Machine Learning Engineer Nanodegree Capstone Proposal

Omar Talabay
stu251087@hotmail.com

Jan 28, 2018

1 Domain Background

Remote sensing (RS) image classification has attracted many researchers in the field of machine learning (ML) [3], [6], [7], [10]. RS images used to be low in resolution so that simple methods were used. In [11], it is stated that such simple methods don't perform well on images with complex structures. Most of those methods rely on hand-crafted features to feed different ML algorithms. This, in fact, requires domain knowledge expertise and takes a long time to capture complex structures in RS images.

Nowadays, RS images have increased in resolution and been publicly available, i.e. via Google Earth. With the increase of RS resolution, it is possible to recognize much of what is on the Earth's surface, i.e. ships, cars, airports, etc. The advent of high resolution RS images has opened eyes on how to utilize and exploit them efficiently since old techniques used early days are not satisfactory [2]. Thus, more advanced techniques are needed and, hence, deep learning (DL) techniques have come into place. There are many applications that make use of RS images, some are, but not limited to, natural hazard detection and geo-spatial object detection [4], [8].

To the best of our knowledge, DL and transfer DL have achieved the state-of-art performance in computer vision. Therefore, the focus of this project will be on these techniques.

2 Problem Statement

This project is an attempt of dealing with RS image classification problem using DL and transfer DL techniques. Authors in [3] provides a large scale data set and applies different classification techniques. The most accurate results are convolution neural networks (CNNs) based. To be more specific, the focus of this project is on CNNs networks transfer learning and fine tuning CNNs architectures other than the ones implemented in the paper.

3 Datasets and Inputs

The authors in [3] collected a data set called NWPU-RESISC45 from Google Earth. This data includes 45 classes of RS images. Each class has 700 images. An image in this data set is of 256 X 256 pixels in Red-Green-Blue color space. The classes included in the data set are airplane, airport, baseball diamond, basketball court, beach, bridge, chaparral, church, circular farmland, cloud, commercial area, dense residential, desert, forest, freeway, golf course, ground track field, harbor, industrial area, intersection, island, lake, meadow, medium residential, mobile home park, mountain, overpass, palace, parking lot, railway, railway station, rectangular farmland, river, roundabout, runway, seaice, ship, snowberg, sparse residential, stadium, storage tank, tennis court, terrace, thermal power station, and wetland.

According to [3], this data set has the following characteristics:

1. Large scale: as compared to the publicly available data sets, this is the larges one exists.

2. Rich Image Variation: images were carefully selected under different conditions, i.e. under all kinds of weathers, seasons, illumination conditions, imaging conditions, and scales. Therefore, this data set provide a better representative sample for each class.

3. High Within-Class Diversity and Between-Class Similarity.

The experimental setup should follow the same pattern used in [3]. First, There are two settings for training and testing split. One is 10% for training and 90% for testing. The other is 20% for training and 80% for testing. Second, experimenting with three CNNs networks not used in the paper in two different settings. The goal of these steps is to be consistent with the experiments done and be able to compare the results.

4 Solution Statement

Pre-trained CNN models, known as transfer DL, have been used widly in computer vision tasks. They ,in fact, have achieved very high results even with limited amount of data. Terefore, the solution is to implement classifiers for RS images using three different pre-trained CNN models as feature extractors with two settings. The difference between them is that one is fine-tuned on RS images while the other is not.

5 Benchmark Model

Keras is a high level neural network API for python. It includes many CNNs architectures with their pre-trained models. This in fact, will increase the deplyment time [5]. In addition, Sklearn is a machine-learning python API. This API has many metrics for performance evaluation like overall accuracy (OA) and confusion matrix. The figures 1 and 2 show the best results achieved by [3] on the data set:

Figure 1: OA under DL based CNN features with different training ratios

Features	Training ratios	
	10%	20%
AlexNet	76.69±0.21	79.85±0.13
VGGNet-16	76.47±0.18	79.79±0.15
GoogLeNet	76.19±0.38	78.48±0.26

6 Evaluation Metrics

To be consistent in comparing the results with the ones in [3], two evaluation metrics are adopted. The first one is the overall accuracy (OA) which is defined as the ratio of correctly

Figure 2: OA of the three different find-tuned DL CNNs features under different training ratios

Features	Training ratios	
	10%	20%
Fine-tuned AlexNet	81.22±0.19	85.16±0.18
Fine-tuned VGGNet-16	87.15±0.45	90.36±0.18
Fine-tuned GoogLeNet	82.57±0.12	86.02±0.18

classified images regardless of which class they belong to, to the total number of images. The second is confusion matrix. It is used to analyze the errors of each class and spots where the confusion a classifier makes.

7 Project Design

The intended plan is to implement three different CNNs architectures other than the ones used in [3] in two different settings. Both are used as feature extractors. However, the difference is one setting is fine-tuned before being used. Keras [5] offers 9 CNNs architectures with their pre-trained models. Those models are Xception, VGG16, VGG19, ResNet50, InceptionV3, InceptionResNetV2, MobileNet, DenseNet, and NASNet.

Before getting started, a pre-processing may be needed. Some CNNs architectures require different types of normalization. Also, the choice of three CNNs requires more investigation. In order to implement this project, there are many python packages needed. However, there are three main ones, TensorFlow [1], Keras [5], and Sklearn [9].

The experiment work flow is divided into two parts. The first one is to use pre-trained models as feature extractors and feed them to a linear SVM. The second one is similar but with fine tuning the feature extractors first. In the end, the results should be compared with figures 1 and 2.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] T. Blaschke. What's wrong with pixels? some recent developments interfacing remote sensing and gis. *GeoBIT/GIS*, 6:12–17, 2001.

[3] G. Cheng, J. Han, and X. Lu. Remote sensing image scene classification: benchmark and state of the art. *Proceedings of the IEEE*, 105(10):1865–1883, 2017.

[4] G. Cheng, J. Han, P. Zhou, and L. Guo. Multi-class geospatial object detection and geographic image classification based on collection of part detectors. *ISPRS Journal of Photogrammetry and Remote Sensing*, 98:119–132, 2014.

[5] F. Chollet et al. Keras. `https://github.com/keras-team/keras`, 2015.

[6] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren. Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning. *IEEE Transactions on Geoscience and Remote Sensing*, 53(6):3325–3337, 2015.

[7] F. Hu, G.-S. Xia, J. Hu, and L. Zhang. Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery. *Remote Sensing*, 7(11):14680–14707, 2015.

[8] T. R. Martha, N. Kerle, C. J. van Westen, V. Jetten, and K. V. Kumar. Segment optimization and data-driven thresholding for knowledge-based landslide detection by object-based image analysis. *IEEE Transactions on Geoscience and Remote Sensing*, 49(12):4928–4943, 2011.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[10] J. Wang, C. Luo, H. Huang, H. Zhao, and S. Wang. Transferring pre-trained deep cnns for remote scene classification with general features learned from linear pca network. *Remote Sensing*, 9(3):225, 2017.

[11] G.-S. Xia, J. Hu, F. Hu, B. Shi, X. Bai, Y. Zhong, L. Zhang, and X. Lu. Aid: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(7):3965–3981, 2017.