Applying Image Feature Extraction to Cluttered Scientific Repositories

Emily Herron
Illinois Institute of Technology
Mercer University
Emily.Joyce.Herron@live.mercer.edu

Tyler J. Skluzacek, Ian Foster, and Kyle Chard (Advisors)
Computation Institute,
University of Chicago and Argonne National Laboratory
{skluzacek, foster, chard}@uchicago.edu

***Abstract*—Over time many scientific repositories and file systems become disorganized, containing poorly described and error-ridden data. As a result, it is often difficult for researchers to discover crucial data. In this poster we present a collection of image processing modules that collectively extract metadata from a variety of image formats. We implement these modules in Skluma—a system designed to automatically extract metadata from structured and semi-structured scientific formats. Our modules apply several image metadata extraction techniques that include processing file system metadata, header information, color content statistics, extracted text, feature-based clusters, and predicting tags using a supervised learning model. Our goal is to collect a large number of metadata that may then be used to organize, understand, and analyze data stored in a repository.**

I. Introduction

There are many examples of well-organized, and indeed valuable, scientific data repositories. However, over time many repositories become disorganized and the lack of descriptive metadata ultimately hinders data discovery and use [1]. Vast reservoirs of scientific data accumulated over the course of decades are concealed by obscurely-named directories and files, buried among irrelevant files, and hidden due to a lack of metadata.

To address this problem, Skluma implements an automated pipeline for making sense of large collections of scientific data [1]. Skluma crawls a repository or file system, extracts metadata from files, establishes relationships between files and ultimately assembles a probabilistic “ball” of metadata about each file. In this poster we present a collection of Skluma modules for extracting image metadata (e.g., resolution, dimensions, etc), image classes (e.g., map or plot), and text embedded within images.

We evaluate our pipeline on the US Department of Energy’s Carbon Dioxide information and Analysis Center’s (CDIAC) repository [2]. CDIAC contains tens of thousands of image files amongst its more than 500 thousand scientific files that cumulatively exceed 500 GB.

II. Methodology

Our metadata extraction pipeline is shown in Figure 1. For each image file, our pipeline employs a series of modules to collect metadata from file system metadata, image information, and text strings embedded within images. Our pipeline also applies methods to cluster images based on extracted features and to tag image content using a supervised learning model. The result of our pipeline is a collection of metadata (in JSON format) for each image.

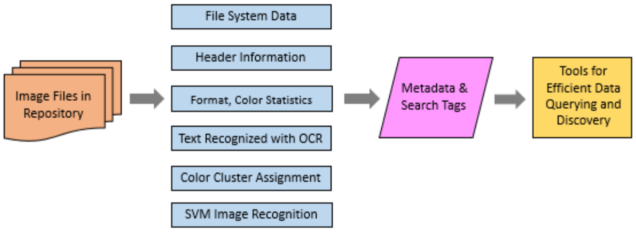

Fig. 1. Skluma image metadata extraction pipeline.

File System Metadata: The first module crawls a repository to locate image files (based on file extension). It then collects file system metadata (e.g., name, path, extension, and size) and tokenizes extracted strings into discrete tags by separating on common characters (e.g., underscores, spaces, and digits).

Image Metadata: The second module extracts structured metadata from within image formats using the Python Image Library (PIL). While header contents vary depending on image format, they commonly include details such as format, dimensions, encoding, creation data, and software.

Image Feature Clusters: The next module clusters images by their features—an important step in determining image similarity and for linking images with the aim of forming associations between common metadata. We utilize features derived from the previous modules (e.g., mean, median, and extrema of image color).

Our module first resizes all image samples in RGB or RGBA mode and divides them into a 4 by 4 grid. Features are calculated for each grid section and are appended to a feature vector. The set of resulting feature vectors are assigned to clusters using K-Means clustering. The group is divided into a specified number, k, groups of equal variance, where each cluster is described by a centroid or mean of each sample in the cluster. We reduce the dimensions of the feature vectors by projection onto a 2 dimensional space using principal component analysis (PCA).

Supervised Classification: The classification model aims to

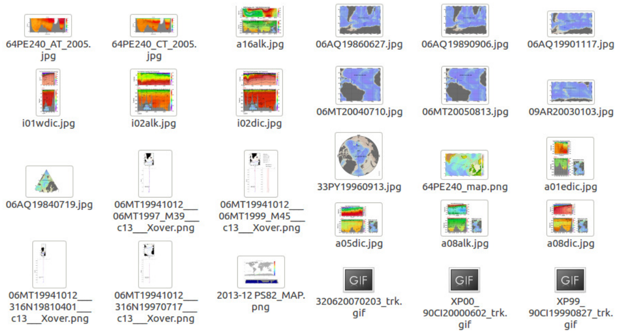

Fig. 2. Sample image clusters resulting from applying K-Means clustering where k = 5.

categorize images into general classes. For this purpose we use a support vector machine (SVM) model. To train this model we labeled a collection of 300 sample images with one of five classes: map, map/depth chart, map/histogram, map/plot, or other.

Our module first resizes images to standard dimensions and converts to grayscale arrays using PCA to reduce dimensions. We trained a SVM classification model using scikit-learn's c-support classification model (SVC) function.

Text Extraction: Images often contain text such as titles, locations, axes labels, and descriptions. To extract this information our final module applies optical character recognition (OCR) techniques using Python-tesseract (a wrapper for Google Tesseract-OCR [3]). To improve accuracy, we first convert images to grayscale.

III. Evaluation

We evaluated our pipeline by testing it on images contained in the CDIAC repository. Given space constraints we present only evaluation of our clustering and SVM-based classification modules.

Clustering: To determine an optimal value for k, we applied silhouette analysis to the PCA-reduced clusters. Silhouette analysis measures the average distance between neighboring clusters. Using the optimal value of $k = 4$, Figure 3 illustrates the resulting clusters and associated silhouette scores. Figure 2 shows examples of images in each of the generated clusters.

Classification: To evaluate our SVM classifier we split our manually tagged images into a training and test set. Table I shows high precision for all classes, and high recall for all but the map and plot class. We applied our model to more than 3500 images in CDIAC and report the number of images in each class.

IV. Summary

Our image feature extraction pipeline is able to extract a variety of useful metadata from many different image formats. In future work, we are investigating the use of convolutional neural networks (CNNs) to recognize physical objects, match maps to their spatial coordinates, and interpret scientific charts in image files.

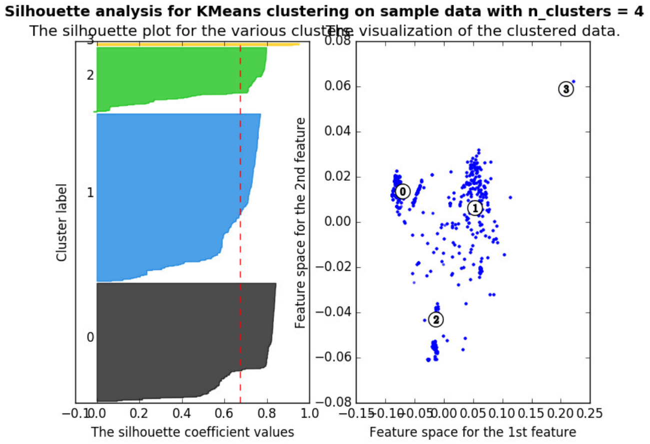

Fig. 3. Silhouette analysis for K-Means clusters where k = 4.

Class Label	Precision	Recall	f1 Score	Support	Total
map	0.94	0.95	0.95	88	2877
map&depth_chart	1.00	1.00	1.00	99	22
map&plot	1.00	0.50	0.67	6	395
map&histogram	1.00	0.43	0.60	7	144
other	0.00	0.00	0.00	1	200
avg / total	0.97	0.94	0.95	201	3638

TABLE I

Precision, recall, F-measure, and support scores for 2:1 ratio split of test and training set. We also include the total for each class in the CDIAC dataset

References

[1] P. Beckman, T. J. Skluzacek, K. Chard, and I. Foster, "Skluma: A statistical learning pipeline for taming unkempt data repositories," in *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, SSDBM '17, (New York, NY, USA), pp. 41:1–41:4, ACM, 2017.

[2] U.S. Department of Energy, "Carbon dioxide information and analysis center," 2017.

[3] S. Hoffstaetter, J. Bochi, M. Lee, and L. Kistner, "pytesseract 0.1.7.," 2017.