Photo Geolocation Recognition Based on Convolutional Neural Networks

Yifan Yu • Yao Chen • Yancheng Li • {yifanyu, yaochen1, lycheng}@Stanford.edu

Introduction

Motivation

Photo Geolocation Recognition could be implemented on a social network platform to provide users with location information on unlabeled photos they may be interested in. Some of previous researches on geo-localization do pay attention to architectures, yet most of them chose to focus only on landmarks. Our study takes care of landmark and non-landmark buildings.

Goal

Utilize the architectural-related content and extract architectural features from different convolutional neural networks which benefit the general geolocation recognition process. Implement different CNN architectures for detecting architectural styles of landmark and non-landmark buildings, and compare their performance.

Data Acquisition & Preprocessing

Dataset

We built a dataset consisting of city labeled architectural photos. We chose **9 major cities** around the world and collected approximately **4500 photos** on each place's architectures.

Fig1. Picture taken in 9 cities Vienna, Amsterdam, Morocco, Rome, New York, Shanghai, Paris, Seoul, Kyoto

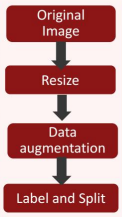


Fig2. Data preprocessing pipeline

label	city
0	Shanghai
1	New York
2	Kyoto
3	Rome
4	Morocco
5	Paris
6	Seoul
7	Amsterdam
8	Vienna

Fig3. labeled data

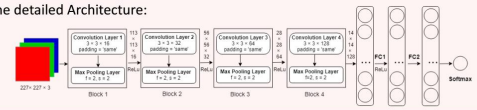
Data Augmentation

Fig4. Pictures obtained after implementing a series of random data augmentation procedures, including rotating, zooming, flipping and distorting

Models

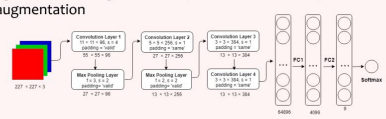
Model I : Baseline Model

- A CNN model with 4 convolutional layers and 2 fully-connected layer
- Trained with softmax cross-entropy loss using AdamOptimizer and utilized the batch normalization technique
- The detailed Architecture:



Model II : Customized AlexNet Model

- Implemented the AlexNet Architecture with the fifth convolutional layer and the last max pooling layer removed (with the goal of reducing overfitting)
- Experimented with a variety of combinations of regularization techniques including dropout, L2 regularization (scale = 0.1, 0.01, 0.01), smaller batch size and image augmentation



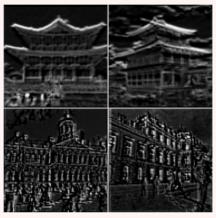
Model III : VGG16 with Pre-Trained Weights

- Performed transfer learning with VGG16 with weights from the Places365 dataset (a scene classification task) and the ImageNet dataset (an object detection task)
- Conducted 4 experiments: 1) retrain the last fully-connected (fc) layer only 2) retrain all the fc layers 3) retrain all the fc layers + the fifth convolutional layer 4) retrain all the fc layers + the fourth & the fifth convolutional layer

Visualization

Transfer Learning VGG – Layer Visualization

To better understand and interpret the performance of VGG16 model and pre-trained weights, we visualized all the ConvNets layers. The 4 images on the right are the visualization of the second convolution layer from the second block of the VGG16 architecture. VGG learns the architectural style but fails to identify the geolocation of the image, due to the similarity of building styles and the lack of additional environmental-related information to distinguish between cities.
(Fig5. Layer Visualization – Seoul, Kyoto, Amsterdam, Vienna)



Results

- The Best Baseline Model is achieved with dropout (keep_prob = 0.5) and learning rate = 1e-3
- The Best AlexNet Model is achieved with using dropout (keep_prob = 0.5), L2 regularization (scale = 0.001), batch size = 32, and learning rate = 1e-4
- The Best VGG Model is achieved with using ImageNet pre-trained weights for the first three convolutional layer, learning rate = 1e-4

Model	Train Accuracy	Evaluation Accuracy
Best Baseline	82.7%	50.4%
Best AlexNet	76.5%	56.7%
Best VGG	96.2%	42.7%

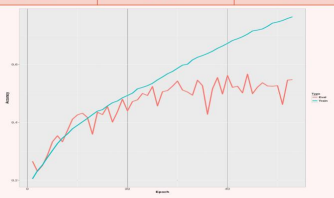


Fig5. Train and Eval Accuracy for the Best AlexNet Model

Discussion

- Performing transfer learning fails to help achieve good performance in this application, as the discrepancies in architectural styles are hard to be detected with weights pre-trained on datasets with broad categories
- Training from scratch is the better approach, with special attention needed for avoiding overfitting

Nest Step

- Increase the number of cities in the dataset
- Further reduce overfitting and improve the current best model by implementing different data augmentation procedures, exploring more combinations of regularization techniques, and experiment with other simplified versions of successful CNN architectures.

Reference

[1] A. R. Zamir, "Introduction to large-scale visual geo-localization," Springer International Publishing, pp. 1–18, 2016.
[2] Flickr dataset." https://www.flickr.com/.
[3] K. A. O. A. T. A. Zhou B, Lapedriza A, "Places: A 10 million image database for scene recognition.," IEEE Trans Pattern Anal Mach Intell., 2017.
[4] E. Russell91, "Caffe to tensorflow," 2015. [6] "Local response normalization in convolutional neural networks."