

MVP

December 9, 2024

0.1 Minimum Viable Product - Geoguessr AI Locator Bot

Owen Kroeger

- [Video Link](#) -

- [GitHub Link](#) -

- [Jira / Confluence](#) -

0.1.1 Problem Inspiration

Geoguessr is an engaging game that challenges players to identify locations based on visual cues. The goal of this project is to combine machine learning and automation to tackle a unique coding challenge: creating a neural network that can predict a location based on an image and then automate gameplay in Geoguessr. This project is a fun and technical way to explore advanced topics in AI, image classification, and bot integration.

Objective

To build a system that: 1. Automatically predicts a country from a geotagged image using a neural network (LocatorBot).s 2. Uses this prediction model to handle new images provided by the GeoguessrAI and automates the map selection in the Geoguessr game.

This project combines the game of Geoguessr with the technical challenge of developing an end-to-end AI-powered bot.

0.1.2 MVP Goals

The MVP for this project includes:

1. Training an image classification model capable of predicting a country from geotagged images.
2. Integrating the trained model into GeoguessrAI to automate map location selection.
3. Demonstrating a functional pipeline from image prediction to map interaction.

0.1.3 Neural Network Overview

LocatorBot is a convolutional neural network (CNN) trained to predict the country of origin for an image based on visual features.

Libraries used - Pytorch for building and training the model - torchvision for data transformations and pretrained models - PIL for image preprocessing

Model Architecture - The core of LocatorBot is ResNet-18, a popular CNN architecture known for its efficiency. - The final fully connected layer was replaced with a layer matching the number of countries in our dataset - *Number of layers*: 18 residual layers - *Activation function*: ReLU - *Optimization*: Adam optimizer - *Loss function*: Cross-entropy loss

Dataset - Contains geotagged images, organized by folders representing different countries

0.1.4 Image Preprocessing

Steps - *Resizing*: All images are resized to 128 x 128 pixels - *Normalization*: Pixel values are normalized to have a mean of [0.485, 0.456, 0.406] and a standard deviation of [0.229, 0.224, 0.225] - *Data Splitting*: The dataset is split into training (70%), validation (20%), and test (10%) sets

Normalization ensures that the input data aligns with the expectations of the ResNet model, as it was pretrained on normalized ImageNet data. Resizing helps the model process images efficiently without memory overload.

0.1.5 Training and Evaluation

Training Process - The model was trained over 5 epochs with a batch size of 16, using the Adam optimizer with a learning rate of 0.001. - During training, the cross-entropy loss and validation accuracy was monitored to avoid overfitting.

Evaluation - After training, the model achieved an accuracy of 36% on the test set.

0.1.6 Integration with GeoguessrAI (Game Bot)

Purpose of GeoguessrAI - GeoguessrAI automates gameplay by taking screenshots, using LocatorBot for predictions, and selecting locations on the game map.

Workflow - Capture a screenshot of the game scene. - Preprocess the image using the same transformation pipeline as during training. - Use LocatorBot to predict the country. - Map the country to latitude and longitude using a predefined dictionary (country_coordinates.py). - Convert coordinates into pixel positions on the minimap. - Click on the predicted location and proceed to the next round.

0.1.7 Major Functions in GeoguessrAI

play_turn(bot: GeoBot, plot: bool): - Captures a screenshot, preprocesses it, gets predictions from LocatorBot, and selects the map location.

GeoBot.select_map_location(x: int, y: int) - Converts the latitude and longitude into minimap coordinates and clicks on the location.

GeoBot.extract_location_from_response(response: str) - Parses LocatorBot's output and converts it into coordinates.

GeoBot.lat_lon_to_mercator_map_pixels(lat: float, lon: float) - Transforms latitude and longitude into pixel positions on the minimap.

0.1.8 Next Steps

- **Expand Dataset**: Add more geotagged images, especially for underrepresented countries.

- **Improve Accuracy:** Use a more complicated model, with longer training times and more data to improve accuracy.
- **Regional Guessing:** Move to a regional guessing model instead of a country model to further enhance guess accuracy (multiple regions for larger countries).
- **Multiplayer / Duels:** Enhance GeoguessrAI to support the “Duels” game mode to allow players to battle the bot.