GeoGuessr Country Classification with Deep and Classical Models

Aaditya Reddy Anugu   Justin Kang   Nathaniel Alexander Koehler   Patrick Soo   Zhixuan Wang

(Course Project Report; May 2024)

***Abstract*—We study automatic geolocation at the country level from Google Street View images in the spirit of the game *GeoGuessr*. Using a public dataset containing ∼50,000 images across 150+ countries, we benchmark three approaches: (1) a custom Convolutional Neural Network (CNN) with z-score standardization, (2) transfer learning with DenseNet121 and min–max normalization while undersampling the United States class, and (3) a classical k-Nearest Neighbors (KNN) model after a log transform. Across settings, severe class imbalance dominated outcomes; CNN achieved the highest reported test accuracy (24.62%) on an unbalanced split, while DenseNet121 achieved 5.36% and KNN 2.98% under more balanced evaluation. The results highlight the primacy of dataset balance and scale for geolocation, overshadowing model choice.**

***Index Terms*—Geolocation, GeoGuessr, Image Classification, Convolutional Neural Networks, Transfer Learning, DenseNet121, KNN, Class Imbalance.**

I. Introduction

GeoGuessr places a player at an unknown location in Google Street View and challenges them to guess the location. We pose a supervised variant: given a single Street View image, predict the country. While transfer learning has proven effective for image classification with limited data [3], geolocation at the country level introduces long-tailed label distributions, subtle region-specific cues (road markings, signage, vegetation), and strong domain imbalance.

We adopt the GeoLocation–GeoGuessr Images dataset from Kaggle (50K images across 150+ countries) [1] and evaluate three models under several preprocessing schemes. Beyond headline accuracies, we analyze confusion matrices, per-class reports, and training dynamics to understand bias and failure modes (Section V). Our study emphasizes that data quantity/quality and class balance dominate performance, often dwarfing architectural gains.[1]

II. Related Work

Transfer learning often yields strong performance on image tasks when data are scarce [3], [4], [2]. For geolocation, PlaNet [5] framed photo geolocation via CNNs with coarse-to-fine cell discretization of the globe, underscoring the challenge of long-tailed distributions and the importance of scale.

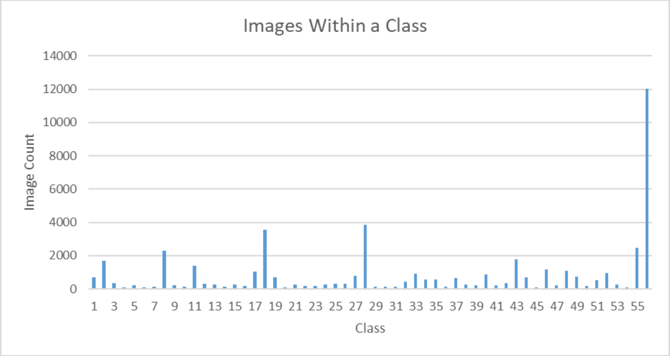

Fig. 1. Images per class: long-tailed distribution drives bias and instability across models.

III. Dataset

We use the Kaggle GeoGuessr dataset [1]. Original images are 1536×662. Following the original project report, we perform:

- **Class filtering:** Remove classes with < 100 images to reduce extreme sparsity.
- **Resize:** Downsample every remaining image to one-third size (512×220) to limit memory/compute.
- **Streaming I/O:** Use TensorFlow Datasets (TFDS) to avoid loading all images into memory.

Splits. We use a two-step split yielding a $60/15/25$ train/val/test ratio: (i) split $75\%/25\%$ into train/test, then (ii) carve 20% of the training set for validation. This setup is reused across models for comparability.

IV. Methods

We study three pipelines that differ in preprocessing and model family. Each pipeline inherits the class filtering and resizing in Section III.

A. Preprocessing #1: Z-Score Standardization + CNN

We standardize each image channel with

$$X' = \frac{X - \mu}{\sigma}. \tag{1}$$

CNNs are a natural fit for image data, extracting local features via learned filters. Our model stacks Conv2D + ReLU, MaxPooling, L1 regularization, and Dropout, culminating in a softmax classifier over countries. ReLU mitigates vanishing

[1] See the original project report for full charts/tables motivating these conclusions.

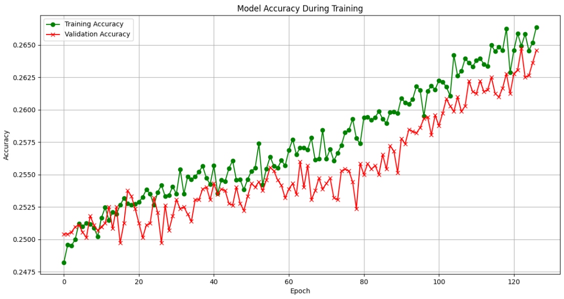

Fig. 2. CNN accuracy vs. epoch (train/val).

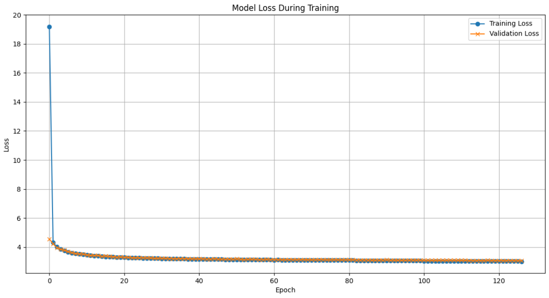

Fig. 3. CNN loss vs. epoch (train/val).

gradients; MaxPooling reduces spatial resolution and computation. L1 and Dropout discourage overfitting by promoting sparsity and stochastic regularization.

Training Dynamics (Observed): Training and validation accuracy increase together across epochs, while loss curves drop sharply early and then converge, suggesting the model learns useful features without severe overfitting under this setup (see Figs. 2, 3). However, downstream analyses (confusion matrix, precision/recall) reveal the model is *highly biased* toward the majority class (United States), reflecting the dataset's imbalance in the unbalanced split.

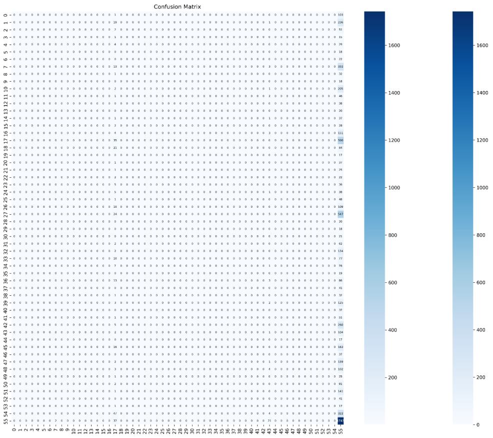

Fig. 4. CNN confusion matrix showing heavy prediction mass on the majority class and weak diagonal.

B. Preprocessing #2: Min–Max Scaling + DenseNet121 (Transfer Learning)

To reduce bias from the disproportionately large U.S. class, we undersample U.S. images to 300 before training and apply min–max scaling:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} . \tag{2}$$

We leverage ImageNet-pretrained DenseNet121 as a frozen feature extractor (transfer learning) with a small classification head and Dropout. Dense connectivity improves gradient flow and feature reuse; freezing early layers speeds convergence and limits overfitting when data are limited.

Training Dynamics (Observed): Compared to CNN, accuracy/loss trends plateau earlier under this more balanced regime, and confusion patterns spread across classes more evenly (weaker single-class collapse). Nonetheless, absolute accuracy is modest due to limited, imbalanced data even after undersampling.

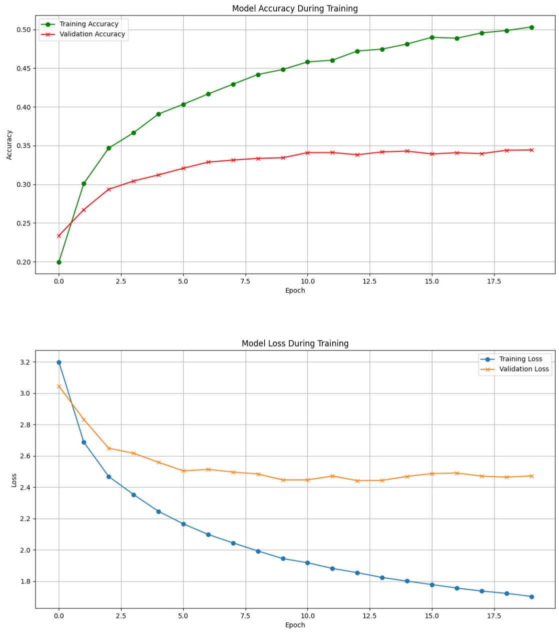

Fig. 5. DenseNet121 training curves: earlier plateaus in val. accuracy/loss than CNN after undersampling + min–max scaling.

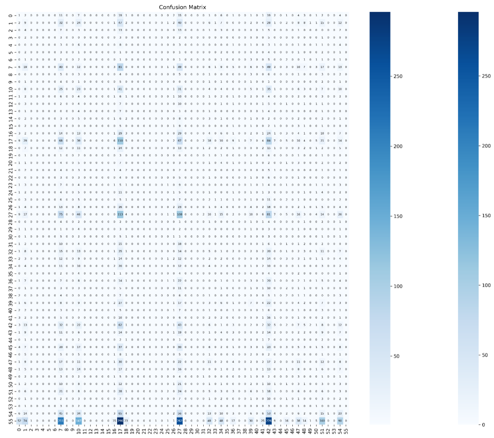

Fig. 6. DenseNet121 confusion matrix: reduced single-class collapse; errors distributed across classes.

C. Preprocessing #3: Log Transform + KNN

We apply a per-pixel transform $f(x) = \log(x)$ on RGB values in $[0, 255]$, maintain U.S. undersampling, and convert TFDS tensors to NumPy arrays to train scikit-learn KNN. Because KNN is distance-based with no learned parameters, it can struggle in very high dimensions. Memory constraints required further downscaling (e.g., to 56×384) to fit data and run queries efficiently.

Observed Behavior: Despite data balancing steps, KNN underperforms: in very high-dimensional spaces with limited per-class samples, nearest neighbors are noisy and often dominated by a few frequent patterns, yielding low accuracy and sporadic per-class recall.

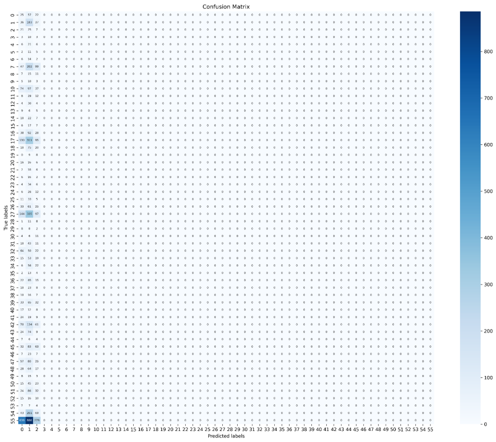

Fig. 7. KNN confusion matrix: concentration on a few labels, reflecting high-dimensional distance unreliability.

TABLE I
OVERALL TEST ACCURACY ACROSS MODELS (AS REPORTED).

Configuration	**Test Accuracy**
Standardization + CNN	24.62%[†]
Min–Max + DenseNet121	5.36%
Log + KNN	2.98%

[†] Likely inflated by an unbalanced test split dominated by the majority class.

TABLE II
CNN CLASSIFICATION REPORT (SUMMARY).

	Precision	**Recall**	**F1**
Macro avg	0.0069	0.0186	0.0089
Weighted avg	0.0720	0.2462	0.1067

V. RESULTS

We summarize overall test accuracy, then expand per-model analyses with training curves, confusion matrices, and precision/recall diagnostics.

A. CNN: Trends, Confusion, and Per-Class Metrics

Learning curves. Accuracy rises for both train/val; losses decrease and converge (Figs. 2, 3).

Confusion. The confusion matrix exhibits heavy mass on the majority class (United States), with little diagonal structure elsewhere, indicating bias toward the dominant label rather than genuine discrimination.

Classification report (summary). Accuracy $= 0.2462$. Macro averages are extremely low given many rare classes; weighted averages reflect majority-class dominance:

TABLE III
DENSENET121 CLASSIFICATION REPORT (SUMMARY).

	Precision	Recall	F1
Macro avg	0.0141	0.0207	0.0148
Weighted avg	0.0304	0.0536	0.0359

TABLE IV
KNN CLASSIFICATION REPORT (SUMMARY).

	Precision	Recall	F1
Macro avg	0.0011	0.0194	0.0022
Weighted avg	0.0019	0.0298	0.0035

B. DenseNet121: Effects of Undersampling and Normalization

Learning curves. Compared to CNN, validation accuracy/loss plateau earlier; transfer learning helps regularize but cannot overcome data scarcity and long-tailed structure alone.
Confusion. Predictions are less concentrated on a single class; errors are more evenly spread, consistent with reduced bias but still weak separability across many countries.
Classification report (summary). Accuracy = 0.0536; macro and weighted averages remain low due to many hard, under-represented classes.

C. KNN: High-Dimensional Distance Pitfalls

Confusion. KNN often collapses to a few labels (e.g., Argentina, Australia) despite undersampling, reflecting unreliable neighbors in high dimensions.
Classification report (summary). Accuracy = 0.0298, with near-zero macro/weighted precision and F1:

VI. DISCUSSION

Bias and calibration. CNN on an imbalanced split achieves seemingly higher accuracy by overpredicting the majority class (United States). DenseNet121 with undersampling exhibits less bias but limited absolute performance due to small effective sample sizes per class. KNN underperforms due to dimensionality and memory issues.

Why transfer learning did not dominate here. Although DenseNet121 is more expressive and benefits from ImageNet pretraining, the combination of (i) long-tailed labels, (ii) small per-class sample sizes, and (iii) geographic subtlety (fine-grained cues) limits generalization without substantial balancing and/or more data.

Evaluation protocol matters. Balanced, stratified test splits are essential. Metrics beyond top-1 accuracy (macro-averaged precision/recall/F1, calibration, top-k) better reflect model utility on rare classes.

VII. LIMITATIONS AND NEXT STEPS

Data. The dataset is small and highly imbalanced. Highest leverage: expand coverage and balance across countries. Augment carefully to avoid leakage and distribution shift; consider targeted augmentation (e.g., color/contrast, slight crops) that preserves geographic cues.

Modeling. Explore class-balanced losses, focal loss, and reweighting; fine-tune deeper layers of DenseNet121 (instead of fully freezing); consider modern geolocation architectures and self-supervised pretraining for domain adaptation.

Engineering. Use mixed-precision and efficient dataloaders; compress features (e.g., global pooling + PCA) before classical models; evaluate retrieval-style methods (nearest-neighbor on learned embeddings) as an alternative to KNN on raw/log pixels.

Evaluation. Ensure stratified, balanced test splits; add top-k accuracy and calibration metrics; analyze per-class performance to identify systematically confused regions and inform targeted data collection.

VIII. ACKNOWLEDGMENTS

We thank the maintainers of the Kaggle dataset and open-source tooling used throughout this project.

IX. AUTHOR CONTRIBUTIONS

- **Aaditya Reddy Anugu:** Introduction, Background, Gantt Chart, Slides.
- **Justin Kang:** Methods; Results & Discussion; Preprocessing.
- **Nathaniel A. Koehler:** Methods; GitHub Pages website.
- **Patrick Soo:** Problem Definition; Methods; Results & Discussion; Model Training.
- **Zhixuan Wang:** Problem Definition; Methods; Results; Preprocessing; Discussion.

REFERENCES

[1] R. K., “Geolocation – GeoGuessr images (50k),” *Kaggle*, 2024. [Online]. Available: https://www.kaggle.com/datasets/ubitquitin/geolocation-geoguessr-images-50k. Accessed: Feb. 20, 2024.

[2] A. Bhandari, “Feature scaling: Engineering, Normalization, and standardization (updated 2024),” *Analytics Vidhya*, 2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalizationstandardization/. Accessed: Feb. 20, 2024.

[3] H. E. Kim, A. Cosa-Linan, N. Santhanam, *et al.*, “Transfer learning for medical image classification: A literature review,” *BMC Medical Imaging*, vol. 22, no. 69, 2022. doi:10.1186/s12880-022-00793-7

[4] “Normalization — Machine Learning — Google for Developers,” [Online]. Available: https://developers.google.com/machine-learning/data-prep/transform/normalization. Accessed: Feb. 20, 2024.

[5] T. Weyand, I. Kostrikov, and J. Philbin, “PlaNet – Photo Geolocation with Convolutional Neural Networks,” in *European Conference on Computer Vision (ECCV)*, 2016.