Comparison of Transformer, CNN, and CNN-RNN Mixed Model for Geo-Classification

Rune Myrskog, Andrew Casas, Michael Chan
Department of Computer Science
University of Toronto
{rune.myrskog, andrew.casas, michael.chan}@mail.utoronto.ca

Abstract

In this paper, we compare three models for a geo-classification task. The model's goal is to determine which country an image is taken from. Three architectures will be investigated: CNN + MLP, CNN + RNN + MLP, and ViT + MLP. For the CNN + MLP architecture, a pre-trained CNN will be used to extract features from the images, which will then be fed into a simple MLP model for classification. The ViT + MLP architecture will use a vision transformer (ViT) to divide the images into patches, which will be passed through an MLP model for classification. Finally, the CNN+RNN model will pass the feature vector into an RNN before being classified by the MLP. The findings of this project may have potential applications in various domains, including autonomous navigation and image recognition. The code for this paper can be found here [1]

1 Introduction

Image classification has become an important topic in AI, with applications in various domains such as remote sensing, surveillance, and automated driving. With the advent of deep learning techniques, image classification has been significantly improved in recent years. CNNs are widely used for image-based classification tasks, as they are designed to capture spa- tial patterns and features within an image. Transformers, on the other hand, have shown promising results in various natural language processing tasks, where they are designed to capture long-range dependencies. With the advent of vision transformers, which use self-attention mechanisms to capture dependencies between image patches, transformers have also been used for image-based classification tasks. However, their suitability for location classification on image inputs has not been widely studied, and it is unclear how they compare to CNNs in this context. In this paper, we aim to provide a comprehensive comparison of these two approaches alongside an RNN (for additional comparison) for the task of location classification on image inputs.

2 Related Works

The Image Classification task has been applied to games such as Geoguessr before [4]. However, this work used CNNs and limited their dataset to the United States. There are other ways to classify images such as Transformers [2] which was applied to the ImageNet dataset. Another model that has been applied to Image Classification is a CNN-RNN hybrid [5]. These works compared their models with CNNs to see how these newer models would perform.

[1] https://drive.google.com/drive/folders/1gZXBbljkFpQC2NJNcN7riYN4mtx6ZpFR?usp=sharing

3 Methods

This is a classification task in which the models predict log probabilities for a given input image within each region.

All three architectures aim to perform feature extraction and classification, using cross entropy as the loss function. with the specific differences in the models impacting the overall performance of the model. The high-level diagram of the models is in Appendix A.

Our transformer model uses the Vision Transformer which accepts images 224x224 pixels, thus the images are converted to 224x224 before being passed as input. This transformer splits the image into a sequence of 16x16 patches fed into the transformer. This approach is particularly useful for larger images, where the ViT can capture both global and local features [2].

We used this model pre-trained on 14 million images from the ImageNet-21k dataset. Using the transformer as an encoder to capture major spatial information, the 1024 feature vector produced by the transformer is fed into a simple two-layer MLP using ReLU and Dropout in between the layers to create the final 19x1 output vector.

Our CNN model is a modified version of [1] where we added an extra layer to increase its complexity and improve the model. The CNN model takes in 128x128 pixel images which we first normalize the pixel values to reduce the impact of lighting and color variations in the images.

Our CNN model has three convolution layers, where each convolution layer applies a 3x3 filter with a stride of 1 and padding of 1. After each convolution layer, we apply 2x2 max-pooling to reduce overfitting and increase efficiency and then use a ReLU activation function. At the end of the Model, we apply a Dropout to reduce overfitting. During training, we use cross-entropy loss, so we apply the log-softmax activation function to predict the class probabilities.

Our CNN-RNN model combines a pre-trained CNN, specifically the MobileNet model, for feature extraction and a Gated Recurrent Unit (GRU), which is an RNN variant, for capturing temporal dependencies in the sequence of extracted features. MobileNet was chosen over ResNet due to its lighter architecture, which offers faster training times and lower memory requirements while maintaining competitive performance. This hybrid approach aims to leverage the strengths of both CNNs and RNNs, allowing for a more robust representation of spatial and sequential patterns in the images.

The input images are resized to 224x224 pixels and transformed into tensors. The pre-trained MobileNet model is used as a feature extractor, generating a feature vector for each image. To maintain the spatial relationships between features, we apply adaptive average pooling and reshape the output of the MobileNet to feed it into the GRU layer.

The GRU layer is followed by a fully connected layer for classification. The output of the classification layer is a 19x1 vector, which represents the log probabilities for each of the 19 countries.

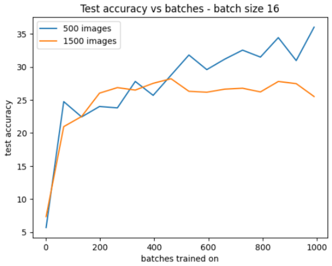
The combined CNN-RNN model is trained using cross-entropy loss and optimized with the Adam optimizer, allowing for an efficient and effective training process. By utilizing a pre-trained CNN for feature extraction and a GRU for capturing sequential patterns, this hybrid model aims to provide a comprehensive representation of the input images, potentially outperforming the standalone CNN and transformer models in the task of location classification based on image inputs. [5].

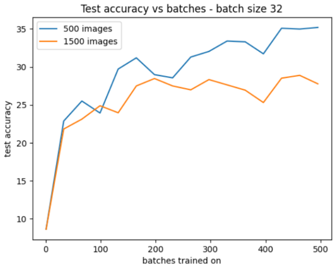
3.1 Data

We collected our data from [3], which contained labeled folders with images of various countries. We selected only the countries that had over approximately 700 images, resulting in a dataset comprising 19 countries. The images include both rural and urban areas, offering a diverse representation of each nation's unique characteristics. For the Transformer and CNN-RNN Hybrid models, we resized the images to a resolution of 256x256 pixels, while for the CNN model, we used a lower resolution of 128x128 pixels. This resizing step ensures that the input data is suitable for each model and allows for a fair comparison of their performance. Although the dataset may not be exhaustive or entirely representative of each country's distinctive features, it provides a solid foundation for exploring and comparing the performance of various model architectures in the geo-classification task.

4 Experiments and Results

For each of the models, we trained on batch sizes 16 and 32, with 500 images per country and (up to) 1500 images per country. results for transformer:

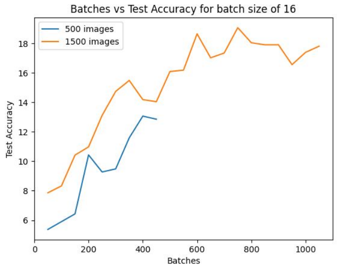


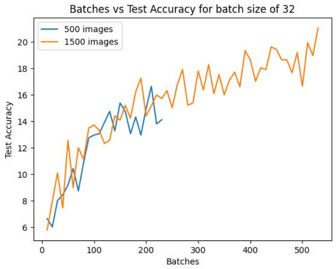


We can see the in the transformer model, the test accuracy being better over time for 500 images per country, achieving a top accuracy of over 35% while 1500 images levels out around 28% accuracy. One reason for this may be that the transformer was trained for a little over 2 epochs in the case of 500 images per country in order to match the number of batches in the 1500 image case. In both batch sizes, the 500 image test accuracy curve starts to pass the 1500 image accuracy after the first pass through the training data. Another possible reason for this discrepancy is that not all countries had 1500 or more images, for example, Sweden has only around 750 images while France has over 1500 images, so Sweden is represented half as much as France, this may contribute to lower test accuracy.

The low test accuracy in general is expected. geo-classification in general is a hard task, and this model is doing a 1 of 19 classification. When only using 10 countries, the validation accuracy goes up to over 60%, comparable to DeepGeo results, and when only using 5 countries the validation accuracy goes up to over 80%.

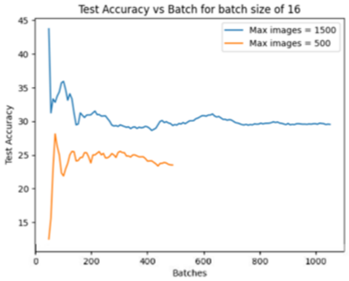
results for the CNN:

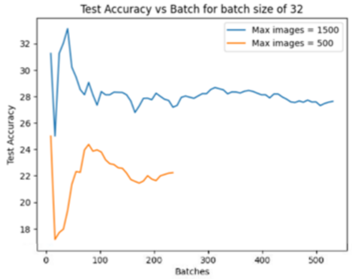




For the batch size of 16, the test accuracy for the model that used 500 images was always below the model that used 1500 images but for the batch size of 32, the test accuracies were very similar. One possible reason for this could be that the batch size of 16 leads to overfitting since it would be more sensitive to noise. Once again the test accuracy is low but it is about 15% lower than the Vision Transformer. Since we only trained the model for only 500-1000 batches the models could perform better after significantly more batches.

Results for the CNN-RNN:





In the CNN-RNN model, performance varies with different maximum images per country (1500 and 500) and batch sizes (16 and 32). With a batch size of 16, the 1500-images model has a stable 28% test accuracy, while the 500-images model fluctuates, peaking at 32% but averaging 25%. With a batch size of 32, the 1500-images model maintains a consistent 26% accuracy, whereas the 500-images model has unstable accuracy, peaking at 30% but averaging 23%. The CNN-RNN model benefits more from larger datasets due to its architecture, capturing spatial and temporal features. However, test accuracy remains low for both scenarios, likely due to the complexity of the 1-of-19 geo-classification task. Reducing the number of countries could improve validation accuracy, suggesting the model might perform better with fewer countries.

The different results can be attributed to the distinct architectures of the models. The Vision Transformer's superior performance with a smaller dataset might be due to the model's ability to capture both global and local features, especially when considering its training for a little over 2 epochs with 500 images per country. The CNN model's lower performance can be attributed to its relatively simpler architecture, which may require more training batches to achieve similar performance to the other models.

The CNN-RNN model's preference for a larger dataset can be explained by its architecture, which combines both convolutional and recurrent neural networks to capture spatial and temporal features in the images. This combination might make better use of a larger dataset, resulting in higher test accuracies. However, the overall test accuracies for the CNN-RNN model remained relatively low, which might be due to the complexity of the geo-classification task itself.

In general, the low test accuracies across all models are expected due to the inherent difficulty of the geo-classification task, which involves a 1-of-19 classification. As evidenced by the transformer model results, reducing the number of countries can significantly improve the validation accuracy. This suggests that all models might benefit from a more focused classification task with fewer countries.

It is noticeable that the three models exhibit different performance characteristics, with the Vision Transformer showing better results with smaller datasets, the CNN model requiring more training batches for improved performance, and the CNN-RNN model benefiting from larger datasets. These differences can be attributed to the unique architectures of the models and provide valuable insights for further optimization and improvement in the geo-classification task.

Overall the performance of the models at first glance seems pretty good when compared to something like DeepGeo [4]. Our best model had validation accuracies reaching up to 40%, the same as DeepGeo. However, there are several factors to consider. DeepGeo classifies states in the USA, this is a 1 of 50 classification versus our 1 of 19. Additionally, our classes (countries) span a wide range of geographical areas, adding more diversity into the images which can be used in the classification. For example, the pictures from Thailand and the pictures from Canada are fairly distinct.

5 Summary

In this project, we compared three architectures (CNN + MLP, CNN + RNN + MLP, and ViT + MLP) for country classification based on images. The Vision Transformer showed better performance with smaller datasets, the CNN-RNN model favored larger datasets, and the CNN model needed more training batches for improvement. The differences in performance are attributed to the distinct architectures, offering insights for future optimization and improvements.

Low test accuracies were due to the geo-classification task's complexity, involving a 1-of-19 classification. Reducing the number of countries could improve validation accuracy, indicating all models might benefit from more focused tasks with fewer countries.

These findings have potential applications in autonomous navigation and image recognition. Future work could involve model optimization, exploring new architectures, or focusing on more specific classification tasks to enhance performance.

References

[1] Alessandro Bombini. Using cnn to classify images w/ pytorch. `https://www.kaggle.com/code/androbomb/using-cnn-to-classify-images-w-pytorch#kln-299`, 2021. Accessed: April 18, 2023.

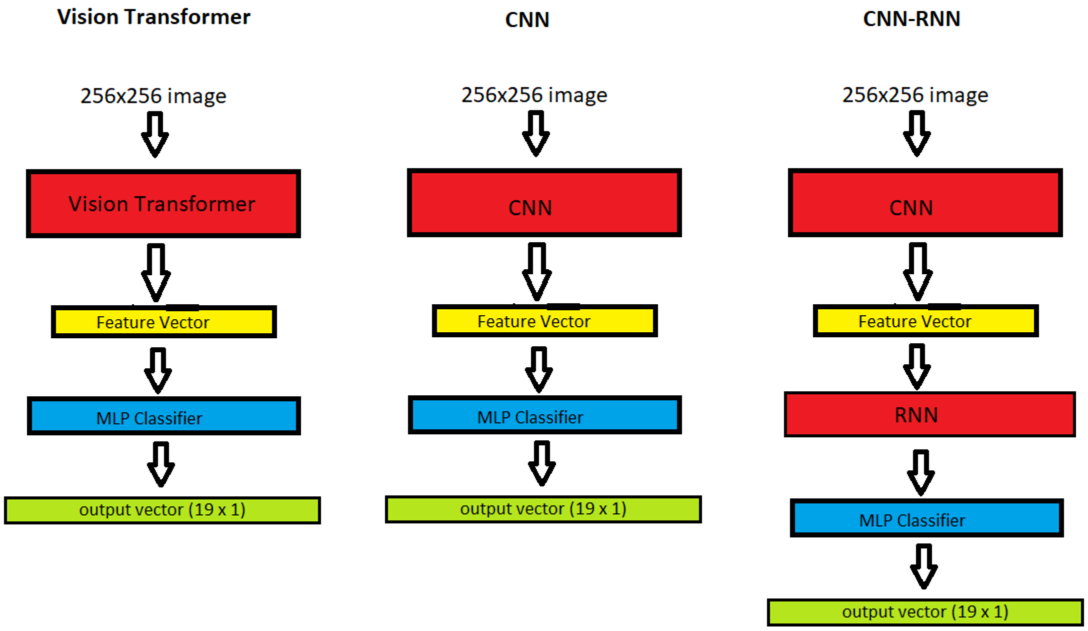
[2] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2020.

[3] Rohan K. Geolocation geoguessr images 50k. `https://www.kaggle.com/datasets/ubitquitin/geolocation-geoguessr-images-50k`, 2023. Accessed: April 18, 2023.

[4] Sudharshan Suresh, Nathaniel Chodosh, and Montiel Abello. Deepgeo: Photo localization with deep neural network. *CoRR*, abs/1810.03077, 2018.

[5] Yin, Qiwei, Zhang, Ruixun, and Shao, XiuLi. Cnn and rnn mixed model for image classification. *MATEC Web Conf.*, 277:02001, 2019.

Appendix A


Vision Transformer
CNN
CNN-RNN
256x256 image
Vision Transformer
Feature Vector
MLP Classifier
output vector (19 x 1)
256x256 image
CNN
Feature Vector
MLP Classifier
output vector (19 x 1)
256x256 image
CNN
Feature Vector
RNN
MLP Classifier
output vector (19 x 1)