar regions and climates. Our best performing model was the medium integration model trained on a learning rate of 0.00001 over 4 epochs with a top 1 accuracy of 48.28% and a top 5 accuracy of 82.30%.

Both model architectures stagnated at decent results with the initial learning rate with only a marginal improvement between the early and medium integration models. The medium model converged far better and faster with the modified learning rate which reflects our intuition and the existing literature. Future work given more computational resources and time could continue to train each model at the lower learning rate.

5.2 ROC Curves

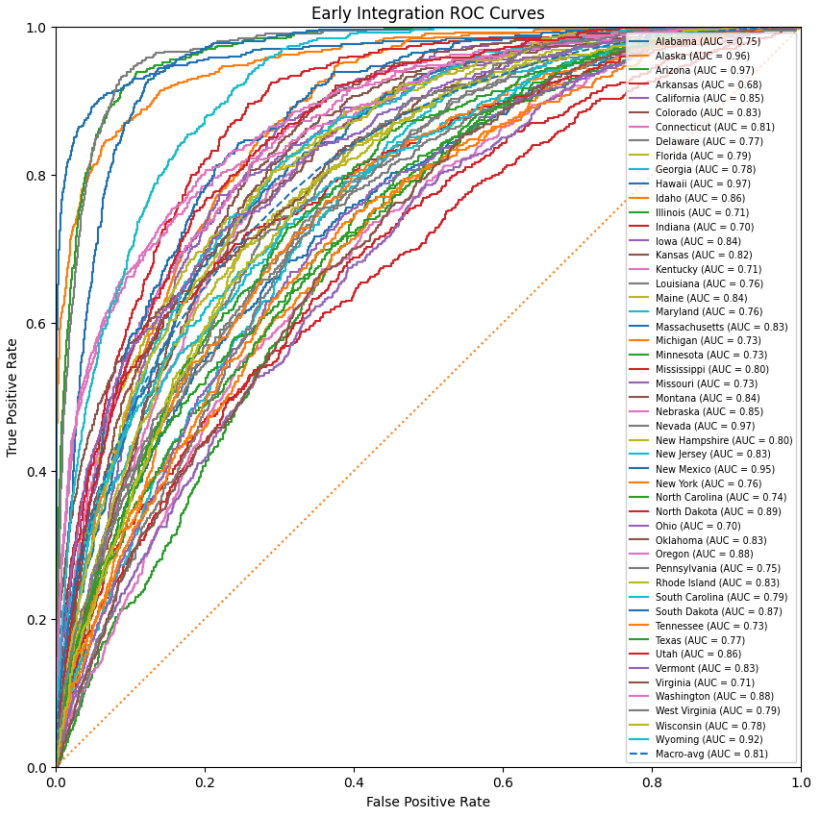

Figure 6: Early integration ROC curves

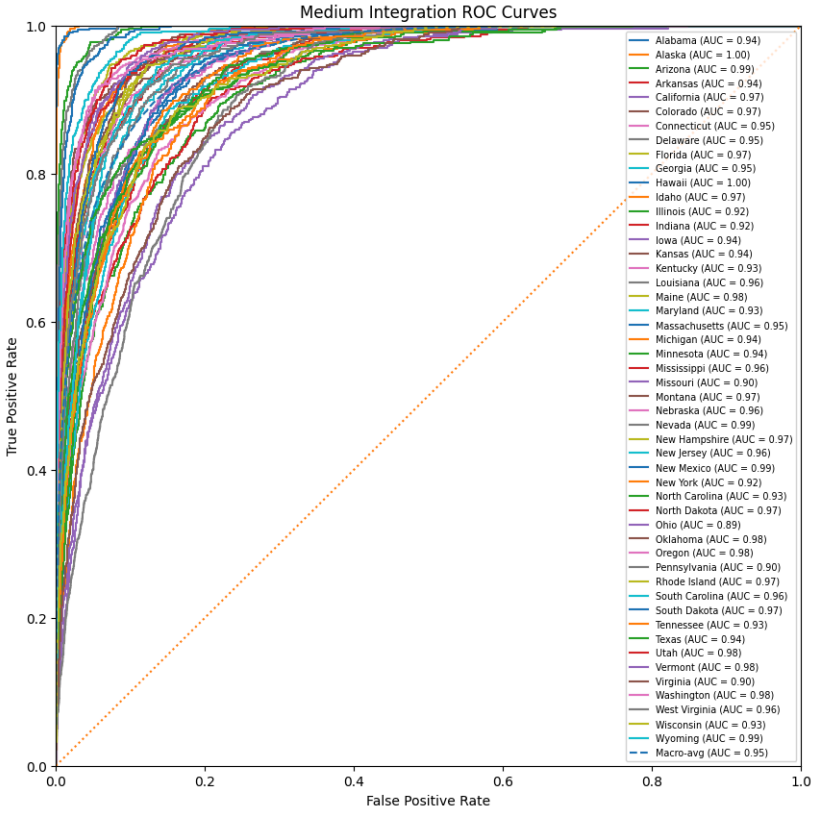

Figure 7: Medium integration ROC curves

Although the train and test datasets are perfectly balanced, with 10,000 and 2,000 samples per class respectively, the ROC curves demonstrates how well the model performs for each class in terms of both true positive and false positive rates. The ROC curves also help objectively evaluate the accuracy of a classification task with 50 classes. Even the worst state on the worst performing model (Arkansas, early integration) performs better than chance and the best state on the best performing model (Alaska, medium integration) is functionally perfectly classified.

The performance of different states also makes intuitive sense. The model is most proficient at classifying states with a dominating climate and corresponding features. For example, unique and distinct climates like the desert in Arizona, snow in Alaska, and sun in Hawaii are easier to classify and have above average ROC curves. Conversely, adjacent states in similar climates, e.g., mid-western states like Illinois, Indiana, and Ohio have below average ROC curves.

6 Discussion

6.1 Comparison

Figure 8: Ground truth: Alaska. Predicted: Alaska. Confidence: 0.9942.

Figure 9: Ground truth: Florida. Predicted: Florida. Confidence: 0.9069.

Figure 10: Ground truth: South Dakota. Predicted: New Mexico. Confidence: 0.2508.

Figure 11: Ground truth: Kansas. Predicted: Kentucky. Confidence: 0.1760.

We compared the early integration model to human guessing with a small sample of 10 locations. We opted for the early integration model as it was easier to analyze and present the images to people in a uniform way. The model achieved a top 5 accuracy of 70% while humans struggled with a top 5 accuracy of around 15%. This data was collected through a Google Forms quiz which prompted players to select their top 5 guesses and any intuition behind their guesses.

Even the early integration model confidently classifies distinct states like Alaska and struggles with ambiguous states like Kansas. Furthermore, the model is highly confident when the top guess is correct relative to when the top guess is incorrect. This difference supports our intuition that there are distinct states that are easily classified and overlapping states that are difficult to distinguish.

6.2 Principal Component Analysis

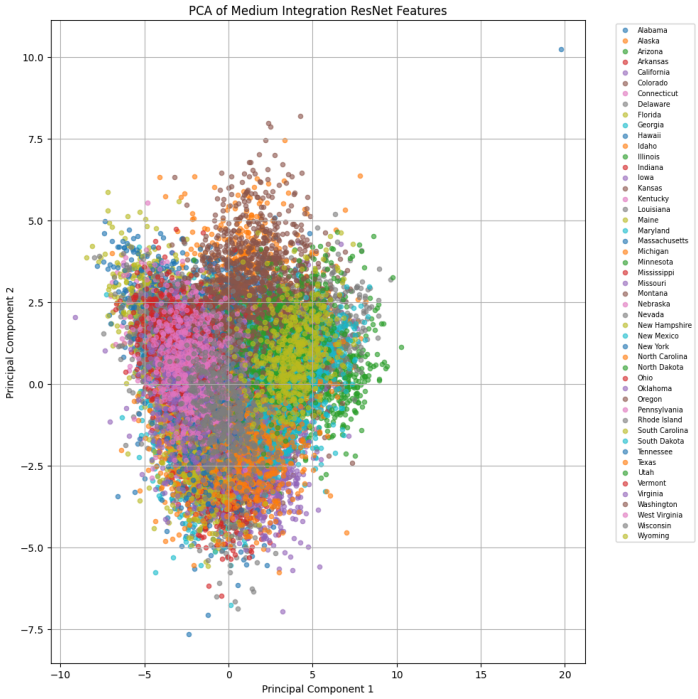

Figure 12: PCA of medium integration features

We performed Principal Component Analysis (PCA) on the 2048-dimensional feature vectors from the penultimate layer of our medium integration ResNet50 model for test samples. The PCA projects high-dimensional feature vector into a 2D space for visualization. The resulting PCA plot revealed significant overlap between state clusters, especially for visually similar regions such as the Midwest and South. This overlap underscores the inherent difficulty of image geolocation task and helps explain why top 5 accuracy significantly outperforms top 1, even when the model's top prediction is incorrect, the correct state is often among its top few guesses.

Conversely, we observed more distinct clustering for visually unique states like Alaska and Hawaii, aligning with their high classification accuracy. States with high intra-class variability or ambiguous visual cues, such as Texas or Virginia, did not form tight clusters, possibly indicating under-fitting or insufficient signal in the visual features. Overall, PCA highlighted the limitations of linear projections and reinforced the need for more discriminative feature extraction or alternative dimensionality reduction methods (e.g., t-SNE, UMAP) for clearer class separability.

6.3 Integrated Gradients

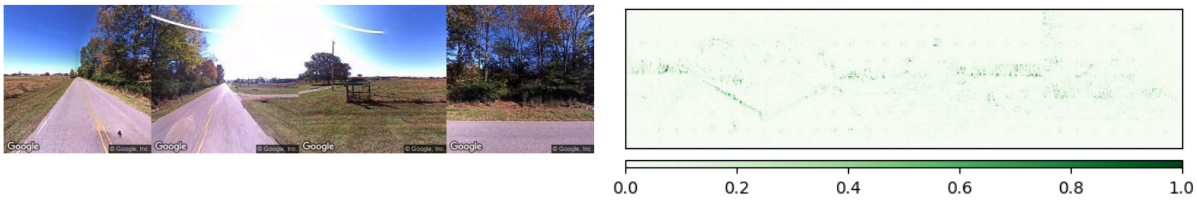

Figure 13: Ground truth: Alabama. Predicted: Indiana.

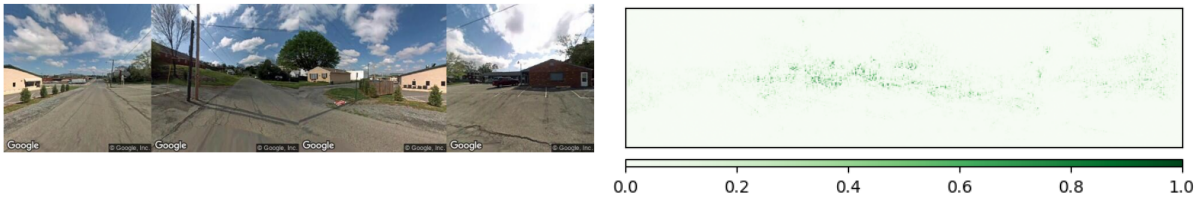

Figure 14: Ground truth: Virginia. Predicted: Iowa.

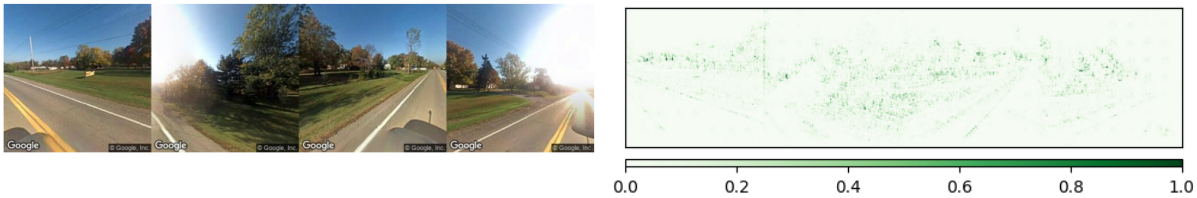

Figure 15: Ground truth: Ohio. Predicted: Ohio.

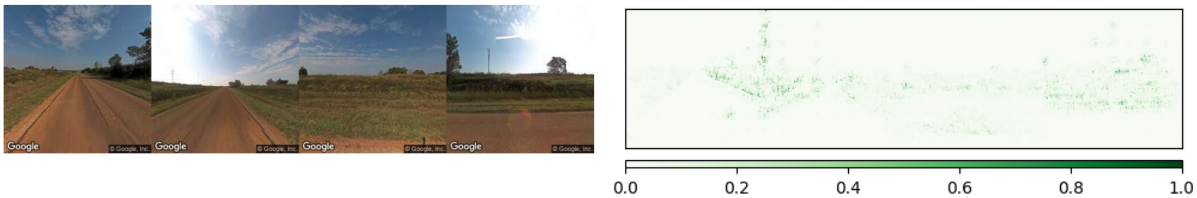

Figure 16: Ground truth: Kansas. Predicted: Kansas.

We applied integrated gradients from the captum library to the early integration model since it was the most straightforward and least error prone (Kokhlikyan et al. 2020). Integrated gradients contribute to interpretability by highlighting how sensitive the final classification decision is to some change in any feature (Sundararajan, Taly, and Yan 2017). The highlighted areas thus correspond to important features that contributed to and can be attributed for the decision.

The correspondence between integrated gradients and human intuition contributes to the literature on explainable AI in image geolocation and computer vision. These integrated gradients seem to pick up discrete segments in the image. Despite an incorrect prediction, figure 13 demonstrates how the model highlights the edge of the road and the horizon. Figure 15 and 16 demonstrate how the model highlights trees and vegetation to make a correct prediction. This interestingly reflects human intuition as some players were able to correctly classify figure 15 as Ohio in under 5 guesses due to the short trees and relative greenery which suggested a state in the upper mid-west.

6.4 Limitations

The significant overlap observed in extracted convolutional features, as indicated by PCA analysis, highlights a substantial limitation in current feature representation methods when applied to visually similar geographic scenes. The inherent similarity between images from geographically adjacent or similar states poses a critical challenge, suggesting that conventional CNNs may struggle to distinguish subtle visual cues effectively. Given the superior performance of medium integration in this task, future work could direct focus onto late integration to see if it has a similarly positive effect.

One key limitation is the model's inability to disentangle nuanced visual characteristics adequately. This issue can be attributed to the CNN's tendency to focus on dominant visual patterns rather than subtle, discriminative features. Future research should explore advanced feature disentanglement strategies such as variational autoencoders (VAEs) or adversarial methods that can explicitly separate different classes' distinctive elements, potentially enhancing model discriminative power.

Another avenue for future work is to explore t-SNE or UMAP as alternatives to PCA for feature visualization. Unlike PCA, which is linear and prioritizes global variance, t-SNE and UMAP are nonlinear techniques that better preserve local neighborhood structure in high-dimensional data. This makes them more effective for revealing meaningful clusters and class boundaries, especially in tasks like geolocation where subtle visual cues differentiate classes. Applying these methods could provide deeper insight into how well the model separates states and highlights areas of confusion more clearly.

Another promising direction involves incorporating spatial attention mechanisms into CNN architectures. Attention mechanisms can dynamically emphasize critical regions within images that carry the most informative cues, helping the model recognize subtle visual differences. Integrating attention-based methods like transformer architectures or spatial transformer networks (STNs) could significantly improve feature localization and enhance overall accuracy. When it comes to transformer architectures, we were limited by the sheer amount of time and compute it requires to train these models.

7 Contributions

Akhil Arularasu was responsible for implementing and analyzing the PCA visualization pipeline. He developed the PCA plots from the penultimate layer of ResNet50 and interpreted class separability and overlap.

Max Roberts was responsible for the original idea for the project itself. He collected human tests and compared the performance of our model with the performance of an average human. He also wrote the first draft of the paper.

Michi Okahata was responsible for model training, training metrics, ROC curves and integration gradient analysis. He revised and rewrote the final draft of the paper.

8 Code

Our code is in the following Google Colab notebook. Our model weights are in the following GitHub repository.

Bibliography

[1] X. Zhao, L. Wang, Y. Zhang, X. Han, M. Deveci, and M. Parmar, "A Review of Convolutional Neural Networks in Computer Vision," *Artificial Intelligence Review*, vol. 57, no. 4, p. 99, 2024, doi: 10.1007/s10462-024-10721-6.

[2] T. Weber, "He Memorized the World With Google Maps. Now He's Exploring It.," *The New York Times*, 2024, [Online]. Available: https://www.nytimes.com/2024/06/19/magazine/trevor-rainbolt-geoguessr-google-maps.html

[3] J. Hays and A. A. Efros, "IM2GPS: Estimating Geographic Information from a Single Image," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008, pp. 1–8. doi: 10.1109/CVPR.2008.4587784.

[4] T. Weyand, I. Kostrikov, and J. Philbin, "PlaNet - Photo Geolocation with Convolutional Neural Networks," in *European Conference on Computer Vision (ECCV)*, Springer, 2016, pp. 37–55. doi: 10.1007/978-3-319-46484-8_3.

[5] S. Suresh, N. Chodosh, and M. Abello, "DeepGeo: Photo Localization with Deep Neural Network," 2018, [Online]. Available: https://arxiv.org/abs/1810.03077

[6] L. Haas, M. Skreta, S. Alberti, and C. Finn, "PIGEON: Predicting Image Geolocations," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2024, pp. 12893–12902. [Online]. Available: https://openaccess.thecvf.com/content/CVPR2024/html/Haas_PIGEON_Predicting_Image_Geolocations_CVPR_2024_paper.html

[7] D. Saraswat *et al.*, "Explainable AI for Healthcare 5.0: Opportunities and Challenges," *IEEE Access*, vol. 10, pp. 84486–84517, 2022, doi: 10.1109/ACCESS.2022.3197671.

[8] B. U. Maheswari *et al.*, "Computer Vision and Explainable Approaches for Chest Tuberculosis Screenings," *BMC Medical Imaging*, vol. 24, no. 1, p. 32, 2024, doi: 10.1186/s12880-024-01202-x.

[9] M. Musthafa, T. R. Mahesh, V. V. Kumar, and S. Guluwadi, "Enhancing Brain Tumor Detection in MRI Images Through Explainable AI Using Grad-CAM with ResNet50," *BMC Medical Imaging*, vol. 24, no. 1, p. 107, 2024, doi: 10.1186/s12880-024-01292-7.

[10] Y. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient BackProp," *Neural Networks: Tricks of the Trade*. Springer, pp. 9–50, 1998. doi: 10.1007/3-540-49430-8_2.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778. [Online]. Available: https://arxiv.org/abs/1512.03385

[12] N. Kokhlikyan *et al.*, "Captum: A Unified and Generic Model Interpretability Library for PyTorch." [Online]. Available: https://arxiv.org/abs/2009.07896

[13] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic Attribution for Deep Networks," in *Proceedings of the 34th International Conference on Machine Learning*, D. Precup and Y. W. Teh, Eds., in Proceedings of Machine Learning Research, vol. 70. PMLR, 2017, pp. 3319–3328. [Online]. Available: https://proceedings.mlr.press/v70/sundararajan17a.html