université de BORDEAUX

Report

Land Use Classification in Remote Sensing Images by Convolutional Neural Networks

Authors:
Xuan Thanh Pham
Manh Tu Vu

Supervisor:
MarieBeurton-Aimar

July 31, 2017

Contents

Abstract

In this project, we use the methods developed in the article Land Use Classification in Remote Sensing Images by Convolutional Neural Networks [**?**] to classify street images of the pagoda in Vietnam (Ho Chi Minh City). We collect the dataset from Google to train the network. After that, we try to experiment and analyze the result by testing our images of the pagoda which taking with our smartphones.

1 Introduction

1.1 Image classification

Image classification is the task of taking an input image and outputting a class which is a dog, cat, etc... or a probability of classes that best describes the image. For humans effortlessly to do the task of recognition, it comes naturally since we were born. But in computer vision, the computer must be trained in order to classify a set of data into different classes or categories. Training is the key to the success of classification.

1.2 Convolutional Neural Networks

Convolutional Neural Networks (ConvNets or CNNs) are a special type of Artificial Neural Networks. These models are designed to emulate the behavior of a visual cortex. CNNs perform very well on visual recognition tasks and has been some of the most influential innovations in the field of computer vision.
This model described in the article[**?**] as the most powerful in remote sensing datasets task. So, we're going to implement in our project in order to gain the best performance.

1.3 Caffe

Caffe is one of the most popular libraries for deep learning (convolutional neural networks in particular). It is developed by the Berkeley Vision and Learning Center (BVLC) and community contributors. It supports the wide range of opera systems and graphical processing units in order to help us to train our model.
Caffe also makes easy to understand and configure our CNN layers.
In this project, we will use Caffe as our main deep learning framework.

2 Task definition

In order to classify street images of the pagoda in Vietnam (Ho Chi Minh City). We have to build a CNN model has an ability to recognize the photo of the pagoda which it has never seen before.
We also have to collect enough images of the pagoda for both trains and validates

tasks.
Finally, after trained our model, we have to test its performance by the other datasets and build the web page to test our street images.

3 Our solution

3.1 Dataset

We found a MIT Computer Science and Artificial Intelligence Laboratory website [?] which is a scene-centric database called Places, with 205 scene categories. After downloaded 132Gb images, we built our own train dataset including:

- Pagoda: 15,100 images.
- Not_pagoda: 34,906 house images which are schoolhouse, courthouse and lighthouse. We chose those kinds of house because their building structure is a little bit similar with pagoda structure.

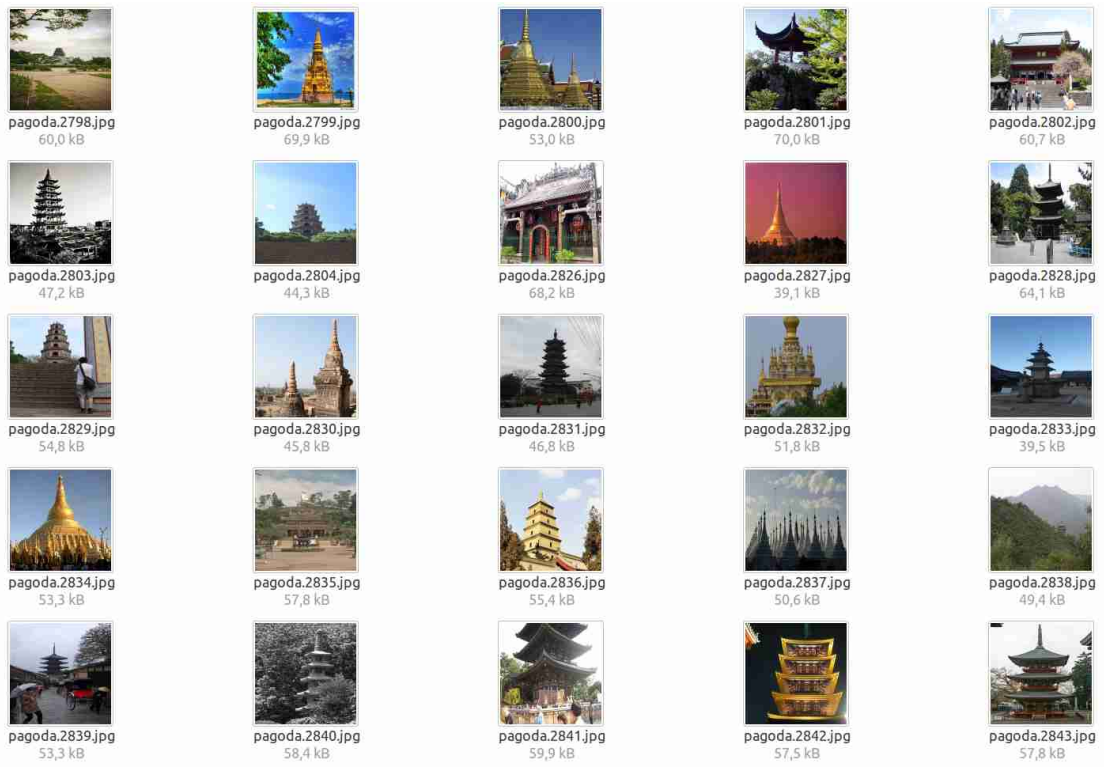


Figure 1: Sample pagoda images of training dataset

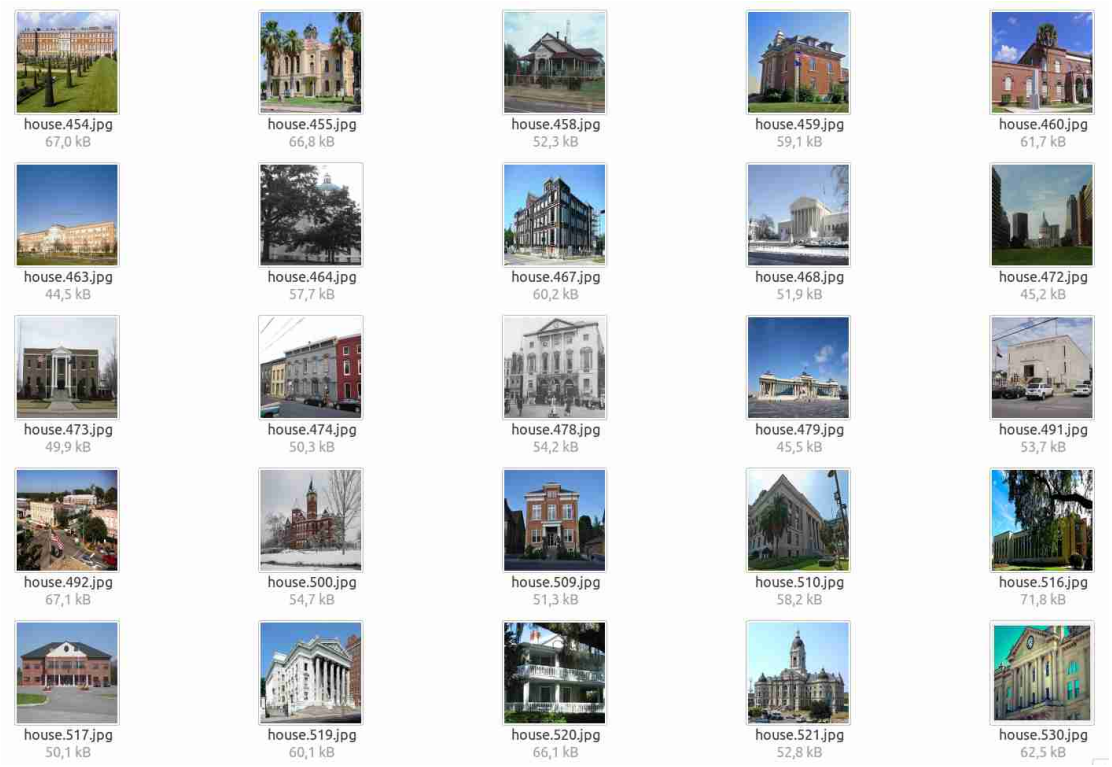


Figure 2: Sample not_pagoda images of training dataset

3.2 Hardware

Because training CNN model with a large of dataset consumes a lot of time and resource, a standard PC or Laptop can't handle that. We already test our CNN model with a MacBook 2015 - Core i5 and 8GB Ram without external GPU. It takes about 12 hours for 200 iterators. This quite slow and our MacBook can't run forever.
To solve this problem, we hire an EC2 instance from Amazone Web Service

CPU: Intel Xeon E5-2670 (Sandy Bridge) Processors
GPU: NVIDIA GRID GPUs, each with 1,536 CUDA cores and 4GB of video memory

Model	GPUs	vCPU	Mem (GiB)	SSD Storage (GB)
g2.2xlarge	1	8	15	1 x 60

3.3 Installing Caffe

After some research, we found that currently, it has no completed script to install Caffe on Ubuntu 16.04 with Cuda 8 and CuDNN. It'll be difficult if we do it manually. So, to solve this problem and also help the other people who want to install Caffe, we decide to write a Bash script to automatic install Caffe with all dependencies, Cuda 8 and CuCNN under Ubuntu 16.04. You can see our

topic at https://github.com/BVLC/caffe/wiki/Caffe-installing-script-for-ubuntu-16.04—support-Cuda-8

3.4 Data preparation

Because of our dataset itself contain a lot of images, if we read them file by file, it can waste much time just because the limit of I/O. To solve this problem, Caffe using LMDB (Lightning Memory Mapped Database) with high throughput. Our task is convert our dataset from stored to LMDB.

Before convert or dataset to LMDB, we use OpenCV to increase Histogram Equalization & resize each images into 227x227 pixel. This help the Convolutional layer easy to recognize the different of the main object in the image.

We separate our dataset into two part, one is training (5/6) & the other is validating (1/6). The validating help us to estimate how well our model has been trained and model properties (mean error for numeric predictors, classification errors for classifiers, recall and precision for IR-models etc.)

The last thing to do is computing image mean because the model requires us to subtract the image mean from LMDB we created before to improve detection accuracy.

3.5 Model definition

Our model is reuse from bvlc_reference_caffenet model, which is a replication of AlexNet with a few modifications. The original bvlc_reference_caffenet was designed for a classification problem with 1000 classes. However, we just need to indentify if given image is pagoda or not. So, we change num_output of the last InnerProduct layer from 1000 to 2.
We also change some information about location of mean image file, train & validate lmdb in order to make the model work.
Finally, we have our model as describe in the image below:

Figure 3: Caffe model

3.6 Solver definition

Our model using Stochastic gradient descent solver method. We run our model with 40.000 iterators, drop leaning rate every 2500 iterators and take a snapshot every 5000 iterators.

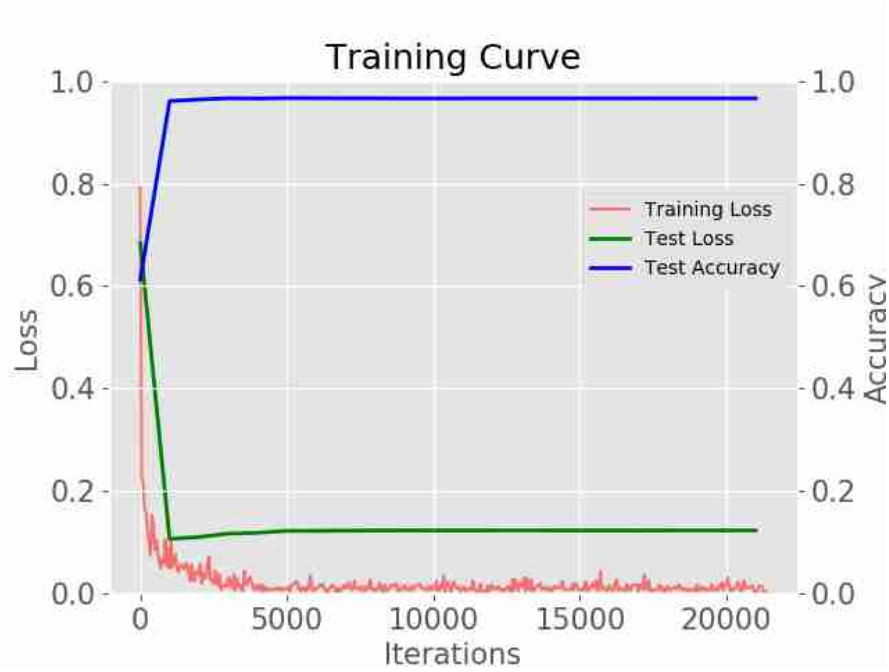
3.7 Model training

In order to reduce the time to train & gain the best performance, we utilize the trained bvlc_reference_caffenet as a starting point of building our pagoda classifier using transfer learning. This model was trained on the ImageNet dataset which contains millions of images across 1000 categories
We used this fine-tuning strategy for training our model.

4 Experimental

4.1 Plotting the Learning Curve

A learning curve is a plot of the training and test losses as a function of the number of iterations. These plots are very useful to visualize the train/validation losses and validation accuracy.
We can see from the learning curve that the model achieved a validation accuracy of 98%, and it stopped improving after 1000 iterations.

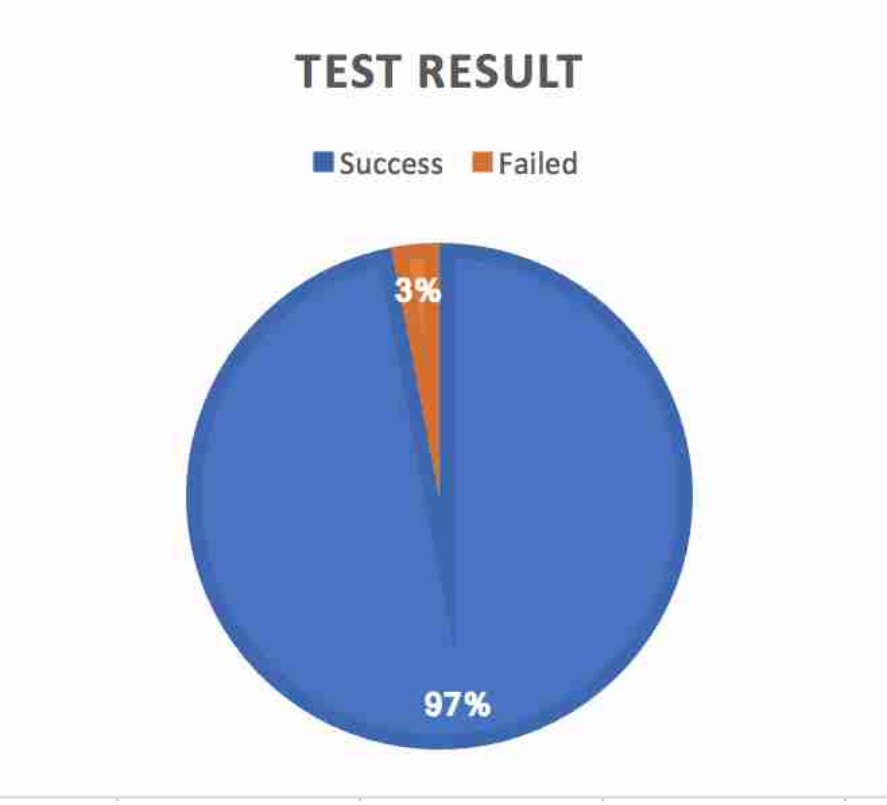


4.2 Test performance

To measure or CNN trained performance in a most objective way, we built the test dataset(1,519 images), which our CNN has never seen before. We collect the result after the test process & compare with our expected result. Our dataset incudes:

- Pagoda: 999 images.
- Not_pagoda: 520 house images.

After train model, we test it with our dataset and get the accuracy 97% (1,477 images is predict right, 47 images is not).



4.3 Resource

Testing page: http://54.255.219.234:8080/
Completed source code: https://github.com/glmanhtu/deeplearning-pagoda
Train dataset: https://drive.google.com/open?id=0B60FAQcEiqEyWlY1Uld-pVU5hT2c
Test dataset: https://drive.google.com/open?id=0B60FAQcEiqEyWUJ2dUh-mUDllb1k
Street image pagoda: https://drive.google.com/open?id=0B60FAQcEiqE-yRzBsYmFvZGNmRGs