Comparison of Transformer and CNN based architectures on Image Geolocation Estimation

By: Eren Aydoslu, Cem Levi, Igor Witkowski

Course: Seminar Computer Vision by Deep Learning (CS4245)

Date: 16th of June 2024

Group: 12

Repository: https://github.com/erenaydoslu/computer-vision-project

---

Introduction

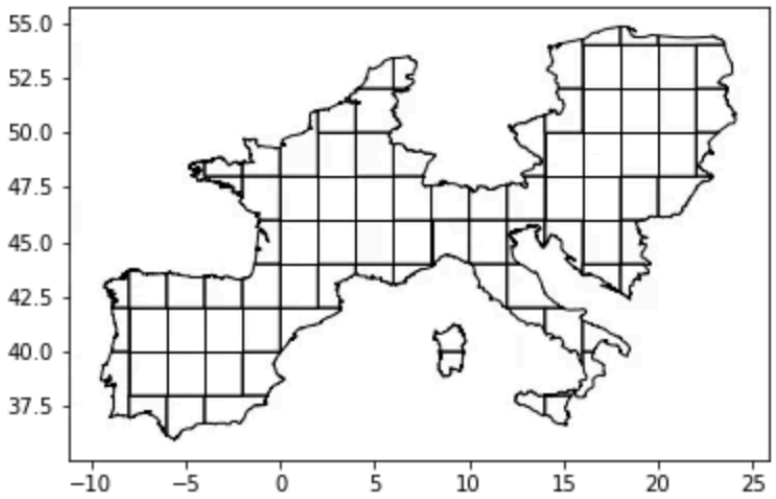
For this project, we were inspired by the game GeoGuessr. Geoguessr is a browser-based geography game that's played internationally and has risen quite a bit in popularity over the past few years. In the game, players are given a randomly selected Google Street View image and their goal is to predict the location of the image. The closer the player guesses to the actual location, the higher the score in the game will be.

The game is an atypical computer vision problem; instead of predicting what's in the image, we're trying to predict where the image is from. Even though atypical compared to computer vision tasks like image classification, people have tried to tackle the problem of estimating geographical location of an image. However, prior work focused on the use of CNN-based models to tackle the task of image geolocation estimation(Weyand et al., Vo et al.,Suresh et al.,). These papers showed that CNN-based models can accurately distinguish between images of different locations.

In this project we aim to tackle the task of image geolocation estimation with the use of visual transformer based architectures. Recently transformers became prominent as an alternative to convolutional neural networks for computer vision tasks like image recognition and computer vision. With the addition of pretraining on very large datasets and finetuning on target datasets,inspired by Natural Language Processing, Vision Transformers (ViT) was developed (Dosovitskiy et al.). ViTs can be competitive with convolutional neural networks; however, they are computationally more expensive, require more training data, making ViT without any clear benefits over convnets as of now. Our goal for this project is to test the ability of a transformer-based architecture on image geolocalization and compare it with that of CNN architectures. For this we ask the research question; How do transformer-based architectures perform on the task of image geolocalization compared to CNN-based architectures?

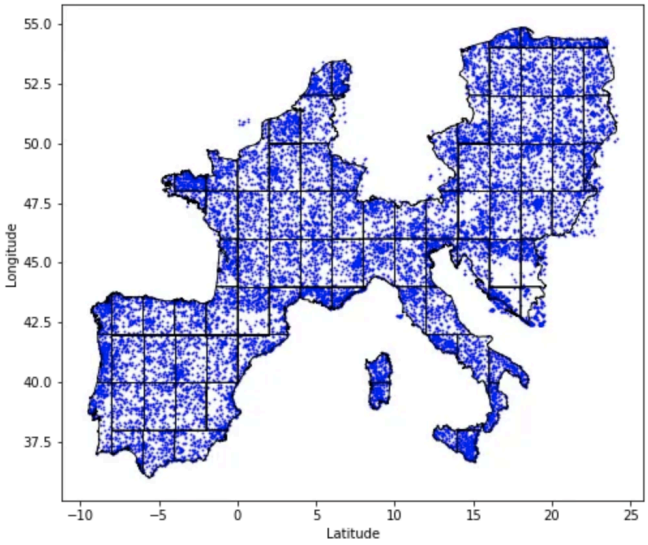
Data

We used the data collected by Geoguessr-AI Github repository (Salehalwer). The data was collected from 15 European countries excluding the Spanish Portuguese islands. The author of the repository merged the maps of these 15 European countries and split the resulting map into a grid. The resulting map can be seen in Figure below. Each square is a maximum of 12,000 km squared. Any squares that were smaller than 3000 km squared were merged with their neighboring squares.



The data is then collected by using Google Street View API to collect three images for 200 randomly generated coordinate pairs within each square on the grid. The images are

sampled at three random heading angles between the respective ranges: 0–120, 120–240 and 240–360. Full data samples can be seen in Figure below, where each blue point represents three images at random angles.



This resulting dataset has a total of 52,800 images in 17,600 locations. The dataset is split into train, test, and validation sets, with 60% of the data used for training, 20% for testing, and 20% for validation. Example images of three gathered data points at different coordinates can be seen in Figure below. For training, validation, and testing single images were used.

Architecture

Each evaluated model consists of two main components: a pretrained image extraction backbone and a prediction head. For the ViT backbone, we used a BEiT Feature Extractor and BeitForImageClassification. For ResNeT50, we used AutoImageProcessor and ResNetForImageClassification. Both models were obtained from HuggingFace.

ResNet-50 v1.5

We used ResNeT50 v1.5 for our convolutional neural network model. The original ResNet50 was introduced in the paper by He et al. The v1.5 version differs from the original model in the bottleneck blocks that require downsampling. The original model has a stride of 2 in the first 1x1 convolution, whereas v1.5 has a stride of in the 3x3 convolution. This difference results in ResNet50 v1.5 being slightly more accurate but also less efficient. The model was pre-trained on ImageNet-1k at resolution 224x224.

BEiT

We used BEiT(Bao et al.) for our transformer-based architecture. The BeiT is short for Bidirectional Encoder representation from Image Transformers. It was introduced in the paper BEIT: BERT Pre-Training of Image Transformers by Hangbo Bao, Li Dong and Furu Wei.

BEiT model a pre-training approach for vision transformers, designed by researchers at Microsoft. It's inspired by the BERT model used in natural language processing but adapted for computer vision tasks. The model architecture is based on the Vision Transformer (ViT).

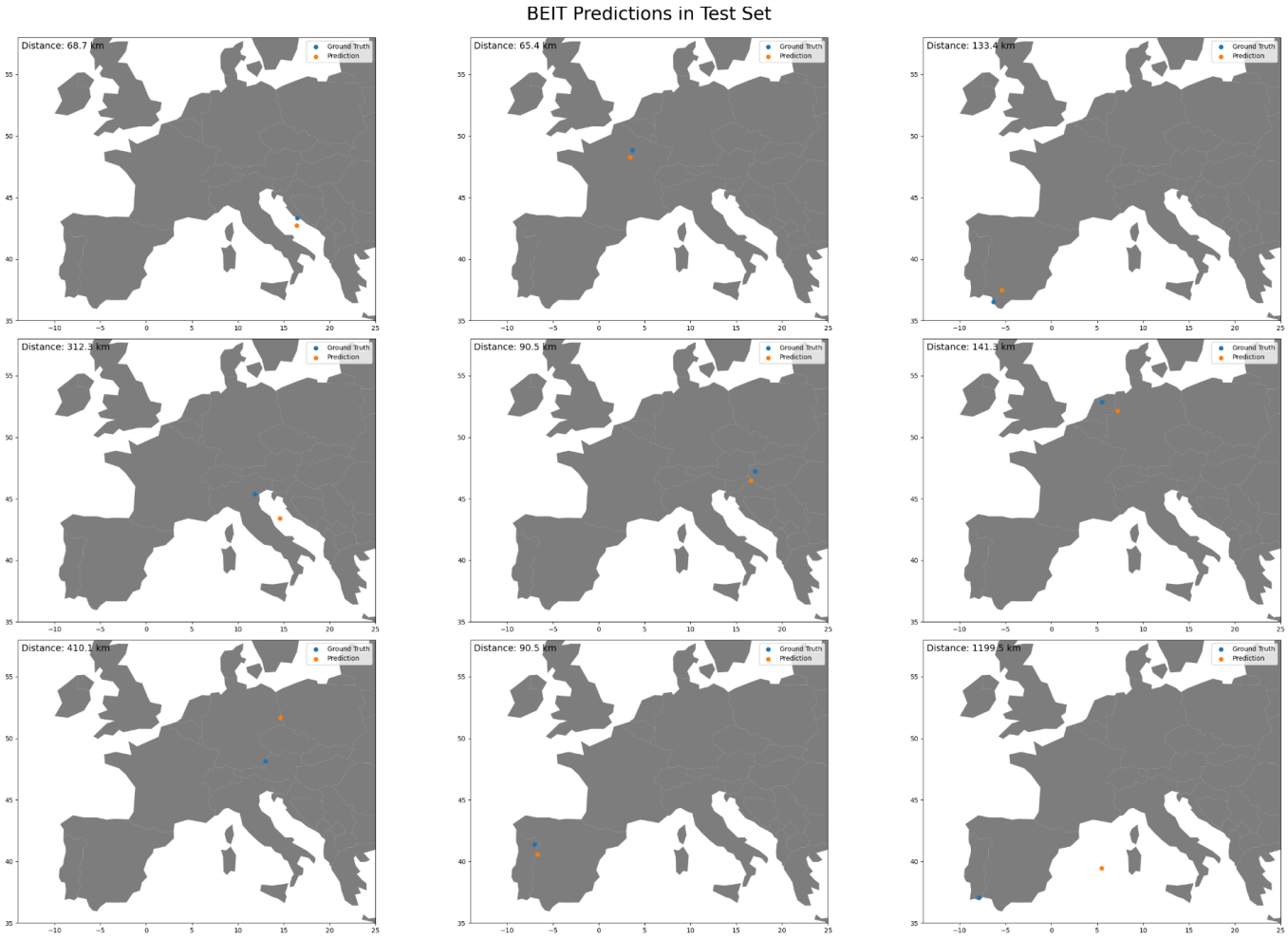
The pre-training task is composed of masked image modeling (MIM). MIM uses two views for each image, image patches, and visual tokens. The image is split into a grid of patches that are processed in sequences and become the input representation of backbone Transformer. The image is "tokenized" to discrete visual tokens by the latent codes of discrete Variational autoencoder (VAE), where discrete VAE is from DALL·E.
In contrast to the original ViT model, BEiT was pre-trained on a ImageNet-21 in a self-supervised fashion, at a resolution of 224x224 pixels, and fine-tuned on ImageNet 2012 at resolution 384x384.

Prediction Head

We have created two prediction head variants for each of the evaluated backbones. The first variant used a single linear layer to predict the coordinates of the image based on the embeddings provided by the backbones. This variant was trained using the Adam optimizer and MSE loss and equirectangular loss (equirectangular distance is an approximation of distance between two points on a sphere that has a simple enough formula to be used as a loss without introducing too complex gradients).

The second head was designed to leverage the additional information of the grid tile the current location was sampled from. The first layer is a linear layer that takes the embedding from the backbone and outputs a sparse vector of size 88, where 1 at position i means that the locations comes from the i-th tile of the grid. This sparse vector is then concatenated to the embedding and fed to a specialized regressor, which outputs the predicted location. The regressor consists of 4 hidden layers of size (250, 150, 100, 50).
For this variant's loss, we used a combination of cross entropy loss for the grid classification layer and equirectangular loss for the coordinate regression .

All the model variants, as well as the training and evaluation code is available in the repository https://github.com/erenaydoslu/computer-vision-projec



Experiments

We performed a series of comparisons between the model variants. Each prediction head variant with the transformer backbone was compared with the corresponding CNN variant with regard to the distance between their predictions and ground truth on the training, validation, and test sets. The results are presented below.

Results

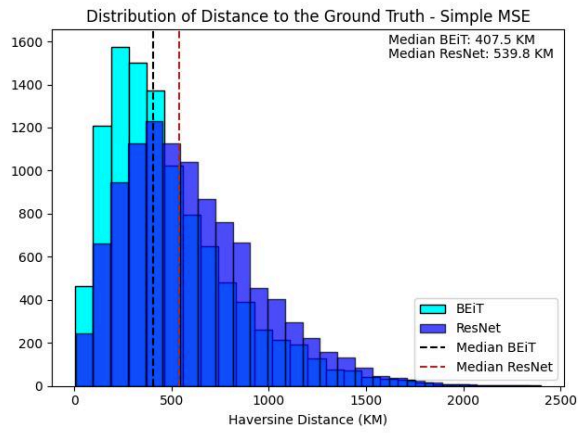
Architecture	Loss Function	Linear Layer Type	Mean Train Haversine	Mean Val Haversine	Mean Test Haversine
BEiT-Base	MSE	Single Layer	415 KM	480 KM	497 KM
BEiT-Base	Eq-Rect	Single Layer	165 KM	397 KM	403 KM
BEiT-Base	Combined	Multi Layer	395 KM	453 KM	459 KM

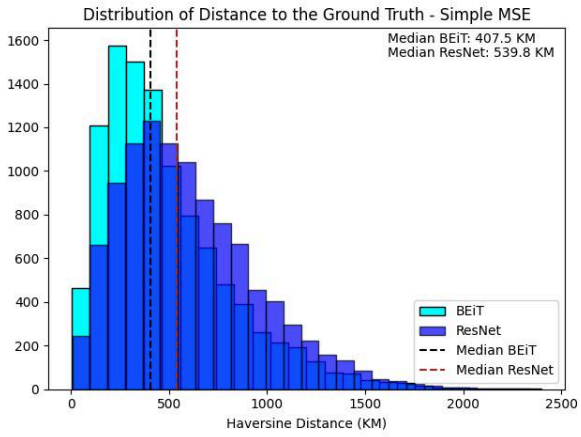
	(CE + Eq-Rect)				
ResNet-50	MSE	Single Layer	524 KM	595 KM	598 KM
ResNet-50	Eq-Rect	Single Layer	509 KM	597 KM	598 KM
ResNet-50	Combined (CE + Eq-Rect)	Multi Layer	496 KM	584 KM	589 KM

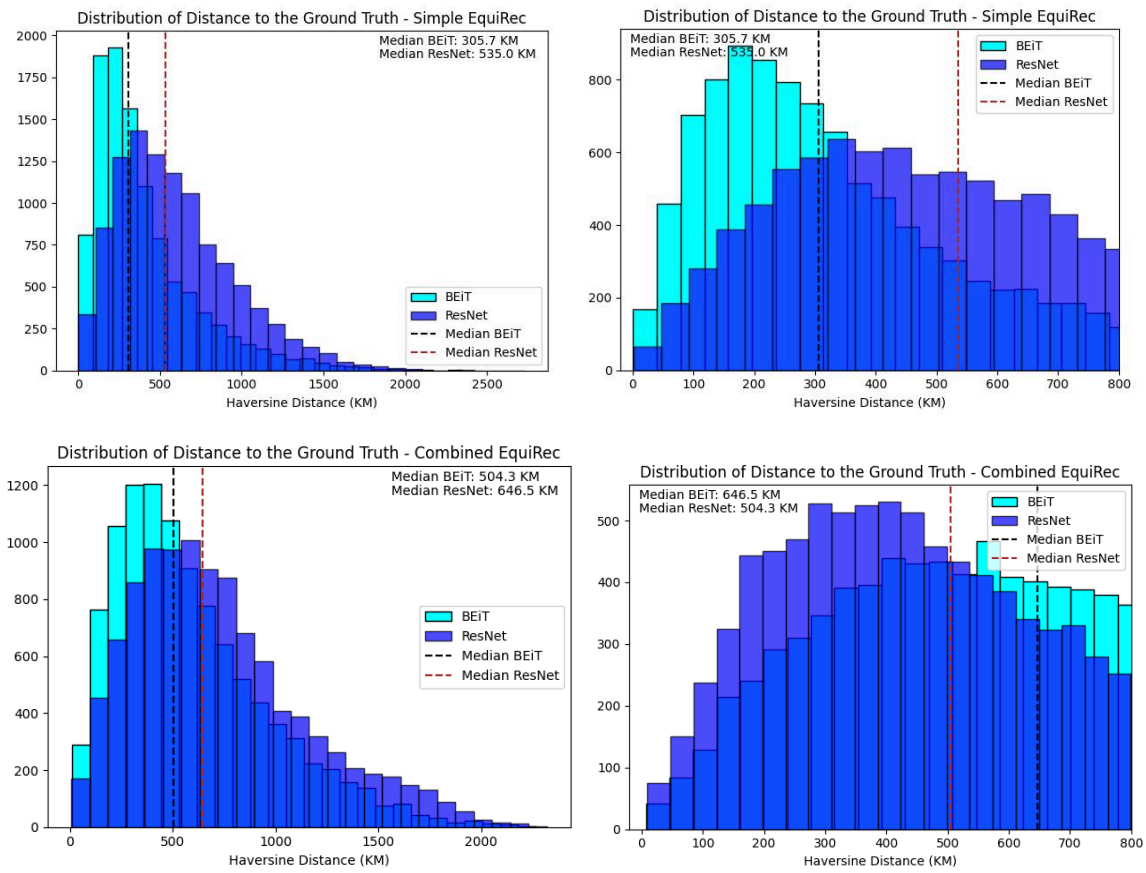
The results of our experiments comparing transformer-based architectures (BEiT) with convolutional neural network (CNN) architectures (ResNet-50) on the task of image geolocation estimation are summarized in the table above. We evaluated each architecture using different loss functions and linear layer types.

During training, we observed that models using the transformer backbone achieved their best validation performance within the first 5-10 epochs of training. Further training resulted in increased overfitting with very little improvement. We attribute that to the sheer size of the transformer models and their ability to quickly zero in on the features of pictures in the training dataset.
The CNN-based models exhibited more balanced, but much slower training, requiring over 20 epochs to achieve the performance the transformer-based models achieved within 3. However, each epoch took 4 times less time to train, due to the difference in model sizes.

The use of the equirectangular (Eq-Rect) loss function significantly improved the performance of both architectures, as seen by the decrease in mean Haversine distance across all datasets (train, validation, and test). For BEiT-Base, the mean test Haversine distance improved from 497 KM with MSE loss to 403 KM with Eq-Rect loss. However, Eq-Rect loss did not seem to improve performance in the ResNet. Furthermore, BEiT seems to consistently outperform ResNet-50 in terms of Haversine distance.







Conclusion

In this project, we conducted a comparative analysis of transformer-based (BEiT) and CNN-based (ResNet-50) architectures for the task of image geolocation estimation. Inspired by the game GeoGuessr, we aimed to determine how well each model can predict the geographical location of a given image.

Our experiments utilized a dataset collected from the Geoguessr-AI GitHub repository, Each model was evaluated using two prediction head variants: a single linear layer and a specialized regressor leveraging grid tile information. Each base variant was evaluated on different losses, including MSE, Haversine, and Equirectangular losses.

Our findings indicate that both BEiT and ResNet-50 architectures are capable of performing image geolocation estimation with varying degrees of accuracy. While CNN-based models like ResNet-50 have been traditionally employed for such tasks and have shown reliable performance, our results reveal that transformer-based models (BEiT) perform better.

The BEiT model, with its masked image modeling pre-training, demonstrated a strong ability to extract meaningful features for geolocation estimation. However, it is important to note that transformers are computationally more expensive and require extensive training data, which can be a limiting factor in some applications.

Future work could explore training transformer-based models on Google Street View Images from 3 different angles as the input together instead using a single image. Also instead of exact coordinate estimation geolocation grid classification with transformer-based architectures can be explored.

References

"Geoguessr." *GeoGuessr - Let's explore the world!*, https://www.geoguessr.com. Accessed 16 June 2024.

---. "DeepGeo: Photo Localization With Deep Neural Network." *arXiv (Cornell University)*, Jan. 2018, doi:10.48550/arxiv.1810.03077.

Weyand, Tobias, et al. "PlaNet - Photo Geolocation With Convolutional Neural Networks." *Lecture notes in computer science*, 2016, pp. 37–55, doi:10.1007/978-3-319-46484-8_3.

Vo, Nam, et al. “Revisiting IM2GPS in the Deep Learning Era.” *arXiv (Cornell University)*, Jan. 2017, doi:10.48550/arxiv.1705.04838.

“CV-GeoGuessr.” *CV-GeoGuessr*, jluij.github.io/CV-GeoGuessr.

Salehalwer. “GeoGuessr-Inspired Exploration of CNNs: Predicting Street View Image Locations.” *Medium*, 21 Jan. 2023, medium.com/@salehalwer/geoguessr-inspired-exploration-of-cnns-predicting-street-view-image-locations-e7aaa2dc19f5.

Dosovitskiy, Alexey, et al. “An Image Is Worth 16x16 Words: Transformers for Image Recognition at Scale.” *arXiv (Cornell University)*, Jan. 2020, doi:10.48550/arxiv.2010.11929.

He, Kaiming, et al. “Deep Residual Learning for Image Recognition.” *arXiv (Cornell University)*, Jan. 2015, doi:10.48550/arxiv.1512.03385.

Bao, Hangbo, et al. “BEiT: BERT Pre-Training of Image Transformers.” *arXiv (Cornell University)*, Jan. 2021, doi:10.48550/arxiv.2106.08254.

Hugging Face – the AI Community Building the Future. huggingface.co.