APS360 FINAL REPORT - PART A
USING DEEP LEARNING TO IDENTIFY LOCATIONS FROM STREET VIEW IMAGES

Alex Guo
Student# 1010142463
alexander.guo@mail.utoronto.ca

Ashwin Santhosh
Student# 1009936848
ashwin.santhosh@mail.utoronto.ca

Daniel Rolfe
Student# 1010208106
daniel.rolfe@mail.utoronto.ca

Nolan Young
Student# 1009905088
nolan.young@mail.utoronto.ca

ABSTRACT

The purpose of this document is to outline the design process and effectiveness of implementing neural networks to pinpoint image locations. Specifically, this document will primarily highlight the model's description and illustration, related works, data processing steps, the model's architecture, the baseline model, and the results. —-Total Pages: 9

1 INTRODUCTION

The web-based game, GeoGuessr, tasks a player to estimate the location of a provided image of a Google Maps street view (Google, 2024a) of a specific location in the world; a closer guess to the real location yields a higher score (GeoGuessr). For human players, a software engineer at Vercel discovered that on a test set with GeoGuessr images from around the world, humans guessed within 750 kilometers of an image's location only 13.9% of the time (Healey, 2021). This highlights that the task of image geolocation is highly difficult.

The primary goals of this project are to construct a GeoGuessr bot which assists GeoGuessr players, achieves an accuracy greater than both 50% and our baseline model, and can be implemented as a web-application, where users can upload photos and have the web-application guess their locations. In order to simplify and scale down our dataset in terms of scope, only U.S. and Canadian Google Maps street view (Google, 2024a) images will be utilized, and the bot will output coordinates within the U.S. and Canada. Unlike other models to be mentioned which predict countries of origin, our model will be dealing with a more enclosed landscape and thus will have to compensate for the lack of geographical differences in the training set compared to the other models' global datasets.

Trevor Rainbolt, a TikTok user famed for GeoGuessr expertise, was able to discover where a picture of a fan's deceased father on his honeymoon was taken (Gorton, 2022). From this, this GeoGuessr project will both provide players with the ability to test their GeoGuessr knowledge against an A.I. opponent and allow users to get a better sense of where in the U.S. or Canada their personal pictures were taken. The team is motivated to attain this functionality and to understand how potent neural networks can be constructed in terms of analyzing the attributes of an image and piecing together the image's distinct geographical components to arrive at a justified estimation on its location. Individuals on the team being familiar with GeoGuessr and the social media content pertaining to it also contributes positively to the team's knowledge of and personal investment in the project.

Since the project requires the extraction of unique characteristics from input images to function and make accurate predictions based on pattern recognition, a deep learning model must be used. This will permit the model to understand and pick up on feature representations and extrapolate intricate patterns from numerous images, rather than use defined steps and be incapable of recognizing complex patterns, specifically those which come with the unique characteristics of a geographical landscape (Archana & Jeevaraj, 2024).

2 ILLUSTRATION/FIGURE

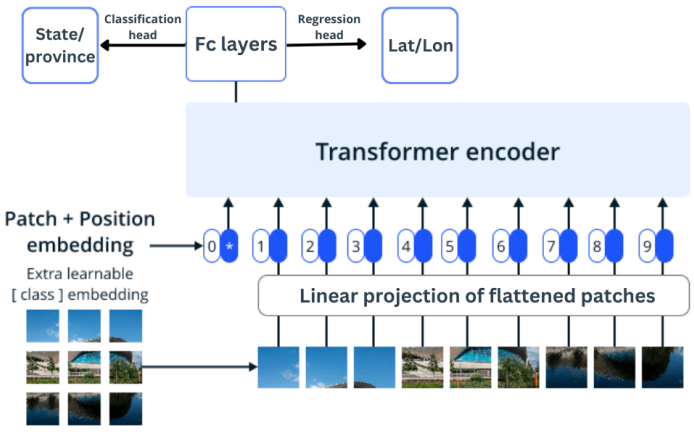

Figure 1: Architecture and process of our model. (Dosovitskiy et al., 2021)

3 BACKGROUND & RELATED WORK

From previous works, a 2018 paper on geolocation estimation of photos from the European Conference on Computer Vision suggests partitioning earth into geographic cells will allow a Convolutional Neural Network (CNN) to more easily distinguish between the unique geographic features of each subdivision and understand the geographical components typically present in a general area (Müller-Budack et al., 2018). While this approach would require additional work on the team's part, out of the models presented in the paper, the one which gained contextual knowledge of relevant locations through fine partitioning performed noticeably better than that which used a lone partition (Müller-Budack et al., 2018). For a GeoGuessr model, DeepGeo, PhD students at Carnegie Mellon University emphasized their use of ResNet, a CNN architecture which allows for the utilization of layers called residual blocks and skip connections, as it allows them to negate the issue of having a neural network with too many layers resulting in a high training and testing error; this approach allowed them to achieve a maximum accuracy of 71.87% and would allow the team to more deeply train a network (Sudharshan et al., 2018).

Similarly, a student at Brigham Young University also mentions their use of ResNet when creating their own GeoGuessr network, however this paper emphasizes the utilization of a recurrent CNN on a model which takes four images, one in each cardinal direction of the location, and memorizes each image when shown in a random order due to the recurrent CNN (Smith, 2021). The model outputs a longitude and latitude for a specific point in the world with an impressive GeoGuessr score of 18000 out of 25000 on average (Smith, 2021). From a related work, Students at Stanford University created a GeoGuessr network with some noticeable differences, the most notable being the use of ResNet50 and fine tuning (Dayton et al., 2023). While this program only receives one location image as input and was designed to output countries rather than coordinates or U.S. states, this fine tuning approach proved more effective than the author's initial feature extraction method, yielding accuracies of 72.5% and 40.3% percent on the same set of GeoGuessr images respectively (Dayton et al., 2023).

One last GeoGuessr model, PIGEON, created by another group of students at Stanford University, which also receives an input of four images in each cardinal direction from GeoGuessr, was trained on 400000 total images, and is able to guess within 25 kilometers of an image's location 40% of the time (Haas et al., 2024). This model also utilizes the concept of partitioning the world into geographic subdivisions, geocells, to alter the challenge into a classification problem and allow for an understanding of the unique characteristics of each geocell (Haas et al., 2024). The existence of

these student built projects contextualizes the difficulty of our project, in that given our timeframe, developing a model which can output a location within 25 kilometers of the answer 40% of the time is no longer out of reach and would be an ideal result considering the precision required in a game such as GeoGuessr.

4 Data Processing

For our GeoGuesser project, we used the Google Street View API (Google, 2024a) to collect images randomly from each province, territory, and state. This method employs a function and predefined regional borders to ensure a diverse and unbiased image set. Leveraging the API standardizes images according to Google's requirements, providing consistency across the dataset. These images capture real-world diversity, including unique geographical features, urban environments, and landmarks, which are crucial for training a model that generalizes well. Using the API also allows us to collect up-to-date images, reflecting recent changes in landscapes and infrastructure, maintaining the model's relevance and accuracy. The random selection process prevents bias from over-represented areas, ensuring balanced distribution and equitable performance across all regions. This approach simplifies the data processing pipeline and offers a scalable, automated solution for data collection. The same process was used to create a separate dataset with unseen coordinate image examples for testing, ensuring the model is evaluated on completely new data, providing an unbiased assessment of its performance.

First, we filtered out incomplete, duplicate, or irrelevant entries to ensure the dataset aligned with our geographical scope, maintaining data integrity and relevance (Shorten & Khoshgoftaar, 2019). Next, we inspected all images to confirm high quality, removing or replacing low-quality images to enhance the model's accuracy and robustness. Since the data was sourced from each region with 325 images per region saved in folders named after their respective regions, relabeling was unnecessary. The images were already organized by province, territory, and state, simplifying data management and ensuring consistency. We standardized the data into a uniform format for model training, resizing images to a consistent resolution and normalizing geographical data. As all images were from the Google Street View API (Google, 2024a), they were in the same file format, eliminating the need for additional formatting.

We applied data augmentation techniques such as cropping, greyscaling, colour jittering, random perspective and rotation randomizer to increase the training set's diversity and improve the model's robustness by simulating various environmental conditions and perspectives (Shuck, 2023). These techniques applied to only the training set helped prevent flawed validation and test accuracies to evaluate the model and also allowed the model to generalize better to new data. We automated the data cleaning and formatting processes to handle large volumes efficiently, using scripts and tools to ensure consistency and save time. We split the data into training, validation, and testing sets for effective model evaluation and tuning. This ensured the model was evaluated based on raw unseen data which were not augmented in any form, providing a reliable measure of its performance and readiness for deployment. This meticulous approach ensured our datasets were clean, consistent, and high-quality, providing a solid foundation for training an accurate and robust GeoGuessr model.

Figure 2: Visualization of applied data agumentations

5 Architecture

The final model consists of a pretrained Vision Transformer acting as the backbone, specifically the "google/vit-base-patch16-224-in21k" model (Hugging Face), followed by a fully connected neural network, named geoNet, which achieves both classification and regression. The team utilized

transfer learning with a Vision Transformer (ViT) model (Hugging Face). The transformer encoder model, ViT, has been pretrained on the ImageNet-21k dataset (Deng et al., 2009), which is a dataset consisting of over 14 million images belonging to over 21 thousand classes; these images were all scaled down to 224x224 pixels and had normalization applied across their RGB channels (Wu et al., 2020). ViT takes in the images as series of linearly embedded, fixed-size patches at a resolution of 16x16 pixels and requires a [CLS] token at the beginning of an inputted sequence and absolute position embeddings (Wu et al., 2020).

To ensure that the input images are consistently preprocessed in a manner that aligns with the transformations applied during the original training of the Vision Transformer, we utilize the ViTImageProcessor provided by the Transformers package from Hugging Face (Hugging Face). The pretrained model is then utilized in the process of extracting the features from the full dataset, as ViT has the necessary and extensive prior training on a wide variety of images to learn the inner representations of the team's images.

These features, labels, and coordinates are then utilized to create the necessary data loaders, allowing the images to be passed through the fully connected neural network. The fully connected neural network starts with a series of four fully connected layers. The first fully connected layer reduces the input features from 768 to 512 dimensions and is succeeded by a batch normalization layer to standardize the outputs. This process is repeated for the rest of the fully connected and batch normalization layers, with the second fully connected layer reducing the dimensions to 256, the third fully connected layer further reducing it down to 128, and the fourth fully connected layer finally bringing the features down to 64 dimensions. Every fully connected layer is succeeded by a corresponding batch normalization layer and using the ReLU function, the outputs of these layers are activated.

Following this process, the model uses two distinct heads. The first is the classifier which receives the 64-dimensional final layer output and returns its predictions for the 63 classes, which can then be used using cross entropy loss to determine loss and to extrapolate a state/province/territory from an inputted image. The second head, the regressor, also receives a 64-dimensional final layer output and predicts 2 values, one corresponding to the predicted latitude and the other corresponding to the predicted longitude, providing predicted coordinates of an image.

6 Baseline Model

The baseline model, geoCNN, is a convolutional neural network which seeks to achieve both classification and regression. For the convolutional layers of the model, the first convolutionl layer processes relevant RGB images and outputs 32 feature maps using a 3x3 kernel with padding, while the second covolutional layer expands upon this by producing 64 feature maps by also using a 3x3 kernel. To stabilize training, each convolutional layer is succeeded by a batch normalization layer, certifying that the feature maps are normalized before proceeding.

From this, the model utilizes a max-pooling layer with a 2x2 kernel and a stride of 2 to reduce the spatial dimensions of the feature maps, allowing for a decrease in the computational load and for the model to focus on the more notable features. Following this, a global average pooling layer is employed to further condense the feature maps into a single value per channel, which allows for a sizable reduction in the amount of parameters in the following layers while maintaining the necessary spatial information.

These features are then passed through a fully connected layer with 50 output units, allowing it to act as the gateway between the convolutional base and the final task-specific outputs. The model is designed to handle classification by using a classifier output layer with 63 units to predict one of the 63 states/provinces/territories and regression using the regression output layer which produces 2 values, one for latitude and the other for longitude.

A simple CNN was chosen as a baseline because the task of image location in and of itself is memory intensive and complex, so naturally a model which is a CNN similar to our final model, but is scaled down in terms of complexity provides a baseline model which is capable of learning and recognizing patterns without exploring the depths of what an A.I. model can really achieve in the realm of GeoGuessr. This model achieved a notably lower peak validation accuracy than our final model, yielding 38.6% as pictured in Figure 3 below.

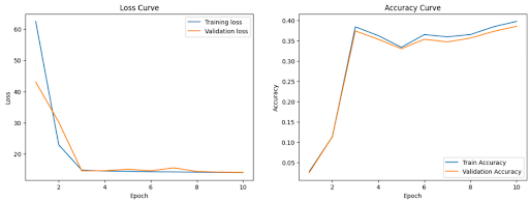

Figure 3: Training curves achieved by the baseline model, geoCNN.

When applied to our test set, the baseline model achieved a test accuracy of approximately 39.5%.

7 Quantitative Results

Our final model achieved a test accuracy of 52.10%. The training curve can be seen in Figure 4. Since GeoGuessr gives a higher score for a closer to actual location guess, we chose to model our scoring based on the following equation: Accuracy $= e^{-\frac{x}{1160}}$, where x is the great circle distance.

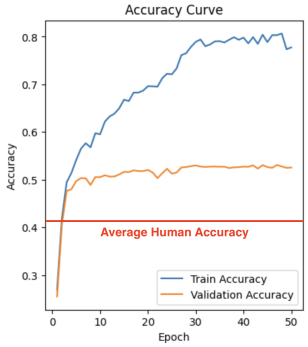

Figure 4: Training curve showing results of training and validation accuracy.

Converting the test accuracy to a distance metric, it is found that our model predicted 756 kilometres on average away from the ground truth. The test classification accuracy of our final model is 16.05%. Given this tough task, it is also worth looking at the top-N classification accuracy, which measures how often the true class label is among the top N predicted classes. The top-3 classification accuracy is 32.10%, top-5 is 43.48%, and top-10 is 62.28%. In addition, a confusion matrix, which can be seen in the Google Colab linked to this document, can help identify patterns such as classes that our model gets mixed up by. Since there are so many classes (63), Table 1 shows the precision and recall of a select few classes ranging from low to high performance. Precision shows the proportion of our model's positive classifications that are actually positive. Recall shows the proportion of all actual positives that were correctly classified as positives (Google, 2024b).

Class	Name	Precision	Recall
7	Colorado	0.0769	0.1111
52	South Dakota	0.2667	0.3636
50	Saskatchewan	0.2857	0.4615
60	Wisconsin	0.1852	0.5000
62	Yukon	0.3571	0.8333

Table 1: Selected classes with low, median, and high performance based on precision and recall.

8 QUALITATIVE RESULTS

To set the context, here is a sample of an input and output produced by our model.

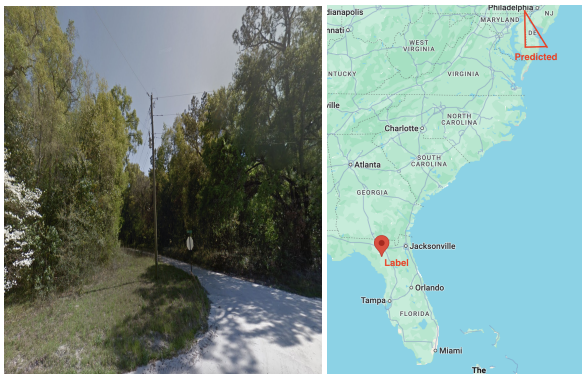

Figure 5: Input image from Florida (left) and model's prediction (right).

As we can see in Figure 5, our model predicted Delaware when the actual location is in Florida. The great-circle distance between the two states is 1180 kilometres(Scripts, 2024), which is worse than our 756 kilometres average on the test set. This puts into context our confusion matrix, where of the true Florida labels, it would confuse it with states like California, Colorado, Connecticut, and Delaware. One possible explanation for this case is that both are near the east-coast, and both are common to many coniferous forests and flat land (Lists, 2024).

To make more observations, a heat-map can be generated by overlaying the attention map across different heads from the final layer of the pretrained Vision Transformer, similar to the method used by researchers from Facebook (2021). For example, the attention map for the input in Figure 5 is shown in Figure 6. More sample outputs are shown in Figure 7.

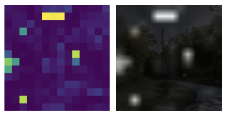

Figure 6: Sample input's attention map (left) and the map overlayed on the original image (right).

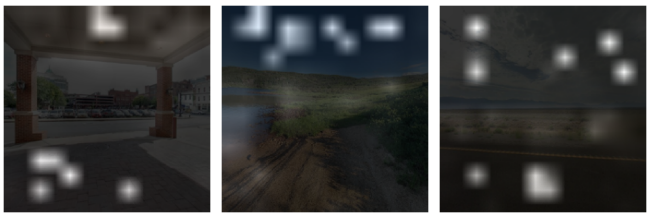

Figure 7: Sample 1 New York(left), Sample 2 Newfoundland & Labrador(middle), and Sample 3 Nevada(right).

Sample 1, 2, and 3 in Figure 7 are from our dataset. Sample 1 & 2 were selected as they performed noticeably poorly, at 2332 and 2787 kilometres away from the label respectively. For Sample 1, the

narrow view seems to have caused the attention of the pretrained Vision Transformer to focus on the ceiling and floor too much instead of the scenery in the background, correlating to the poor performance. Sample 2 seems to suffer the opposite issue, where the scenery is complex and attention is spread everywhere. Sample 3 with relatively simple scenery performed very well with a prediction of only 14 kilometres away from the label.

9 Evaluate Model On New Data

As mentioned in our results, the model achieved approximately 52.10% accuracy on the test data, which was left separate from the training and validation data and was only used once. While sufficient for testing the model's efficacy on new data, the team decided to consult an additional source beyond the Google Street View API to demonstrate the model's intended performance. They explored large geolocation-based datasets on Kaggle, but each proved insufficient for testing—many were too large, lacked coordinates, or did not accurately capture unique geographical features. For example, the Muninn dataset, a Kaggle dataset with 15,000 images labeled with coordinates, was not only too large and required scaling down but also had a potential difference of up to 500 kilometers between labeled coordinates and the actual image location, which would have severely skewed and hampered test accuracies (Tsegai, 2024).

From this, the team decided to approach the task of further assessing the effectiveness of the model by adopting the perspective of the user, in that the user would likely have a small batch of photos they would wish to evaluate rather than an extensive dataset or would be playing the game GeoGuessr for only a few rounds and with a few images. To not utilize the Google Street View API again but still use Google Maps given its utilization in the GeoGuessr game, the team compiled a dataset of 24 screenshots manually taken from Google Maps, that were each spread out across the two countries and each from a different state/province/territory. These screenshots were obtained by opening Google Maps, clicking on a random location within a predetermined state/province/territory and taking a screenshot of the area selected. Examples of screenshots in this dataset are pictured in Figure 8 below.

Figure 8: Screenshots taken in Google Maps Street View of Arizona (left), British Columbia (center), and Maine (right).

This dataset was pushed through the model in the same manner as the primary test set and the dataset achieved an average error distance of approximately 875.1 kilometers and a test accuracy of approximately 50.3%, which only differs slightly from the primary test accuracy.

To further demonstrate the capabilities of the model on new data, a dataset composed of 20 images from a team member's iPhone camera roll was employed. The model should and can be used to predict a user's images' locations, so taking a small dataset of camera roll images is an accurate depiction of the data that the model will use on a frequent basis. Furthermore, the small size allows the dataset to be monitored and controlled, as the team is able to check the entire contents of the dataset swiftly. This dataset contained an equal split of images from Arizona, California, Ontario, and Quebec. Employing this dataset, an average error distance of approximately 827.6 kilometers and a test accuracy of approximately 52.4% is achieved, demonstrating that our model is consistently strong with quality test data. Cropped examples from this dataset can be pictured in Figure 9 below.

Figure 9: Team Member's iPhone camera roll cropped photos taken in California (left) and Ontario (right).

10 Discussion

The model's peak validation accuracy of 53.1% and test accuracy of 52.10% demonstrate its effectiveness in geolocation prediction, surpassing the 50% baseline and outperforming other approaches, such as a Stanford model with 40.3% accuracy using feature extraction. Even without fine-tuning, our model competes well, especially compared to another Stanford model with 72.5% accuracy. While strong, there is room for improvement through fine-tuning, data augmentation, or metadata integration, which could further enhance accuracy, generalization, and reliability. These results underscore the potential for continued development to achieve even greater robustness.

In the early stages of development, before data augmentation, the model began differentiating regions based on the image author's tag in the bottom right corner. This unintended learning revealed the model's reliance on superficial features rather than the geographical and environmental cues essential for accurate geolocation. The author's tag led to artificially high accuracy, as the model focused on metadata instead of image content. To address this, we cropped the bottom pixels of the images to remove visible tags, forcing the model to focus on relevant geographical features and environmental characteristics.

Our model showed mixed performance in distinguishing regions with similar ecological, biome, and geological features, leading to some confusion in predictions. However, it excelled in certain regions, as seen in the confusion matrix. Delaware and Saskatchewan had the most correct predictions, with 6 each, likely due to Delaware's small size and coastal geography and Saskatchewan's vast prairies and unique landscapes. British Columbia, New Hampshire, Wisconsin, and Yukon also had high accuracy, with 5 correct predictions each, attributed to distinctive landscapes like British Columbia's coastal terrain and Yukon's wilderness. However, the model struggled with regions sharing similar features, such as the prairie landscapes of North and South Dakota and the Midwestern agricultural environments of Indiana and Illinois, leading to confusion in street view images.

The confusion matrix's darker diagonal shows the model's ability to recognize key regional features, effectively capturing geographical and environmental cues. However, where the model struggled highlights the limitations of relying solely on visual data, especially in regions with similar ecological characteristics. This suggests the potential benefits of incorporating additional data sources—like metadata, climate data, or region-specific landmarks—to better distinguish visually similar regions. While the model is on the right track, there's room for improvement, particularly with more time and computational power, to enhance its ability to differentiate subtle regional differences.

Through this project, we learned that building a geolocation prediction model requires both a sophisticated neural network and meticulous data preprocessing. A key insight was recognizing the model's tendency to focus on irrelevant features, like an author's tag, highlighting the need to avoid biases. We also faced challenges in balancing the dataset to prevent overfitting, particularly with unevenly distributed data, where certain regions might dominate training and lead to inconsistent performance elsewhere. Fine-tuning proved essential for improving accuracy, and the difficulty of distinguishing visually similar regions emphasized the need for diverse data sources or more granular information to improve generalization. Overall, this project deepened our understanding of neural networks' handling of complex visual data and reinforced the value of iterative refinement in developing robust applications. Comparing our model to others, with accuracies ranging from 40.3% to 72.5%, showed it performs well within the expected range and even exceeds some benchmarks.

11 ETHICAL CONSIDERATIONS

Extensive precautions were taken during data collection and model design to address potential ethical issues. Our model was carefully trained on a balanced and diverse dataset to mitigate biases, such as those observed in datasets like GeoLocation (Rohan, 2021). We ensured equitable representation across different regions, including both developed and less developed countries. This approach minimizes the risk of the model disproportionately favoring images from more developed countries, thus enhancing its accuracy in recognizing diverse geographical locations. By avoiding reliance on a limited number of images from specific countries, we have addressed the "danger of a single story" (Adichie, 2009), ensuring that no single image unfairly represents an entire nation.

Moreover, the Google Streetview images we sourced have already implemented robust preventative measures to protect personal information. These include address blurring, face/identity blurring, and number plate blurring, effectively mitigating the risk of the model being misused for purposes such as doxxing where personal information could be revealed without consent (Nguyen, 2023). By using images with these built-in privacy protections, we have ensured that our model adheres to strict privacy-preserving standards, further safeguarding against any potential misuse.

12 PROJECT DIFFICULTY / QUALITY

With a deep learning model, the difficulty of a task given to perform generally correlates to the human ability to complete the same task. For instance, identifying an animal is generally an easy task for a human, and similarly, an easy task for a deep learning model. Our project involved geolocation, which is a very challenging task that most humans, unless extremely well knowledged, perform poorly at, which is evidenced by the average human score (which is already inflated due to the higher skill level of those that regularly play the Geoguessr game) on the Geoguessr game, 10486 points out of 25000 (corresponding to an accuracy of 41.9%) (GeoGuessr, 2024). This human score, despite being fairly low, is still greatly inflated by a player bias, since those who play the game are generally well-practiced and have lots of experience on Geoguessr, therefore not being representative of the average human. There are a variety of reasons that contribute to the difficulty of a task involving geolocation, including very limited information that can come from a streetview image, visual similarity between multiple regions, and extremely small and specific features that serve as clues to a location. Furthermore, the model is given significantly less information then what a human player would receive, given that the model receives a static image of a street, whereas an actual player would receive an entire 360 degree and movable view. It's worth noting how the score in Geoguessr is calculated, given that it is what we based our model accuracy on. The game is very rewarding for short distances between the predicted location and actual location, but will always reward some number of points no matter how far the difference is. Using data from actual Geogeussr games, a user from the Geoguessr community (Smith, 2017) derived the formula for how the score was calculated to be approximately $5000e^{-\frac{x}{Z}}$, where $\boldsymbol{Z}$ was a constant that corresponded to the scale of the land mass over which the game was to be played, which the user calculated to be 820 km for the USA. Given that our model was scoped to both USA and Canada, scaling $\boldsymbol{Z}$ up in proportion to the landmass (becoming 1160 km) led us to find that the score function for our region was approximately $5000e^{-\frac{x}{1160}}$, corresponding to an accuracy of $e^{-\frac{x}{1160}}$, meaning to even reach an accuracy of 50%, the model would have to get on average within 800 km of the correct region for each image in our dataset, a difficult task given that Canada and the USA combined span approximately 5500 km from east to west (of Canada, 2016) and 7250 km from north to south (O'Neill, 2024). Given that our model reached a validation accuracy of 53.1% and test accuracy of 52.10%, we can say with confidence that given the difficulty of the project, our model was greatly successful.

REFERENCES

C.N. Adichie. The danger of a single story. https://www.ted.com/talks/chimamanda_ngozi_adichie_the_danger_of_a_single_story, 2009. Accessed: 2024.

R. Archana and P.S.E. Jeevaraj. Deep learning models for digital image processing: a review. *Artifical Intelligence Review*, 57, 2024.

F. Dayton, J. Heo, and E. Werner. Cnn plays geoguessr: Transfer learning on resnet50 for classifying street view image. https://www.finndayton.com/CS229_Final_Report.pdf, 2023.

J. Deng, W. Dong, R. Socher, L. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Facebook. Dino. https://github.com/facebookresearch/dino/blob/main/LICENSE/, 2021.

GeoGuessr. Geoguessr. https://www.geoguessr.com/, 2013.

GeoGuessr. World - map - geoguessr, 2024. URL https://www.geoguessr.com/maps/world.

Google. Street view static api. https://developers.google.com/maps/documentation/streetview?_gl=1*1pm7q35*_up*MQ..*_ga*Mjc3ODUxNTA1LjE3MjAxMjExNDg.*_ga_NRWSTWS78N*MTcyMDEyMTE0Ny4xLjAuMTcyMDEyMTE0Ny4wLjAuMA.., 2024a. Accessed: June 2024.

Google. Classification: Accuracy, recall, precision, and related metrics. https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall/, 2024b.

T. Gorton. Trevor rainbolt: the man who saw the world. *The Face*, 2022.

L. Haas, M. Skreta, and S. Alberti. Pigeon: Predicting image geolocations. https://arxiv.org/pdf/2307.05845, 2024.

A. Healey. Geoguessing with deep learning. https://healeycodes.com/geoguessing-with-deep-learning, 2021.

Hugging Face. google/vit-base-patch16-224-in21k. https://huggingface.co/google/vit-base-patch16-224-in21k.

Objective Lists. The most similar states to delaware. https://objectivelists.com/which-states-are-most-similar-to-delaware/, 2024.

E. Müller-Budack, K. Pustu-Iren, and R. Ewerth. Geolocation estimation of photos using a hierarchical model and scene classification. In *Computer Vision – ECCV 2018*, pp. 575–592, 2018.

S. Nguyen. What is doxxing and what can you do if you are doxxed. https://www.cnn.com/2023/02/07/world/what-is-doxxing-explainer-as-equals-intl-cmd/index.html, 2023. Accessed: 2024.

Government of Canada. Highlights of canada's geography. https://www150.statcan.gc.ca/n1/pub/11-402-x/2012000/chap/geo/geo-eng.htm#, 2016.

A. O'Neill. Geography of the united states - statistics & facts. https://www.statista.com/topics/9782/geography-of-the-united-states/#topicOverview, 2024.

K. Rohan. Geolocation - geoguessr images (50k). https://www.kaggle.com/datasets/ubitquitin/geolocation-geoguessr-images-50k, 2021. Accessed: June 2024.

Movable Type Scripts. Calculate distance, bearing and more between latitude/longitude points. https://www.movable-type.co.uk/scripts/latlong.html/, 2024.

C. Shorten and T.M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 2019. URL https://journalofbigdata.springeropen.com/articles/10.1186/s40537-019-0197-0#citeas.

C. Shuck. What is data augmentation in machine learning. https://robots.net/fintech/what-is-data-augmentation-in-machine-learning/, 2023.

B. Smith. 474 final project - geoguesser ai. https://justbraydensmith.com/CS474Report.pdf, 2021.

J. Smith. Geomath: Deriving the geoguessr point formula. https://www.reddit.com/r/geoguessr/comments/7fon8u/geomath_deriving_the_geoguessr_point_formula/, 2017.

S. Sudharshan, N. Chodosh, and M. Abello. Deepgeo: Photo localization with deep neural network. https://arxiv.org/pdf/1810.03077, 2018.

S. Tsegai. Muninn dataset (15k). https://www.kaggle.com/datasets/samsontsegai/muninn-dataset-15k, 2024.

B. Wu, C. Xu, X. Dai, A. Wan, P. Zhang, Z. Yan, M. Tomizuka, J. Gonzalez, K. Keutzer, and P. Vajda. Visual transformers: Token-based image representation and processing for computer vision, 2020.