Novel vs Pre-Trained Deep Learning Models for Classification of Landscape Images

Project Milestone

Aidan Gray **Nathaniel Keplinger** **Kael Kleckner** **Manav Vats**

Abstract

Image classification is a well known and well studied supervised learning problem in machine learning and deep learning. Many datasets exist containing labeled images, photographs, or pictures with a defined set of target classes, so for our project we select a dataset of natural landscape images. In this project, we build and train our own multi-class classification deep learning model, and compare its performance on the test set to that of several best-of-breed pre-trained models refined on the specific training data. In addition to our novel deep learning model, we train a couple of elementary statistical learning models including a Random Forest and Support Vector Machine (SVM) to provide a measure of baseline performance (i.e. assess whether more complex deep learning techniques are justified). Additionally, we briefly explore model performance after the introduction of noise patterns to the original images. We then compare and contrast all of the models and discuss their performance and future work that could be done.

1 Project Description

1.1 Project motivation and goals

Our motivation for this project is to increase our understanding and experience with transfer learning and building deep learning models. We put specific focus on developing convolutional neural networks (CNNs) as a way to gain knowledge and experience from the practical application of such models. In addition to building our own model from scratch, we find it valuable for us to explore existing state-of-the art models, and gain familiarity with the process of using external and pre-trained models. While the dataset we will use is not very large, our goal is to develop a more basic but high-performing (> 90% accuracy) deep learning model. We also aim to successfully import several pre-trained models, refine them on our specific data, and compare their structures to our model through a discussion of their classification performances. We do not anticipate any discoveries or breakthroughs, as our objective is to gain more experience applying advanced machine learning techniques.

1.2 Programming languages

We will use python and standard machine learning libraries for analysis and model building. These libraries include:

1. Scikit-learn
2. Tensorflow
3. Pandas

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

4. Numpy

2 Previous Work

1. On image classification: City Images vs. landscapes

   The purpose of this experiment was to identify the abstract and discrete parameters that define a landscape vs a city image. The abstract parameters were determined by survey. Volunteers identified one of eleven sub categories for each picture. These sub categories were then placed into two primary categories, landscapes and cities. This served as the primary database for their analytical approach to the discernment. With the database in hand, they then analyzed key parameters under each sub category such as color based features and edge direction. Some important notes on these analytical parameters: cities tended to have a specific color range and a plethora of vertical edges whereas natural scenes had earthy tones mixed in with blues and whites and had very little as far as distinguishable vertical edges. The best resulting classifier was a 5-NN classifier with a 0.939 accuracy.

   A. D. I. T. Y. A. VAILAYA, A. N. I. L. JAIN, and H. O. N. G. J. I. A. N. G. ZHANG, "On image classification: City Images vs. landscapes," Pattern Recognition, vol. 31, no. 12, pp. 1921–1935, 1998.

2. A survey of image classification methods and techniques for improving classification performance

   The basis for many environmental and socioeconomic applications is remote-sensing research focused on image classification. D. Lu and Q. Weng summarize the major developments in advanced classification, the techniques that will be used to improve the accuracy of the classification, and issues that affect the chance of success of image classification. The three main areas that image classification has advanced the most are: using multiple remote-sensing features; using ancillary data in the classification procedures; and development and use of advanced classification algorithms, like subpixel, knowledge-based classification, and per-field. The most common technique used to improve accuracy of image classification is accuracy assessment based on error matrix. This is commonly used for evaluating per-pixel classification. Lastly, the success depends on many factors, such as the availability of high-quality remotely sensed imagery and data.

   D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," International Journal of Remote Sensing, vol. 28, no. 5, pp. 823–870, 2007.

3. 3D object representations for fine-grained categorization

   An important and growing sub-field of computer vision and object detection is fine-grained recognition. The difficulty with this problem lies in capturing minute differences in a target object despite varying object pose. A new and successful route to try and accomplish this is representing targets in 3D object space rather than a 2D plane, as this allows both the critical features of a target and their relative locations to be represented and and available [3]. A new standard for object recognition performance has been set by raising two state-of-the-art 2D object representation methods (Bubblebank and Spatial Pyramid) to 3D. While our objective in this project focuses more so on image classification than individual object recognition, we can view this problem fine-grained recognition as a possible methodology for successfully classifying images by recognizing shared objects within them.

   J. Krause, M. Stark, J. Deng, and L. Fei-Fei, "3D object representations for fine-grained categorization," 2013 IEEE International Conference on Computer Vision Workshops, 2013.

4. Textural features for Image Classification

   The mathematical techniques for image classification are essential for effective model design. Haralick, Shanmugam and Dinstein investigate, methods for extracting the textural features for image classification. Aerial photographs and satellite images in particular benefit from textural feature extraction. The paper outlines as a procedure for identifying textural features based on the gray-tone spatial dependencies and then applies these techniques in an image detection model classifying photomicrographs of sandstone. Additionally they use these methods to classify panoramic aerial photographs of eight land-use categories. Digital images are usually stored in two dimensional array where each element is assigned some corresponding gray tone value. Textural features contain information about the spatial distribution of tonal variations within a band. Using gray-tone textural

features two decision rules were used to classify images. The first was a piece-wise linear decision rule and the second was a min-max decision rule. These methods produced model accuracies of 82 percent for aerial images and 89 percent for photomicrographs.

R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for Image Classification," IEEE Transactions on Systems, Man, and Cybernetics, vol. SMC-3, no. 6, pp. 610–621, 1973.

3 Experimental methods

3.1 Data pre-processing

Each image in the repository is a 150 by 150 image with three color channels. To work with these images we need to represent them as 150 by 150 by 3 numpy arrays. Additionally, we can reshape these arrays as one dimensional vectors to construct a feature matrix. We will perform principal component analysis for dimension reduction to decrease computational costs and combat overfitting.

3.2 CNN

A key model we would like to implement in our model is a convolutional neural network. Unlike other statistical learning methods used in this project, the convolutional neural network does not require us to flatten the (150 x 150 x 3) image arrays into one dimensional vectors for feature learning. Instead the convolutional will take in a multi-channel input and then does the feature learning in its convolution and pooling layers. In addition to our novel CNN model, we will develop a CNN to be trained on ImageNet to transfer its learned representations to our problem.

The specific network architecture, hyperparameters, and activation functions we will use are still in the works but the general procedure for building and testing these models is as follows:

1. Load images as (150,150,3) dimension numpy arrays
2. Using tensorflow and keras python libraries build a convolutional neural network that takes a 3d input and outputs a softmax output for classification.
3. Use a confusion matrix, precision and recall, and F-1 scores as metrics to evaluate model performance.
4. Write functions to for k-validation tests for best performance using different activation functions, stride lengths, kennel sizes, and padding.
5. Test model on segmented test data set can compare to the other modes we built.

3.3 SVM

We plan to implement a Support Vector Machine (SVM) to provide us with a baseline idea of performance on our image classification with a non deep-learning approach. SVM requires all images to be resized to a fixed size before being inputted into the SVM. We will test the model with the test data sett and evaluate the classification results using the sklearn model evaluation methods.

4 Experiments

4.1 Dataset

We source the dataset of images of natural spaces from kaggle.com.

`https://www.kaggle.com/datasets/puneet6060/intel-image-classification`

The original dataset was provided by Intel to `https://datahack.analyticsvidhya.com` for a Image classification Challenge. The data set consists of around 25,000 images of size 150 by 150 pixels distributed into six classes. The images are classified as either a building, forest, glacier, mountain, sea, or a street. Furthermore, the data set comes pre-segmented into balanced training, test, and prediction subsets. Where the training set contains roughly 14k images with each class containing between 2000 and 2500 images.