Geoguesser: Identifying Image Location by City with the GSV-Cities Dataset

Team: Aashraya Jha, Dante Bonolis, Francesca Balestrieri, Leonhard Hochfilzer, Zachary Bezemek

GitHub: https://github.com/hochfilzer/geo-locator

Background and Project Overview: In the popular online game Geoguessr, the player is shown a random image from Google Street View and is tasked with guessing their location on the globe as accurately as possible. Recently, students at Stanford created a project called PIGEON, which uses a deep learning model to play the game with great success, recently beating one of the world's top Geoguessr players. In this project, we seek to solve a simplified version of this problem but using a different methodology.

Our Goal: We create a model that can take an image taken outdoors in one of 17 different cities and identify in which of these cities the image was taken. We seek to explore if using domain-specific knowledge to our advantage can improve upon the "black box" deep learning strategy employed by PIGEON.

Stakeholders: Police, government/private security agencies, and investigative journalists may be interested in determining the location of an image they have either obtained in physical form or online with stripped metadata from, e.g., a social media website for investigative purposes. Model explainability could give rise to new useful strategies for professional Geoguessr players.

Modelling Approach: We make use of the publicly available GSV-Cities Dataset, which consists of around 500k street-view images taken in 23 different cities. Each image is given to an "end-to-end" CNN trained on this dataset, the backbone of which is a pre-trained model named MobileNetV2. In addition, 'key objects' used by human Geoguessr players, such as street signs and cars, are extracted from the image using object detection tools (mobilenet pre-trained on the CoCo dataset and Faster RCNN with a pre-trained ResNet50 backbone fine-tuned for detecting traffic signs). Each extracted "key object" is then sent to a lightweight, vanilla CNN trained on the task of identifying which city that specific type of object came from. The results from both the end-to-end location classifier and the object-specific classifiers are then aggregated via a weighted average in order to make a final prediction on which city the picture was taken in.

Key Performance Indicators:

Train-Test Split: We use a stratified split based on the city ID to ensure that the proportions of each class are consistent in the training and test sets. The dataset contains several images (an average of about 8) taken in the same place at different times, so we randomly sample 20% of the place_ids for each city to be in the testing set, and every image corresponding to that place_id is placed in the testing set.

Balancing: The dataset is highly imbalanced in terms of the number of images per city, with London being highly represented (>12%) compared to, e.g. Medellin (1%). Thus, we drop the 6 most under-represented cities from the dataset, reducing the problem of identifying images from among 17 cities. We also disallow a given place within a city to have more than 6 images associated with it, which greatly reduces the size of the dataset of the overly represented cities such as London, which originally had an average of over 11 images per place.

Results: We chose the baseline model to be a majority class classifier, which has an accuracy of 9.4 %. Our final accuracy of the end-to-end model is 63.5%. The object-specific classifier had accuracies ranging from 12.8% (for cars) all the way up to 34.9% (for buses). The weight for the end-to-end model is ~1 since it performs much better than the object-specific classifier. The accuracy of the final model for the top 5 guesses is 93.8%.

Looking at the confusion matrix, we note the model confuses cities in similar geographical locations, such as Barcelona and Paris. Cities that did not have a similar counterpart in the dataset, like Osaka, are predicted with higher accuracy. This is what we would expect from a human player, too.

Future Work:

- Find/develop more reliable object-detection tools
- Train on a more robust dataset representing more locations and with higher-resolution images
- Develop an application which allows the user to assist the network in making its predictions by confirming its object detection accuracy / manually detecting key objects
- Expanding the number of key objects supported
- Probing the end-to-end CNN in order to potentially discover new key objects for human Geoguessr players to use in their strategies