Exploring CLIP's Geo-location capabilities

Alan Ispani and Johan Laursen
IT University of Copenhagen
{alai,jocl}@itu.dk
https://github.com/Ispanicus/geoguessr-ai

Abstract

We used OpenAI's CLIP model to determine where in the world a photo was taken in order to test its ability to play the online browser game GeoGuessr. We utilised the Mapillary dataset as a substitute for the Google Street Maps used in GeoGuessr and performed a variety of geo-location experiments including country, city, U.S states, and French regions. Moreover, we performed experiments in an attempt to provide CLIP with extra information to improve results, but they were not successful. We show that for geo-location of images by predicting place names CLIP is better than a state-of-the-art coordinate guessing approach by comparing CLIP's results to GeoEstimation, the best performing coordinate geo-location model.

1 Introduction

When trying to solve tasks using machine learning, a task-specific model is typically built. In order to perform any other task, the model needs to be fine-tuned with a new domain specific dataset. This cross domain transfer is often costly, due to the need for curating a new dataset and retraining of the model. As models become exponentially larger (Bernstein et al., 2021), the cost of transfer learning may become infeasible. Building large generalised language models, such as GPT-3 (Brown et al., 2020) and GLAM (Du et al., 2021), which can be applied to a wide array of tasks, has become more common in recent years. In this vein, CLIP (Contrastive Language–Image Pre-training) (Radford et al., 2021) looks to combine generalizability across domains with training efficiency in the visual recognition field. It can generate a distribution of possible captions for a given image, as shown in Figure 1.

Since CLIP is not fine-tuned to any specific task, we look to explore its limitations and compare it to a specialised visual model. This paper thus attempts to examine CLIP's geo-location capabilities, in particular its ability to play Geoguessr [1], the geographical browser game in which players guess locations from street view imagery.

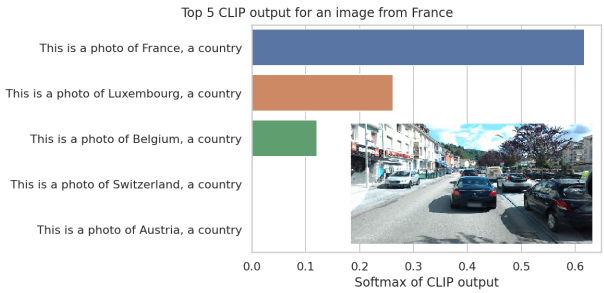

Figure 1: Example CLIP prompt and probability distribution for country prediction.

2 Data

2.1 Country211

The original CLIP paper used the Country211 dataset for geolocation testing. This dataset consists of images from 211 different countries with an equal distribution (300 images for each country). This dataset is taken from the YFCC100m dataset (Thomee et al., 2016) by filtering for images with GPS coordinates corresponding to a ISO-3166 country code[2]. The images from this dataset are varied in content with close-ups of flowers and selfies in hotel rooms being just as common as photos of landmarks. Since the inspiration for this project was Geoguessr, we decided this dataset was inadequate for testing geolocation of outdoor images.

2.2 Mapillary

After an initial attempt at constructing a Google street view dataset using the Google API, which was scrapped due to prohibitive costs of using the API, we found the Mapillary dataset as an alternative. This dataset was the source of all the images

[1] https://www.geoguessr.com/
[2] https://www.iso.org/iso-3166-country-codes.html

in our test sets. The Mapillary Planet-Scale Depth Dataset[3], is a dataset of 50,000 image clusters with 750,000 total images covering varying geographical areas globally. Mapillary was especially suited for our task since all images are of outside areas. Furthermore, this dataset has been used for Geoguessr in the past as a free alternative to Google's Street View data, so it was an ideal replacement for Country211.

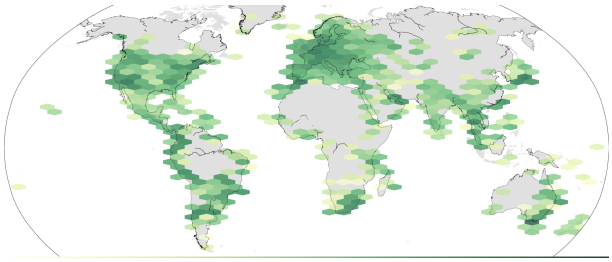

Figure 2: Mapillary spatial distribution from Mapillary website. Darker regions contain more images

The dataset is 250GB in size, which provided a challenge of where to store the data and how to access it. Most of our experiments were run on an HPC (High Performance Computing) cluster provided by ITU. ITU's HPC cluster is a collection of servers on which researchers can schedule resource-intensive tasks. These tasks typically need a large number of computational resources, such as a lot of CPU cores, high-performance GPUs, and large quantities of RAM. Researchers can schedule jobs using a scheduler that provides fair access to these resources.

We initially attempted to move the zipped data to the cluster using the Linux scp command. This failed since scp cannot resume aborted downloads caused by random network interruptions, which are almost unavoidable with a 10 hour upload time. We eventually settled on using the Linux command rsync as it provided a reattempt functionality that allowed for random failures during the upload.

2.3 LAION-400M

We used several different types of labels in our experiments such as country, state, and city names. To get an idea of how well CLIP understands these labels, we wanted a count of their appearance in CLIP's training data. Some of the labels, in particular city names, might not exist in the training data for CLIP, so getting a count and checking this was important to explain failures in CLIP's predictions. Furthermore, this count allowed us to compare the distribution of CLIP's predictions among the classes to the class counts. This comparison provided information about the relationship between word presence in the training data and CLIP prediction accuracy.

The CLIP model itself was trained on 400 million (image, text) pairs collected from a variety of publicly available sources on the Internet. The exact distribution is unknown, since OpenAI has chosen not to publicly disclose this. A private request for the distribution was denied, citing lack of legal clearance. However, CLIP's co-author Jong Wook Kim did suggest using the LAION-400M dataset during e-mail correspondence as the data distribution should be similar to what CLIP was trained on.

Since the caption distribution for CLIP was unknown, we decided to use the LAION-400M Dataset[4] captions as a proxy for CLIP's training distribution. LAION-400M is an open image-text-pair dataset with 400 million samples. LAION gathered the data in a similar manner to CLIP, and filtered the dataset by using CLIP itself. The filtering was done by using CLIP's vector encoding of the images and texts to drop any image-text-pair with a cosine similarity lower than 0.3, thus we assume the distributions are similar.

The caption data was stored in 400 parquet files, each with an individual download link. We downloaded them using a script which scraped all URLs and then individually downloaded the files. The dataset was then converted into a single 25GB text file with each line containing an image caption. Counts were calculated for each experiment by using CLIP's tokenizer to tokenize the label (e.g. the region name) and counting the captions where the label tokens were a subset of the caption's tokens.

LAION-400M (and CLIP's WebImageText dataset) have recently faced criticism. Birhane et al. (2021) show that LAION contains troublesome and explicit images and text pairs of rape, pornography, malign stereotypes, racist and ethnic slurs, and other extremely problematic content, and they outline several concerns and downstream harms of using such a dataset. It is indeed true that LAION-400M contains many of the harmful racist and sexist biases shown by humans on the internet.

[3] https://www.mapillary.com/dataset/depth

[4] https://laion.ai/laion-400-open-dataset/

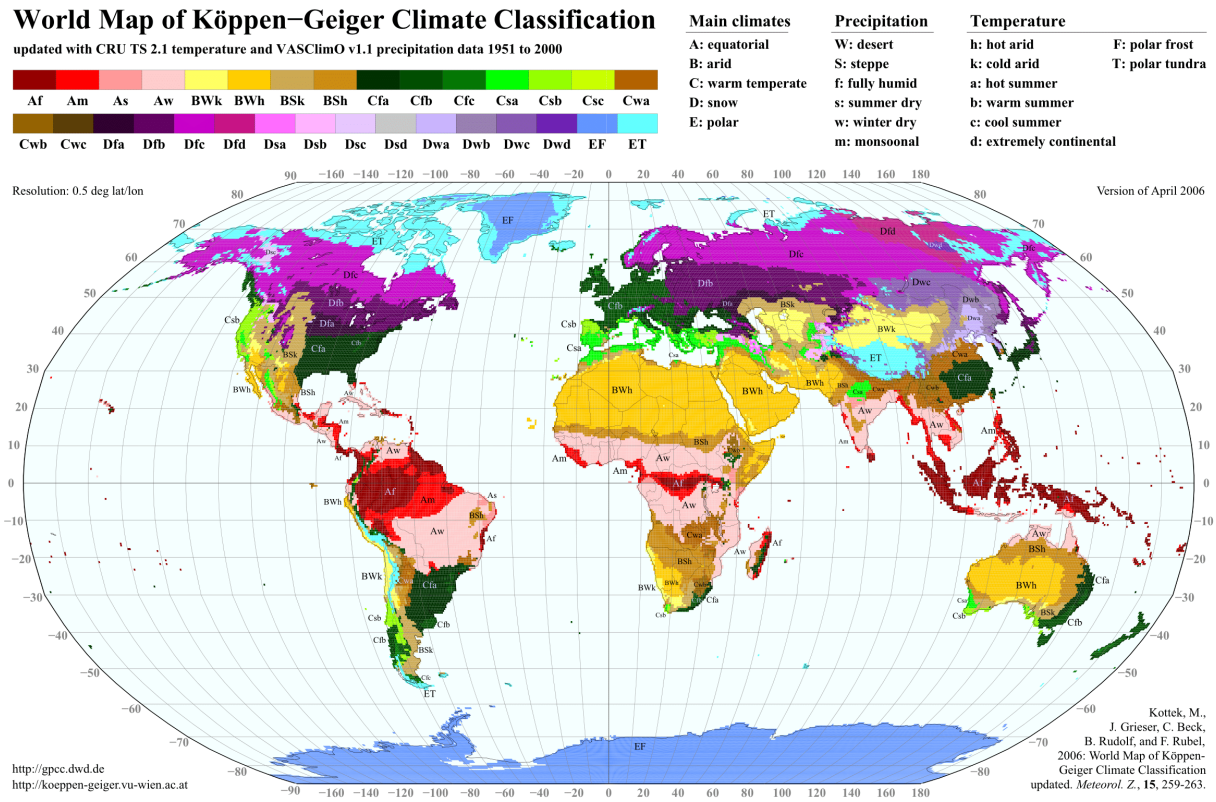

Figure 3: The Köppen-Geiger world map of climate classification visualised

However, we are only using it as a proxy to roughly understand CLIP's pre-training distribution. Since we are only using CLIP to test its capabilities to play Geoguessr, which doesn't contain problematic images, we find it unlikely that using either CLIP or the LAION-400M dataset will prove harmful.

2.4 Geopy

Since Mapillary only provides coordinates for each image, we needed a way to get geographical information from the coordinate to construct the labels. We started by using the Google Maps API[5], which returned some very detailed information, but their pricing scheme was cost prohibitive. We eventually found Geopy[6], a Python library that provided an easy way to query OpenStreetMap's Nominatim API[7]. Through Geopy we got varying metadata on each image, depending on the information that the OpenStreetMap database contained. Each coordinate returned information such as country, state, city and street name, among others. If the image was not located in a city or some other information was not available that image was not included in the relevant experiment. A major limitation with using the Geopy library was the rate limit of 1 request per second. Geocoding the entire Mapillary dataset by running a script on the HPC cluster took roughly a week.

2.5 Shapefile data

We wanted to explore the model's ability to detect a photograph's climate information as seen in Figure 3, and see if we could combine it with other predictions to improve accuracy. For this, we used the updated World Map of Köppen-Geiger Climate Classification (Kottek et al., 2006). They provide a shapefile containing geographic polygons of the different climates across the Earth down to a resolution of $\frac{1}{12}$ of a degree. To assign a climate to each image, we used GeoPandas[8], a Python module that allows spatial operations on geometric types. We loaded the images' coordinates and climate shapefiles into two GeoPandas dataframes and then joined them, so that each coordinate would be joined to its corresponding climate polygon. Figure 3 gives an overview of the different classifications.

We also joined the climate shapefile with US state and country shapefiles to get the percentage of each climate in each individual US state and country.

[5] https://developers.google.com/maps
[6] https://geopy.readthedocs.io/
[7] https://nominatim.org/release-docs/latest/
[8] https://geopandas.org/en/stable/

2.6 Region Data

For our world region experiments we needed to know which countries are in which regions. We decided to manually create this dataset by looking up world region names and including the ones we thought CLIP might know. This of course means that the inclusion of various regions is biased by our own non-expert understanding of world regions. Countries could be in multiple different regions so we leaned towards being lenient in which countries were included in which regions. An example would be that Morocco and Algeria are sometimes considered parts of the Middle East and at other times aren't depending on the source, so we included them in the Middle East region in case CLIP thought they were in the Middle East. Since we are not geography experts some mistakes were made, for example the United Kingdom was erroneously not included in the Northern Europe region. Since we didn't know if being too specific in location could be a detriment, we also included continental regions, such as Africa and Europe, that encompassed smaller regions.

3 Theory

3.1 GeoGuessr

GeoGuessr is an online game where the player is shown a Google Street View and has to place a pin on a map as close as possible to the location of the street view. Scoring typically takes into account the distance from the location in kilometres but alternative game modes where the country, state or city has to be guessed also exist. Due to CLIP only being able to predict the best matching caption of an image the latter game mode was simulated with the Mapillary dataset.

3.2 CLIP

CLIP is a neural network which efficiently learns visual concepts from natural language supervision. Unlike standard vision models, which are usually good at one particular task, CLIP is not optimised for any distinct task, and therefore has good performance across a wide variety of tasks and domains. The best performing CLIP model uses the modern Vision Transformer (ViT) architecture (Dosovitskiy et al., 2020), which divides an image into fixed-size patches, embeds each of them, and includes a positional embedding as an input to the transformer encoder. The self-attention layer in ViT makes it possible to embed information globally across the overall image, which the previous best performing models for image recognition (Convolutional Neural Networks or CNNs) are unable to do. This makes Transformers more robust and efficient (Paul and Chen, 2021). CLIP was designed to mitigate a number of major problems in previous deep learning approaches to computer vision:

- Costly datasets: The ImageNet dataset (Deng et al., 2009), one of the largest efforts in this space, required over 25,000 workers to annotate 14 million images. In contrast, CLIP learns from text–image pairs that are already publicly available on the internet.

- Narrowness: An ImageNet model is good at predicting the 1000 ImageNet categories without fine-tuning with a new dataset. In contrast, CLIP can perform a wide variety of visual classification tasks without needing additional training examples. The only thing needed is providing the possible classes for prediction.

- Poor real-world performance: Deep learning systems tend to suffer in performance when deployed in the wild. This gap between "benchmark performance" and "real performance" is avoided with CLIP, since it can be evaluated on benchmarks without having to train on their data. This results in CLIP's benchmark performance being more representative of its real performance. To test this, a linear classifier was fitted on top of CLIP's features, improving CLIP's accuracy on the ImageNet test set by almost 10%. However, this classifier does no better across an evaluation suite of 7 other datasets measuring "robust" performance (Taori et al., 2020).

CLIP uses a bag-of-words encoding with contrastive representation learning. This pre-training method was chosen for efficiency, as the bag-of-words encodings with contrastive objectives model learned 12 times more efficiently than a transformer language model with a predictive objective.

Given a batch of N (image, text) pairs, CLIP is trained to predict which of the N × N possible (image, text) pairings across a batch actually occurred. To do this, CLIP learns a multi-modal (i.e. image and text) embedding space by jointly training an image encoder and text encoder to maximise the cosine similarity of the image and text embeddings of the N real pairs in the batch while minimising the

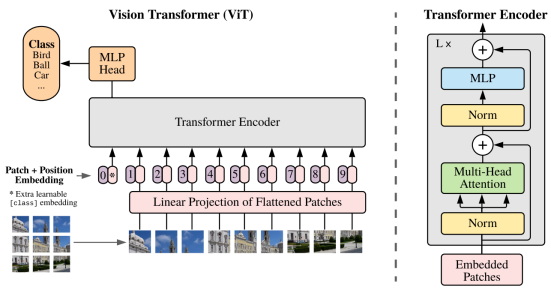

Figure 4: Vision Transformer overview

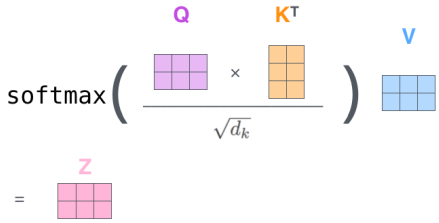

Figure 5: Self-attention head internal structure. d_k denotes the length of the key matrix, division by d_k in each self-attention head leads to more stable gradients.

cosine similarity of the embeddings of the $N^2 - N$ incorrect pairings. It then optimises a symmetric cross entropy loss over these similarity scores, as described by Recht et al. (2019).

CLIP's text encoder is a Transformer (Paul and Chen, 2021) with the architecture popularised by Radford et al. (2019). The transformer operates on a lower-cased byte pair encoding (BPE) representation of the text (Sennrich et al., 2016) with a 49152 vocabulary size. The activations of the highest layer of the transformer are treated as the feature representation of the text which is layer normalised and then linearly projected into the multi-modal embedding space.

CLIP experimented with different image encoder architectures, but the architecture that yielded best results (and the one we use in our project) is the Vision Transformer. ViT splits an image into fixed-size patches, linearly embeds each of them, adds position embeddings, and feeds the resulting sequence of vectors to a standard Transformer encoder, as seen in Figure 4. The encoding component used in our project (ViT-L/14) consists of several layers of encoders (24 for the Vision Transformer and 12 for the Text Transformer). Each individual encoder consists of two main parts: a self-attention layer and a feed-forward neural network (FFNN).

The self-attention layer consists of several attention heads (16 for the Vision Transformer and 12 for the Text Transformer). Each attention head multiplies each input (an image patch or word embedding for the first encoder, otherwise the output of previous encoders) against a query, key and value matrix. These matrices' values are learned during training of the Transformer. These new query, key and value matrices Q, K, V are then multiplied as shown in Figure 5. The query vector q_1 (e.g. the first word in the sentence) is multiplied against each key vector k_n. This represents how important each vector in K is to q_1. The results are then softmaxed and multiplied by value vector v_1 to get vector z_1. This is done simultaneously for all inputs in Figure 5. All attention heads get concatenated and multiplied with a weight matrix that was trained jointly with the model. The resulting matrix is the output that gets sent to the FFNN. Each sub-layer (self-attention, FFNN) in each encoder also has a residual connection (He et al., 2016) followed by a layer-normalisation step (Ba et al., 2016).

Unlike Figure 4, CLIP foregoes the last softmax layer used for classification, and instead calculates the cosine similarity of the vectors that come out of the last Multi Layer Perceptron Head (i.e. fully connected FFNN) of both the Text and Vision Transformer.

CLIP uses zero-shot inference for its predictions, which means that it can classify images with labels it has never seen before. This is because CLIP was trained with unstructured text data and learned to extrapolate which phrases and words correspond to which patterns in the images without needing specific labels.

3.3 GeoEstimation

The baseline model used to compare CLIP's results was the GeoEstimation model (Müller-Budack et al., 2018), the top performing model in the image to GPS benchmark[9]. GeoEstimation treats geolocalization as a classification problem by subdividing earth into non-overlapping geographical cells and predicting which cell the image belongs to. It uses multiple partitions to make predictions at different spatial resolutions. The geographical cells are created in a hierarchy by adaptively subdividing cells using thresholds of image counts in cells. This hierarchy of cells allows for predictions of the finest subdivision to incorporate knowledge of coarser representations by multiplying the respective probabilities at each level of the hierarchy.

[9] https://paperswithcode.com/sota/photo-geolocation-estimation-on-im2gps3k

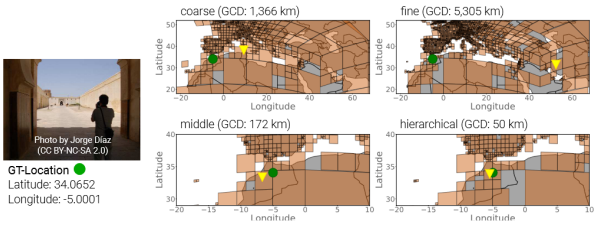

Figure 6: GeoEstimation cells with different hierarchical levels. The figure shows how using hierarchical information improves predictions.

GeoEstimation also uses a multi-task network by adding a pre-trained scene classifier model[10]. This scene classifier uses ResNet-152 (He et al., 2016) and is trained on Places365, a subset of the Places2 dataset (Zhou et al., 2017) which provides 365 different place categories. The hierarchical geolocation model and the scene classifier are joined by utilising a fully-connected global pooling layer with 365 output neurons (the amount of scene categories). The model then takes the cell with the highest probability and outputs the mean location of all training images in the predicted cell. GeoEstimation's accuracy can be seen in Table 1, where they define an accurate country guess if the predicted coordinate is within 750km of the image.

Distance to target				
Street 1km	City 25km	Region 200km	Country 750km	Continent 2,500km
10.5%	28%	36.6%	49.7%	66.0%

Table 1: GeoEstimation's best results on the Im2GPS3k benchmark

4 Experiments

Each experiment involved providing CLIP with a set of text prompts as labels to predict on. The standard format suggested in the CLIP paper (Radford et al., 2021) was used. This format consisted of "This is a photo of label, a category", where the label and category is inserted depending on the experiment. The MixedPrompt experiment differed slightly by customising the prompt further. When testing on these prompts, CLIP would evaluate each image-prompt pair and returned the prompt with the highest cosine similarity between the image embedding and the text embedding.

[10] https://github.com/CSAILVision/places365

4.1 Country100

The main experiment in our project is guessing the country of origin of an input image. To create a balanced dataset, we take the Mapillary images and randomly sample 100 for each country where at least 100 images exist. For the unfiltered Mapillary dataset this consists of 115 countries with 100 samples for a total test set of 11500 images. The labels in the form of prompts were formatted as strings e.g "This is a photo of Egypt, a country". Using the best performing CLIP model as described in Section 3.2 we achieved an F1 Score of 0.49 and an accuracy of 51%. The top 3 accuracy was 75% and top 5 was 85%, which is remarkable for a classifier with 115 labels that was not supervised to solve this particular task.

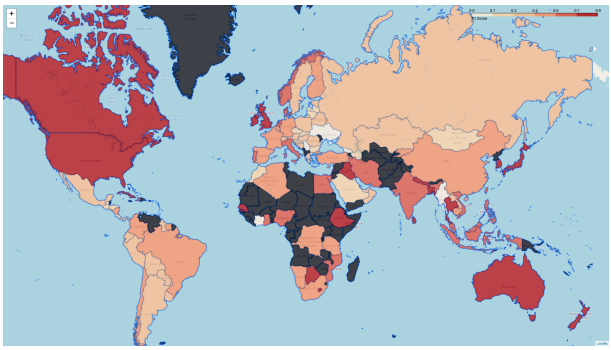

Figure 7: Choropleth of F1 scores for Country100. Black countries have no data. Small islands are not visible.

The choropleth in Figure 7 shows the distribution of F1 scores across the globe. Many countries in the dataset are not visible on this map due to their small size. English speaking countries, e.g U.S, Canada, Australia and U.K, all appear to have very high F1 scores. Due to CLIP being trained on English captions, a greater variety of pictures from these countries can probably be expected to exist in the training data. Pictures taken and captioned by locals in non-English speaking countries would largely be filtered out from CLIP's training data. This would likely lead to an external or tourist view of non-English countries which could account for part of the difference between English and non-English speaking countries.

Another explanation for the difference would be the amount of training data for each country. This was estimated using the LAION dataset label counts mentioned in Section 2.3. The scatter plot in Figure 8 shows that the number of times a country is mentioned in the caption data is correlated with

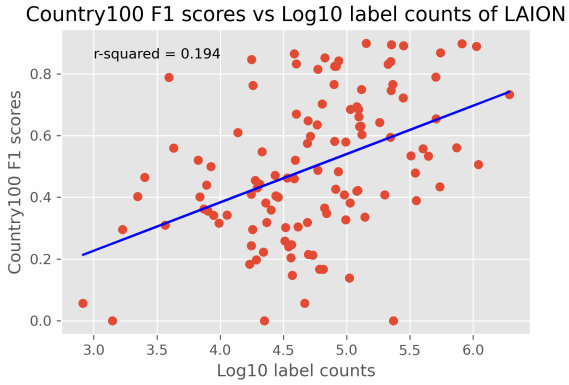

Figure 8: Plot of country-specific F1 scores against Log10 label counts for Country100. Label counts were calculated from LAION dataset

the F1 score. The correlation is not perfect, however, showing that there is more to the difficulty of guessing a particular country than the amount of training data. An example would be the point in the figure at (5.4,0) which corresponds to the country of Georgia's results. "Georgia" can cause quite a bit of ambiguity as it can refer to a person, the U.S. state or the country. Examining the errors for Georgia shows that for the 100 samples, 26 different countries were guessed. Furthermore, of the 3 images incorrectly predicted as being from Georgia only one of them was from the U.S. This suggests that CLIP had difficulty with the prompt and its internal Language Model probably has difficulty recognising "Georgia" as being a country entity.

A further in depth look at particular countries can be seen in Figure 9. In this figure it is clear that some of the least mentioned countries have a much lower accuracy. Similar to the example of Georgia this can be explained by the ambiguity in the country names. Both North Macedonia and Czechia have changed their names recently (2019 and 2016 respectively). This name change is the likely cause of the low count in the LAION dataset. These countries are likely referenced by multiple different names which is probably what makes it difficult for CLIP to understand the entity. This confusion matrix also highlights that even when CLIP predicts the wrong country it often predicts a nearby country instead. Democratic Republic of Congo is a long name which is probably part of the difficulty for CLIP but as can be seen in the confusion matrix it incorrectly predicts Comoros, Ethiopia and Côte d'Ivoire 42 times instead, all of which are also countries in Africa. The same holds true for Qatar and China, where Qatar predicts Bahrain and UAE 84 times and China guesses Singapore and Taiwan 48 times. These countries are either direct neighbours or very similar looking and would likely confuse even a human expert.

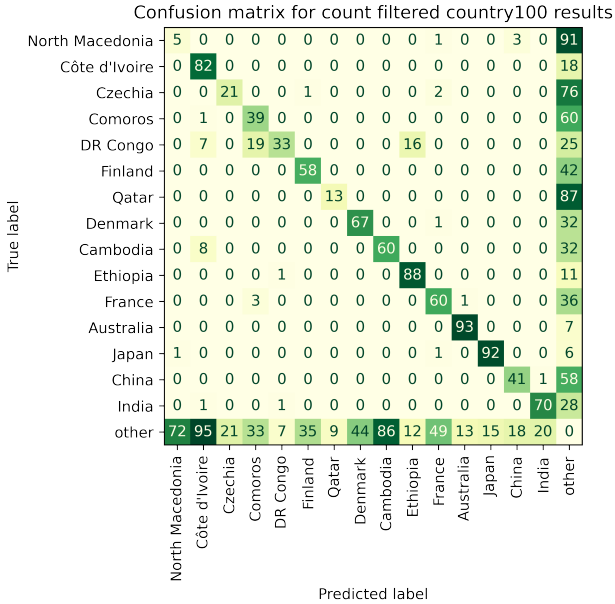

Figure 9: Confusion matrix for country100 results. Countries were selected by taking the 5 least mentioned, 5 median mentioned, and the 5 most mentioned countries in the LAION dataset. The countries are in order from least to most mentioned. Note that Democratic Republic of the Congo has been shortened to DR Congo.

4.1.1 Filtered Country100

Since the Mapillary data consisted of image clusters, we discovered that many of the images were nearly identical, as they were taken only a few meters apart. We tried different methods of data filtering, such as image hashing and feature matching. In our qualitative analysis, the best performing method for detecting very similar images was however cosine similarity of the image embeddings calculated using the Img2Vec python module[11].

We filtered the images using address data and cosine similarity. Images were grouped by road and only one image was included per road unless the cosine similarity of the image vectors was at most 0.9 (this threshold was determined through human evaluation). The assumption was that travelling through a city or facing a different direction would be different enough to not be duplicate data. A further distance check was made where if two images were on the same road but at least 1 km apart, then it was assumed that the location was

[11] https://github.com/christiansafka/img2vec

far enough away that despite high cosine similarity the images could still be considered different. Where road data was not available the same check was made by grouping the images by country and region/states in those countries.

The filtering was tested by applying it to the Country100 experiment. We took 100 randomly sampled images from each of the 97 countries that remained after filtering. The resulting accuracy and F1 score was 0.55 and 0.54 respectively. The increase in accuracy can likely be attributed to the lower number of classes in the test set. Due to the filtering only slightly increasing accuracy but reducing test size and complexity, it was decided not to filter the rest of the test sets.

4.2 Region

Since there was room for improvement of CLIP's country guessing, we wanted to see if this could be achieved by providing CLIP with additional information about the target country. The idea was to provide CLIP with information of the world region that the image belongs to. The information for which country is in which region was taken from our own dataset as described in section 2.6.

We devised three experiments:

1. We checked if CLIP could predict regions more accurately than countries.
2. We included the region of the world the country belongs to in the prompt.
3. We guessed the region and used that prediction to limit the search space of countries. We then guessed which country in that region the image came from (the idea being, if the region guessing was more accurate, then we could eliminate countries that CLIP might incorrectly predict).

4.2.1 Region using Country100

Since each country could be part of multiple regions, we decided that it was only necessary to pick one of the correct regions. This would also provide us with some information about which regions CLIP preferred to pick. Since we did not end up using this preference to eliminate regions, we have not included it as a result. We used the Country100 input data and provided CLIP with 31 regions to guess from. The accuracy was 70%. This is significantly higher than guessing the country so we assumed that further improvements were possible and continued with the Mixed Prompt and Country from Region experiments.

4.2.2 Mixed Prompt

Part of the goal of this project was an attempt to provide the model with more information to see if it was possible to increase the accuracy of the results. One of these experiments involved prompt engineering. Since the labels that CLIP predicts on are strings it is possible to give the model full sentences, which is shown in the CLIP paper to increase accuracy. For the basic Country100 experiment, we simply followed the suggested format of "This is a photo of" + country name + ", a country". This experiment, on the other hand, tried giving CLIP region information as well as country. Since different regions overlap, countries can often be parts of multiple regions. As an example, Bulgaria is considered to be part of the Balkans but is also part of eastern Europe, so Bulgaria had two different prompts: "This is a photo of Bulgaria, a European country in the Balkans" and "This is a photo of Bulgaria, an Eastern European country." The number of prompts a country was in was not equal so there was significant bias in this experiment from that.

There were 321 prompts in the final test set for this experiment, in comparison to the original 115 for Country100. This experiment ended up with an accuracy of 45% and an F1 score of 0.32. This is noticeably lower than the Country100 results. Therefore, we concluded that adding an excessive amount of information to the prompt will simply confuse CLIP and decrease the accuracy. There should only be enough information added to the prompt to help distinguish between different types of entities but not too much to avoid that additional information overwhelming the actual target label.

4.2.3 Country from Region

In this experiment we attempted to limit CLIP's search space by first predicting the region and then only letting CLIP guess between countries in that region. Due to the accuracy of guessing the region being higher than guessing countries directly, we assumed this experiment might improve country predictions results.

Unfortunately this was not the case as we only got an F1 score of 0.28 and an accuracy of 39%. Due to certain regions only having 2 countries, we do not calculate top 3 or top 5 accuracy. This result confirmed that although CLIP can guess regions better than countries, providing it with that extra information will only introduce the possibility for more errors.

4.3 States

The main goal of the project was to predict countries, but we also attempted to distinguish a finer spatial level. The metadata from Geopy (2.4 provided a full address for each image, which included administrative level information. The United States and France were selected for this experiment since both had a large enough presence in the Mapillary dataset to provide almost full coverage of their states/regions. United States has 116,087 images in Mapillary and France has 22,951. The prompts were of the format "This is a photo of Oregon, a state" and "This is a photo of Mayotte, a French state." The French prompt erroneously used "state" instead of the proper term "region" which was not caught in time to rerun the experiment.

U.S State	Image count
New Hampshire	0
Oklahoma	73
Vermont	7
Nebraska	68
Maine	92

Table 2: U.S States that did not have the desired 100 images but the images that existed were still included

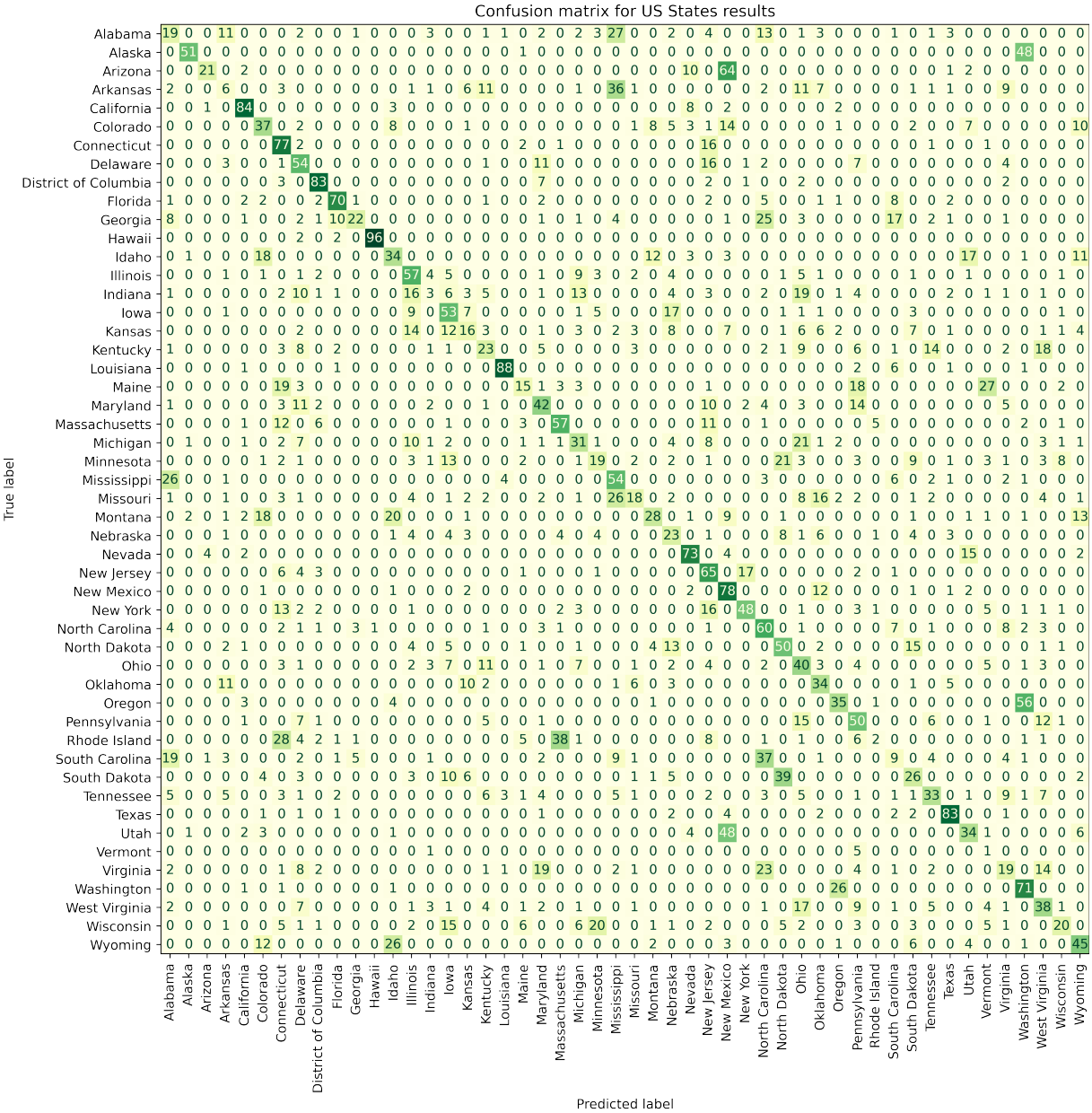

Figure 10: U.S. States confusion matrix

4.3.1 U.S States

For each state, at most 100 pictures were added to the test set. A few states did not meet the required number of pictures but were added regardless. District of Columbia was also included despite not being a state. The states without 100 images are listed in Table 2.

The final result was an accuracy of 43% and an F1 score of 0.41. These results are very good if not quite as good as the Country100 experiment, but this is to be expected. Guessing between States should be more difficult than between countries, as there is not as much visual distinction within a single country. The top 3 result was an accuracy of 71%. This top 3 is also reflected in the confusion matrix in Figure 10 as looking at individual states will show that CLIP often predicts a neighbouring state instead. For example, Arizona is only correctly predicted 21 times, but predicts the neighbouring states of Nevada and New Mexico 10 and 64 times respectively.

4.3.2 French Regions

Every French region had at least 100 images apart from Corsica which had 78. The accuracy for this experiment was 28% and the F1 score was 0.26, whereas the top 3 accuracy was 58%. This is significantly worse than the U.S states experiment despite France having 18 regions in comparison

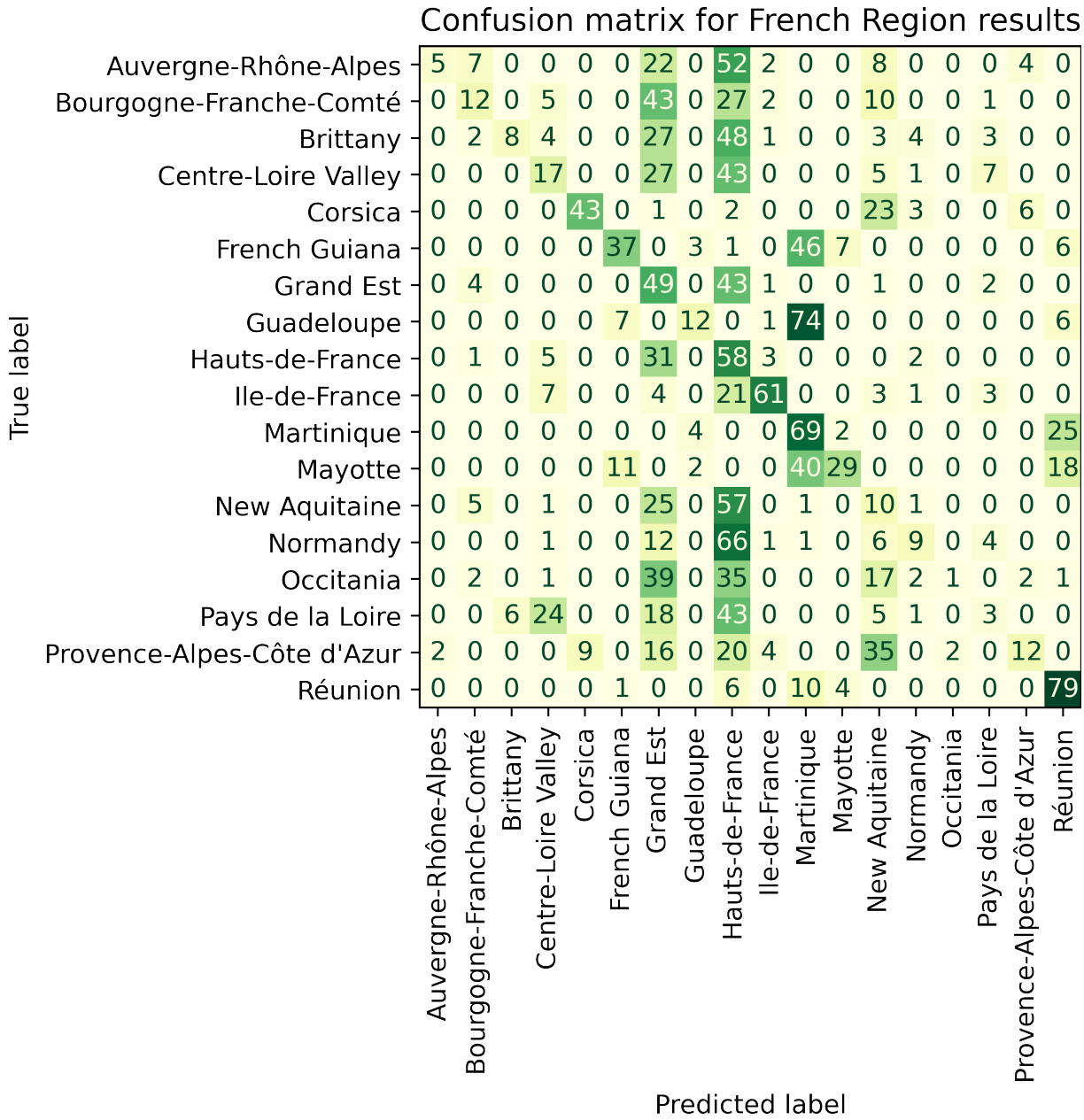

Figure 11: French Regions confusion matrix. Class size was 100 (apart from Corsica, which only had 78)

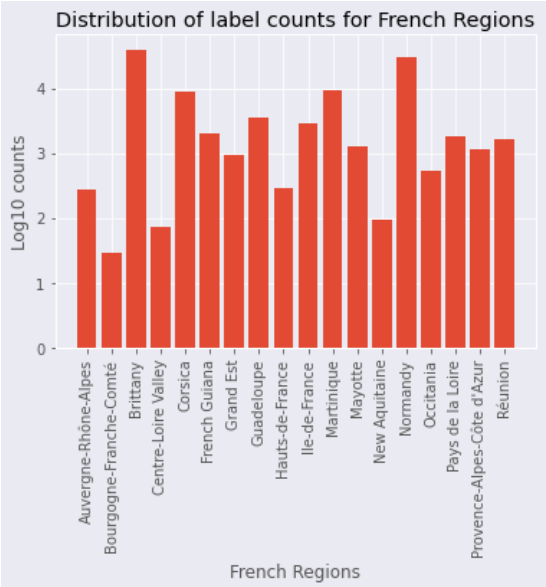

Figure 12: Log10 count of French region tokens in LAION dataset

to the 50 U.S states. This is likely in part due to the language difference with CLIP being primarily trained on English captions. This might mean that the images of these regions are not taken by a local, or if they are, the caption might not be written by a local. Therefore, a certain level of bias could be involved in which pictures are uploaded to the specific English part of the internet in which the photo-caption pair was collected.

The region names are also much less commonly found in the LAION dataset, with the highest counts for French regions being "Normandy" with 30181 and "Brittany" with 39206 as can be seen in Figure 12. In comparison, the two least commonly named U.S states was "District of Columbia" with 6520 and "North Dakota" with 26069 and in the former's case this is because it is colloquially referred to as "D.C" or "Washington D.C". The term "France" is mentioned 670355 times so there are plenty of pictures from France. However, they might simply not reference the specific region as often. This could be due to cultural differences where the U.S is a federation and citizens often identify themselves as coming from a specific state. In contrast, in France it could be much more common to refer to the city or town one is from. Geographic differences between French regions in mainland Europe are also much smaller than between the U.S states and could be another reason for the discrepancy between U.S and French results.

The confusion matrix for French regions in Figure 11 shows the difficulty CLIP has in predicting French regions. "Grand Est" and "Hauts-de-France" are (incorrectly) used as predictions significantly more than the others and it is unclear why, as both are relatively rarely mentioned in the LAION dataset. The only guess we have as to why would be that certain individual tokens in the region names are singled out as being important by CLIP. Another interesting note is that CLIP almost exclusively confused the French islands of "Réunion", "Mayotte", "Guadeloupe" and "Martinique" with each other. This indicates that despite these islands having relatively few references in the LAION dataset, with all being in the 1000s, CLIP was still able to successfully understand that these names refer to islands even if distinguishing between them accurately was not possible.

4.4 City300

There were 261 cities that contained at least 300 images in the Mapillary dataset. Any image address from the OpenStreetMap metadata gathered in Section (2.4) that contained a city name in the address was considered to be in a city of that name. This does not guarantee that the place is actually a city as Geopy also encodes some neighbourhoods as cities. No cleaning was done on the city names or to remove non-cities. The lack of cleaning means that 26 of the city names occurred less than 10 times in the LAION dataset and of those 19 of them occurred 0 times. Part of the reason for this was that names containing Arabic text such as "arrondissement de Menara *arabic*" (a city quarter in the city of Marrakesh, Morocco) were included. Although CLIP can tokenize the Arabic, it has not been trained on many Arabic captions and therefore most likely can not interpret those tokens meaningfully. Despite these messy labels, CLIP still managed to achieve an accuracy of 30% with an F1 score of 0.29. The top 3 accuracy was 50% and the top 5 was 61%. Although this result is not quite as good as some of the other experiments it is by the far the most difficult experiment for CLIP so far with 261 unique labels.

In order to compare GeoEstimation, we checked the distance between the predicted coordinate and the actual coordinate. GeoEstimation was able to get within 100km 16% of the time and within 25km 11% of the time. The comparison is not completely fair as GeoEstimation's prediction space is the entire globe whereas CLIP's is only the provided labels. That said, CLIP's results could also be im-

proved with extra cleaning of the city names. Since the goal of this project was to test CLIP's ability to play Geoguessr, being able to easily limit the search space is a benefit since the data distribution for Geoguessr is known and factors into expert players' guesses. Therefore, the higher accuracy of CLIP and ease of limiting the search space make it better suited than GeoEstimation at guessing cities in Geoguessr.

The confusion matrix in figure 13 shows some surprising results. Despite the 5 first classes not having their entire name in any one caption in the LAION dataset, CLIP was still able to correctly predict some of the images. "Baviaans Local Municipality", a now defunct municipality in South Africa, has surprisingly high recall of 0.77 although the precision is only 0.22. This would indicate that despite the entire label's tokens not being in any one caption, the "Baviaans" tokens were probably unique enough for CLIP to learn the concept of this location from however many images mentioned "Baviaans". A better cleaning of the city names would have given a better indication of the actual number of images mentioning this location in its caption.

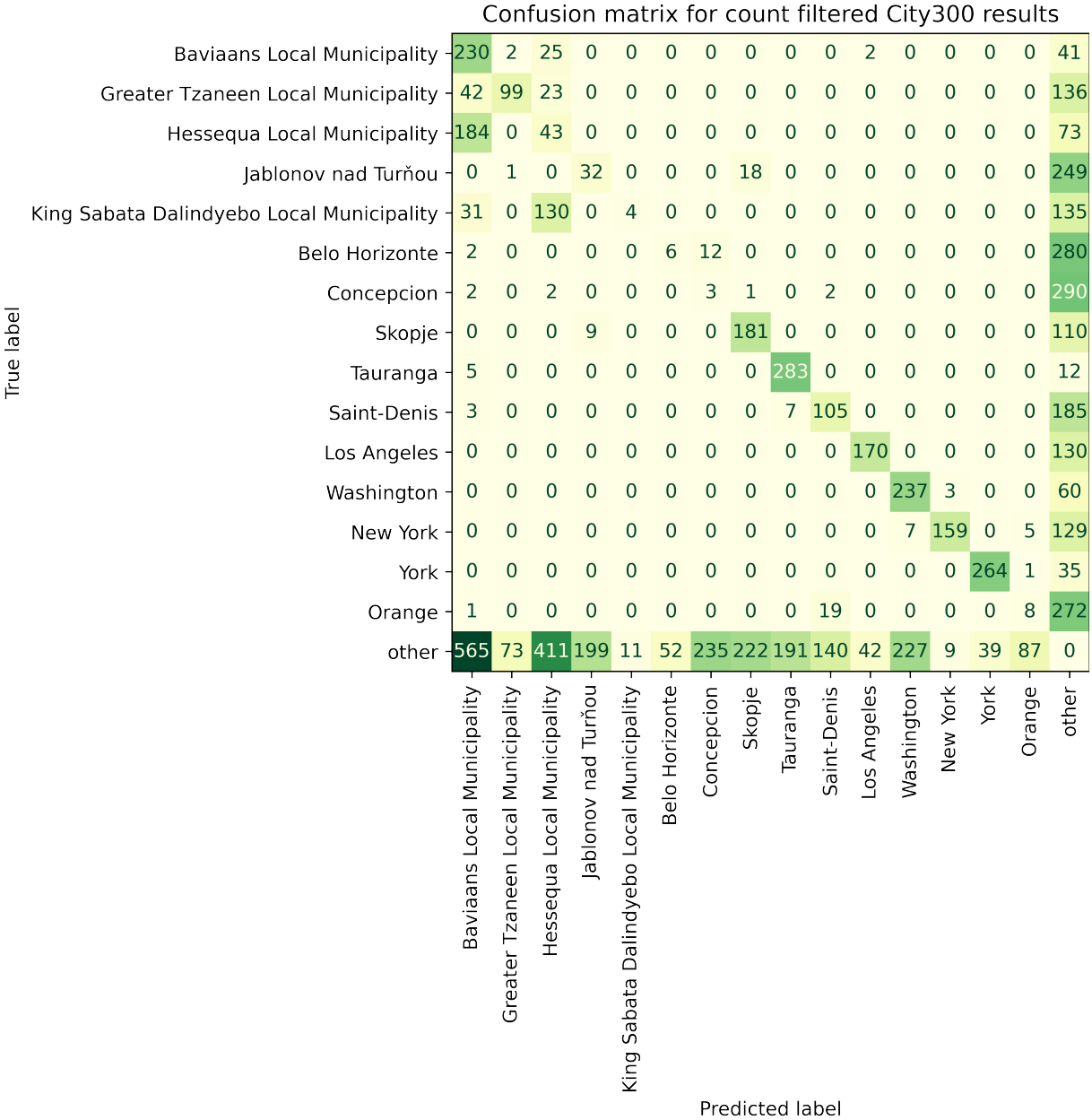

Figure 13: Partial confusion matrix of City300 results for 5 lowest, median and highest counts in LAION dataset. Classes are in order of counts by lowest to highest. Class size was 300 for all 261 classes and other classes that predicted one of these 15 or were predicted by the 15 are in the "other" label.

Another interesting result from this confusion matrix is that there was no confusion between "New York" and "York". The two locations look very different and are in two different countries so it's not surprise that the image encoder would be able to separate the two. However, it's quite surprising that the text encoder is able to separate the two enough to cause no confusion despite the only difference between the two locations being the single presumably common token for "New".

The result for "Orange" in the confusion matrix is also worth looking at. The location is ambiguous, so after checking which location this label was referring to, we discovered that we'd made a mistake in this experiment. Part of our assumption was that city names would be unique, and if not, they would likely refer to the location most commonly understood (i.e. "Paris" would not be confused with "Paris, Texas"). However, this does not hold up for "Orange", as there are 177 images from Orange, Vaucluse in France and 123 images from Orange County in California. This would indicate that despite the mixed prompt experiment in Section 4.2.2 not providing good results for country labels, it might be necessary for city labels. Regardless, we would still expect some recognition of either of these cities if this was the only source of confusion for CLIP. The fact that we don't is likely because "Orange" also refers to a colour and a fruit, so CLIP probably has too many different concepts tied to this single token. A further experiment in prompt engineering to differentiate the two cities would need to be done to ascertain if prompt engineering for cities is necessary.

4.5 Climate

The best performing humans in picture geoestimation make use of a variety of contextual clues to guide their guesses. One such clue might be the climate information encoded in the image (e.g. a picture of a polar, snowy climate will not be confused with a tropical country). Our idea was to get CLIP to guess the climate of a picture and use that to adjust the probabilities according to the climates present in each region (e.g. state or country).

The first step was to gather world climate data, for which the most reputable source is the World Map of the Köppen-Geiger climate classification updated (Kottek et al., 2006). This data came in the form of a shapefile with polygons for each climate. After loading it into a GeoPandas dataframe, we joined it with the coordinates for each image in our test set. We also gathered shapefiles of country borders and US state borders. By joining these shapefiles with the climate shapefile we got geodataframes of country and US state climates. Using the groupby function, we were able to calculate the percentage of each climate category for each of these administrative boundaries.

However, to test whether we could use CLIP's climate guesses to aid its geolocation, we first needed to see if the model had encoded climate information.

Since some of the climate labels are so specific, we found many of them had no appearances in the CLIP training data proxy (LAION-400m), so we did not expect great results for climate classification from the model. This is why we decided to re-label some to more commonplace climate descriptions (e.g. "cold with cold summer without dry season climate" became just "cold climate"). This reduced the decision space from 28 to 10 unique labels.

We ran two separate experiments on both our hand-made labels and the original labels by letting CLIP predict the climate for each picture in the Mapillary data-set and comparing it to the gold label for that coordinate taken from the shapefile. Results can be seen on Table 3.

Although the hand-crafted labels improved the metrics substantially, the experiments revealed that CLIP is not good enough. As can be seen in Figure 14, CLIP tends to guess temperate and tropical for most images, so adding the climate guess to the geographic prompts would not provide new high quality information to the model. Therefore, we decided not to go further with this experiment.

Label type	Accuracy	Macro-averaged F1
True labels	0.07	0.04
Hand-crafted	0.37	0.16

Table 3: Climate results

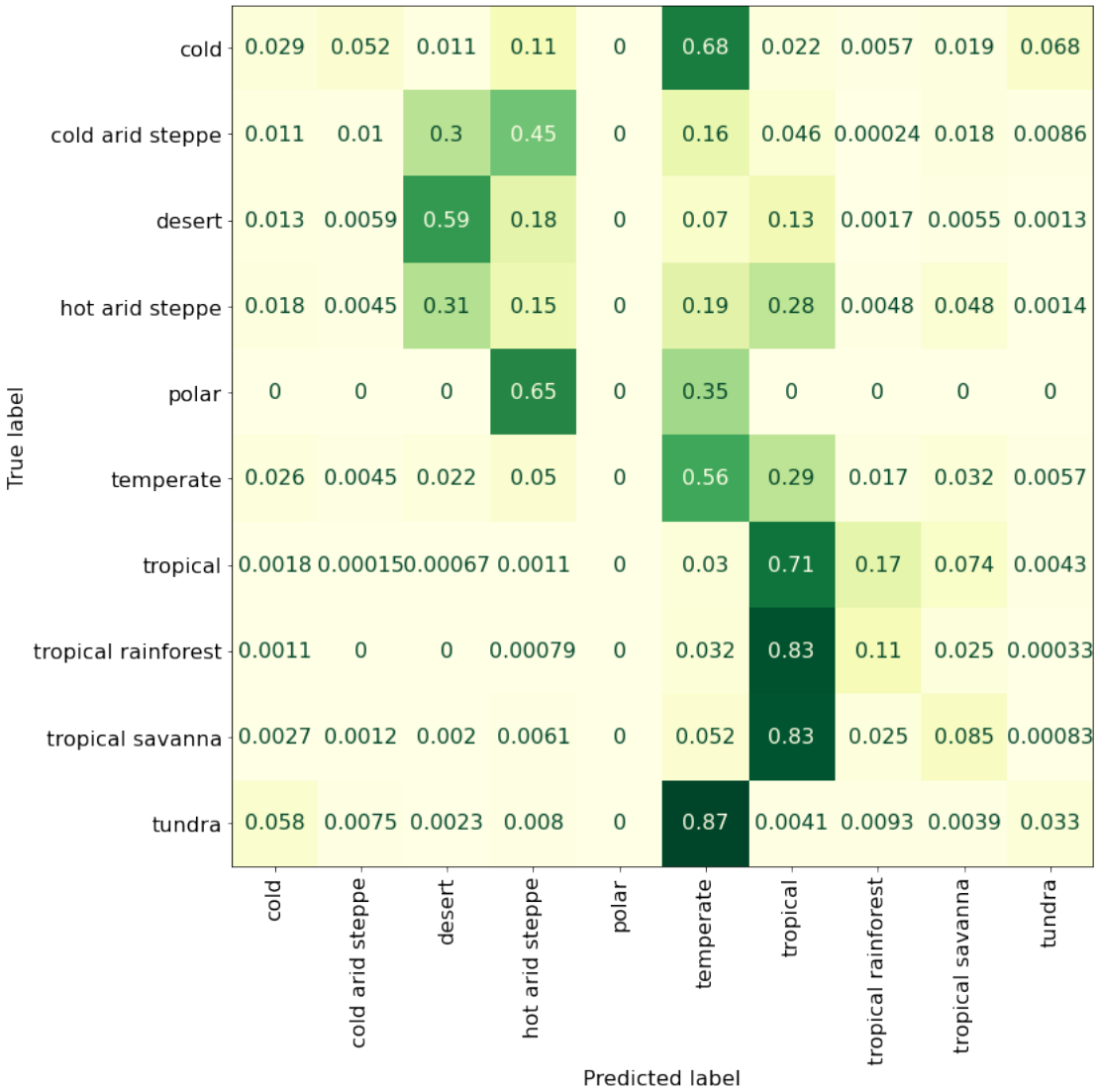

Figure 14: Climate prediction confusion matrix, normalised by true labels.

Experiment	Accuracy	Macro-averaged F1
Country100	0.51	0.49
Filtered Country100	0.55	0.54
MixedPrompt Country100	0.45	0.32
GeoEst for Country100	0.23	0.16
City300	0.3	0.29
GeoEst for City300 100km	0.16	N/A
GeoEst for City300 25km	0.11	N/A
U.S States	0.43	0.41
French Regions	0.28	0.26
Region	0.7	N/A
Country from Region	0.28	0.39
Koppen-Geiger Climate	0.07	0.04
Hand-crafted Climate	0.37	0.16

Table 4: Results for all experiments.

5 Discussion

We set out to investigate the capabilities and limitations of CLIP for Geolocation and gathered some insights around how to interact with a joint vision-language model. Each experiment we undertook to try to help CLIP reach a better conclusion did not produce successful results. We could not help it by pointing to climate distributions, and could not help it by using its understanding of regions and subsequently reducing its decision space. It seems as though CLIP has encoded these and many more aspects of image-caption pairs and is already exploiting them so any help that we attempt to provide simply confuses CLIP.

The difference between U.S states and French regions results suggests that CLIP's geo-location capabilities suffer from the English language filtering of the captions. This is also supported by the results in the Country100 experiment 4.1, where English speaking countries appear to have a higher F1 score than non-English speaking countries. The byte pair encoding of CLIP means it does not have issues tokenizing non-English text, but it does not ap-

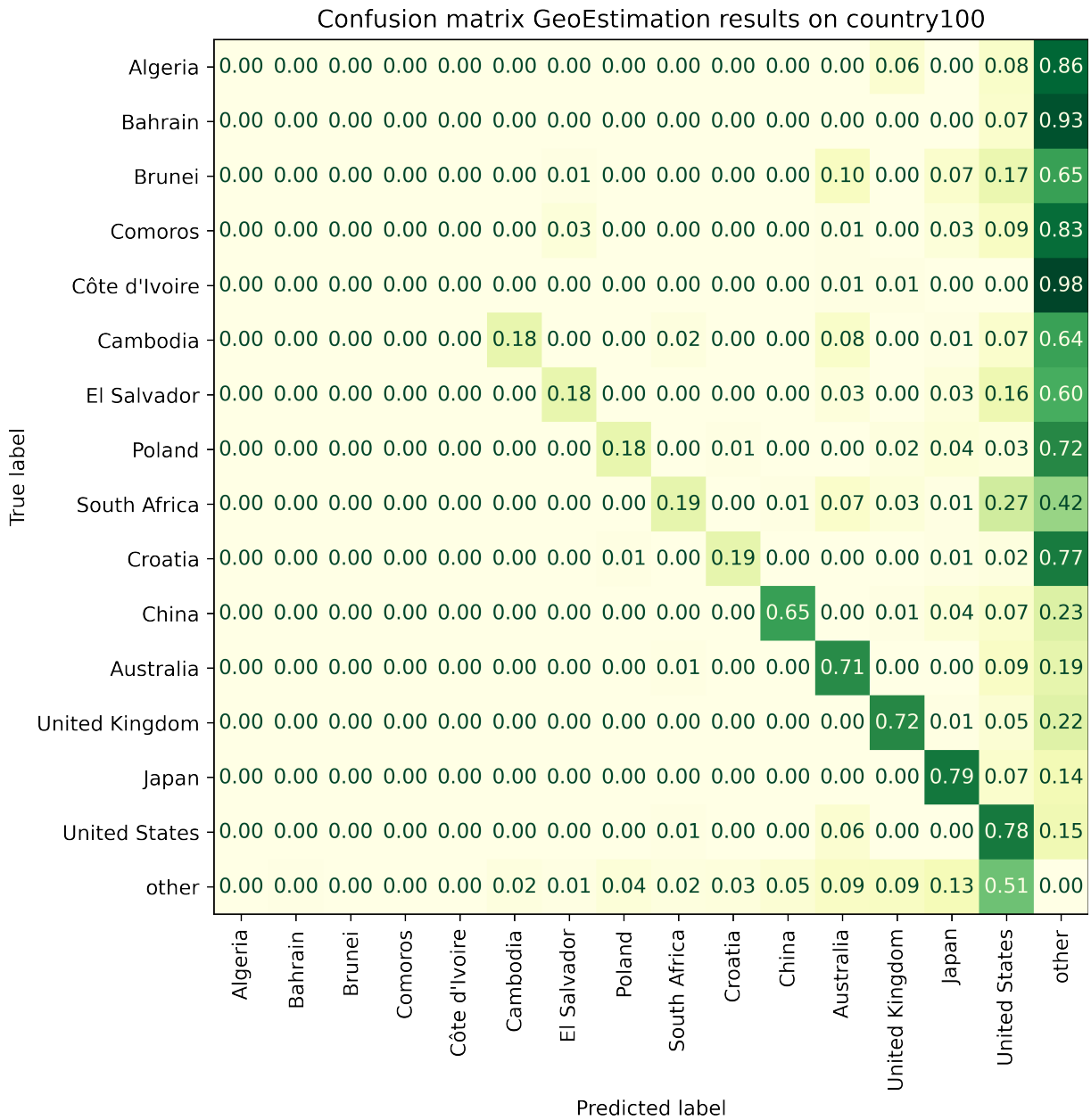

Figure 15: GeoEstimation results confusion matrix for bottom, median and top countries by accuracy. Normalised by true labels

pear to meaningfully interpret place names that are not the original English spelling. A multi-lingual approach to CLIP could substantially improve its geo-location accuracy. Alternatively, a more rigorous cleaning of labels needs to be done to avoid named entity disambiguation such as for "Orange" in the City300 experiment.

CLIP is much better at predicting specific named places than the GeoEstimation model we used to compare it to. GeoEstimation claims a country level accuracy of 49.7%, except this is simply a check of whether or not the predicted coordinate is within 750 km. This is inadequate for GeoGuessr when guessing specific place names, which can be seen when we test GeoEstimation on the Country100 dataset where it only gets an accuracy of 23% after converting the coordinate to an address using Geopy (as explained in Subsection 2.4). The results for a selection of countries can be seen in Figure 15. This confusion matrix shows that GeoEstimation has a tendency to incorrectly predict the United States instead of the true label, with 1081 incorrect predictions for United States, which account for 9% of the entire test set. Bahrain, Comoros and Brunei, which are small nations less than 100 km across, also get no correct predictions. The 750 km accuracy metric that GeoEstimation used would likely get more results for these

countries by predicting their neighbours. Another explanation for the discrepancy between GeoEstimation's claimed results and its performance on the Mapillary dataset could be because the Mapillary dataset is a more specific domain than what GeoEstimation was trained on. GeoEstimation was trained on outdoor and indoor images similar to Country211 (Subsection 2.1), so retraining GeoEstimation on Mapillary might improve its results substantially. The need for retraining is reinforced by the City300 experiment, which uses the distance metric of 25 km and 100 km from the original coordinate to compare GeoEstimation results. Despite the larger margin of error provided by the 100 km compared to their distance of 25km for cities, GeoEstimation still only gets an accuracy of 16% on Mapillary compared to its claimed performance of 28%. For the two different game modes in GeoGuessr, a retrained GeoEstimation would likely provide the best results for coordinate guessing, but CLIP is much better at guessing named locations. When compared to GeoEstimation, OpenAI's claim that CLIP's zero-shot capabilities can reach near state-of-the-art performance for some task specific models is well founded.

6 Limitations

CLIP uses Bag-Of-Words encodings because of a need for high training efficiency, which leads to a number of issues. For one, it loses all word order: for CLIP, "cold with hot summer without dry season" is the same as "hot without cold season with dry summer". Another issue is it does not understand contextual embeddings (e.g. the difference between Georgia as a name, state or country), which limits the model's ability to correctly disambiguate homographs. Unfortunately, there is no easy way to correct these problems without significantly increasing training time, as mentioned in 3.2.

CLIP is limited to choosing only from the labels provided, which is a significant restriction compared to a truly flexible approach like image captioning, which could generate novel outputs. This makes feature engineering crucial and puts significant responsibility on the human to choose a suitable set of labels. If the human misses something or words things incorrectly, the model may never return the desired output even though it has encoded the desired information.

Image-caption pairs are only included if they have a word occurring at least 100 times in the English version of Wikipedia. This makes the images included biased towards what an English speaking person would take pictures of and talk about, or the type of images that would get translated to English.

Our caption count does not necessarily show the true amount of pictures of each country, and the model may correlate certain words with certain countries without the specific country needed to be named in each picture. For example, the model may correlate pictures of bicycles with The Netherlands more than the US.

Our comparison between CLIP and GeoEstimation has one flaw: CLIP only needs to guess from the list of countries in Country100, but GeoEstimation can guess any cell on Earth. This makes it harder to compare them exactly. This may also be an advantage of CLIP, though. If, for example, you wanted to play GeoGuessr (which does not include every country), you could only give CLIP the possible countries to improve its accuracy. To do this with GeoEstimation, however, would be much more convoluted: you would have to find the cells that included the GeoGuessr countries (some of which overlap or partially include countries not present) and remove them from the neural network, messing up the weights in the process.

7 Future Work

Even though our experiments that attempted to improve CLIP's initial guess were not successful (sections 4.2.2 and 4.5), there still may be other ways to help it. One option would be letting the model predict left hand/right hand driving, reducing the possible labels to the correct countries or simply adjusting the probabilities for each country given the prediction. This was not attempted however, as there was already a high accuracy for most left-hand drive countries so we assumed that CLIP had likely already factored this into its predictions. An experiment to verify this would still be interesting.

Another idea would be to use the position of the Sun to determine the hemisphere. This would require knowing the direction the camera is facing, which is possible in GeoGuessr, but not possible with the Mapillary dataset. The idea is that if the Sun is to the North, then the photo is of the Southern Hemisphere, and if the Sun is to the South, the photo is of the Northern Hemisphere. This has some caveats if the photo is of the Equator or in-

side the tropics, but otherwise works quite well according to GeoGuessr enthusiasts[12].

One could also use biome information similar to our climate experiment by using, for example, the Terrestrial Ecoregions of the World by Olson et al. (2001). They provide a shapefile with biome information across the Earth. Since different biomes have different flora, this could theoretically help CLIP identify specific countries according to their plant life.

OpenAI has experimented with fitting a linear classifier on the representation extracted from the model and achieved good results across a variety of datasets. Since our main focus was to explore the zero-shot capabilities of CLIP to investigate the representations it has learned, we did not follow this route. Fitting a linear classifier on top of CLIP's features using the Mapillary dataset may be a promising route if one wishes to improve the accuracy of the model regarding Mapillary-like images.

As a final note, the mixed prompt experiment where we attempted some prompt engineering, did not improve results. However, we discovered a web tool [13] for exploring the LAION dataset at the very end of our project. This tool uses CLIP to query the LAION dataset and returns the images that best match a given prompt. This tool is invaluable for prompt engineering as it quickly provides feedback for how well a given prompt matches images similar to Mapillary. As an example, the prompt "A street view image of France, a country" returns images very similar to Mapillary where as our original prompt of "This is a photo of France, a country" primarily returns images of maps of France. Whether this would impact our results is unknown so further experiments would be required.

[12] https://www.reddit.com/r/northernlion/comments/uggqnq/sun_position_tips_for_geoguessr/

[13] https://knn5.laion.ai

8 Conclusion

In this paper, we have investigated the zero-shot geo-location capabilities of CLIP on the specific domain of street view imagery used in the browser game GeoGuessr. We have performed several experiments testing potential ways to help the model improve its predictions, by giving CLIP world region information as well as climate information in the prompts. Neither the region nor climate experiment improved the results in comparison to giving CLIP a simple prompt without too much extra information. In the future, we would recommend not over-engineering the prompts. Furthermore, one should be careful when constructing the prompts to avoid the polysemy of certain labels (e.g. Georgia or Orange) confusing the model.

We have also performed country, U.S. state, French region and city prediction experiments and present an analysis of the results. We have shown that CLIP performs better than the state of the art GeoEstimation model in country and city label guessing.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Liane Bernstein, Alexander Sludds, Ryan Hamerly, Vivienne Sze, Joel Emer, and Dirk Englund. 2021. Freely scalable and reconfigurable optical hardware for deep learning. *Scientific Reports*, 11(1):3144.

Abeba Birhane, Vinay Uday Prabhu, and Emmanuel Kahembwe. 2021. Multimodal datasets: misogyny, pornography, and malignant stereotypes. *arXiv preprint arXiv:2110.01963*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

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. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten Bosma, Zongwei Zhou, Tao Wang, Yu Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathy Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc V. Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2021. Glam: Efficient scaling of language models with mixture-of-experts. *CoRR*, abs/2112.06905.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Markus Kottek, Jürgen Grieser, Christoph Beck, Bruno Rudolf, and Franz Rubel. 2006. World map of the köppen-geiger climate classification updated. *Meteorologische Zeitschrift*, 15(3):259–263.

Eric Müller-Budack, Kader Pustu-Iren, and Ralph Ewerth. 2018. Geolocation estimation of photos using a hierarchical model and scene classification. In *Computer Vision – ECCV 2018*, pages 575–592, Cham. Springer International Publishing.

David M. Olson, Eric Dinerstein, Eric D. Wikramanayake, Neil D. Burgess, George V. N. Powell, Emma C. Underwood, Jennifer A. D'amico, Illanga Itoua, Holly E. Strand, John C. Morrison, Colby J. Loucks, Thomas F. Allnutt, Taylor H. Ricketts, Yumiko Kura, John F. Lamoreux, Wesley W. Wettengel, Prashant Hedao, and Kenneth R. Kassem. 2001. Terrestrial Ecoregions of the World: A New Map of Life on Earth: A new global map of terrestrial ecoregions provides an innovative tool for conserving biodiversity. *BioScience*, 51(11):933–938.

Sayak Paul and Pin-Yu Chen. 2021. Vision transformers are robust learners. *CoRR*, abs/2105.07581.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. 2019. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5389–5400. PMLR.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. 2020. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems*, volume 33, pages 18583–18599. Curran Associates, Inc.

Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. Yfcc100m: The new data in multimedia research. *Commun. ACM*, 59(2):64–73.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. 2017. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.