Pipeline for Image Metadata Extraction and Contextualization in Large Scientific Data Repositories

Emily Herron, Kyle Chard, Tyler Skluzacek, Ian Foster
Mercer University; Computation Institute, University of Chicago

Introduction

- Poor organization and clutter in large scientific repositories complicates data discovery and analysis
- Automated tool Skluma processes extracting metadata and inferring contextual relationships between files in such repositories, , converting them into indexed, searchable collections.
- Metadata extraction from image files is complex problem not yet addressed by Skluma: image formats vary, represented by one or more matrices of pixel color values and text, and objects within images must be recognized or predicted
- We present a module for Skluma containing methods for extracting and processing feature and content-based metadata from image files in large data repositories.

Image Metadata Extraction Pipeline

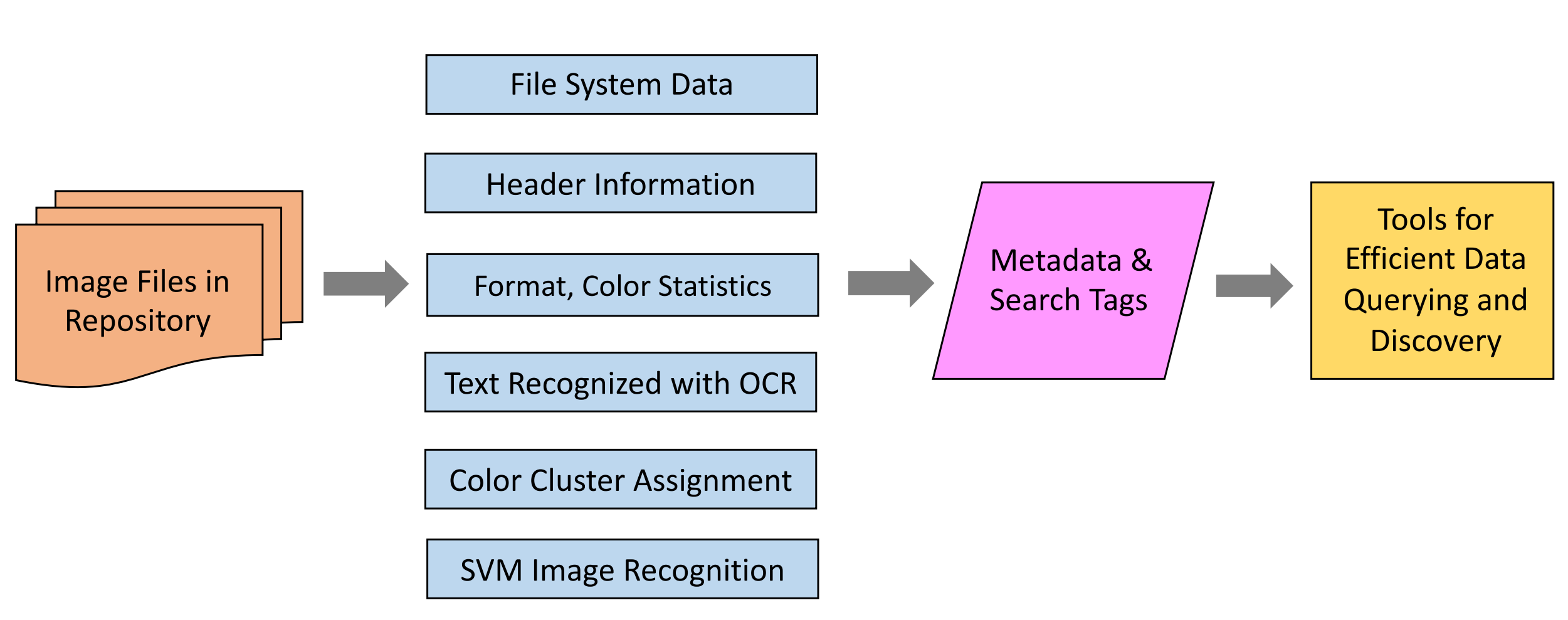


Image Files in Repository

File System Data
Header Information
Format, Color Statistics
Text Recognized with OCR
Color Cluster Assignment
SVM Image Recognition

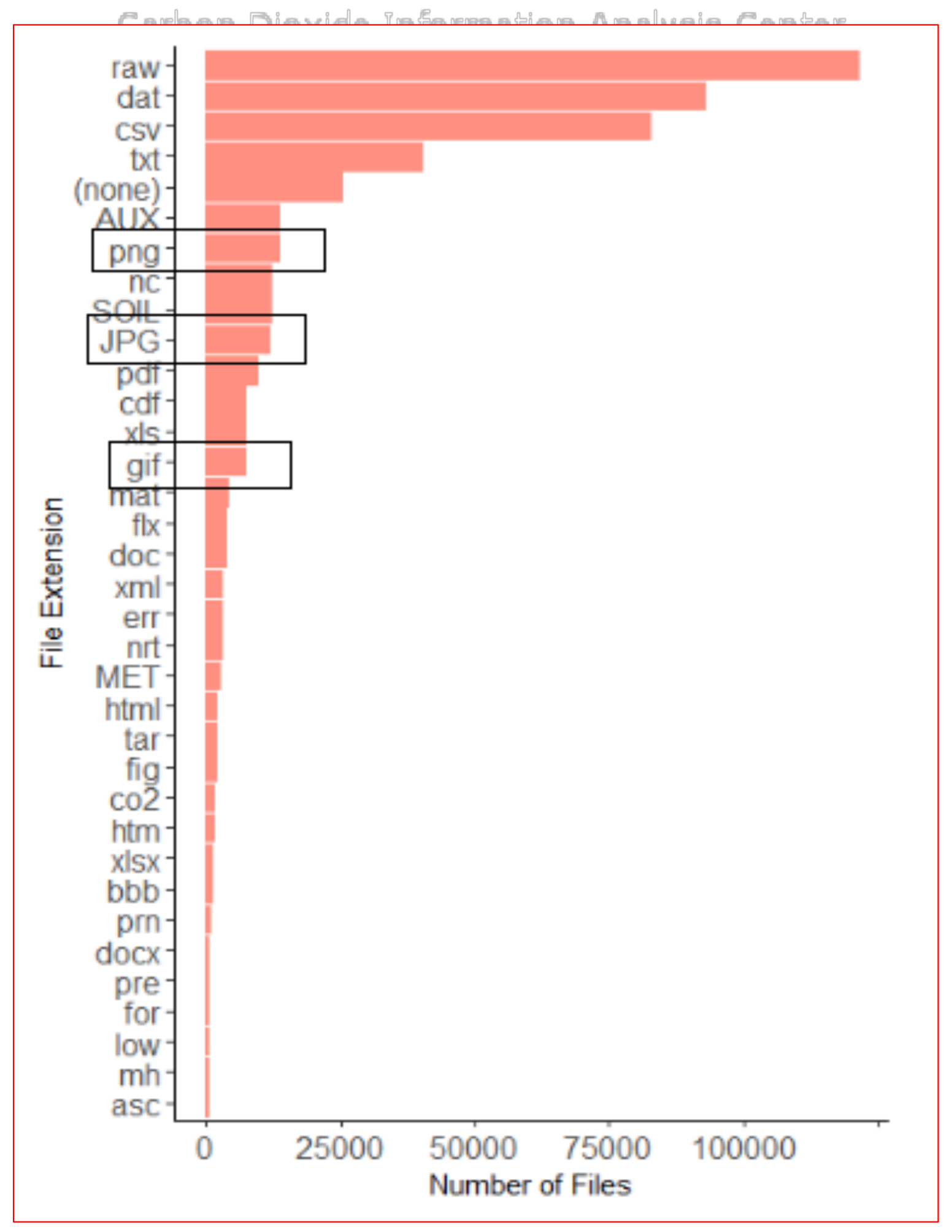
Metadata & Search Tags

Tools for Efficient Data Querying and Discovery

- Image feature extraction module designed to crawl repository, locating all files with image extensions
- Metadata collected for each image file from file system data, image headers, text and content recognized using optical character recognition (OCR) and a supervised learning model, and color and feature-based cluster assignments
- Result: metadata strings and search tags useful for organizing and querying images in scientific repositories by content and topic.

Preparing a Sample Image Repository

- US Department of Energy's Carbon Dioxide Information and Analysis Center's climate data repository used to develop and test metadata extraction module.
- Script used to crawl repository and download image files including PNGs, JPEGs, and GIFs (among CDIAC's 35 most common file types) located and downloaded via file transfer protocol and run through pipeline.

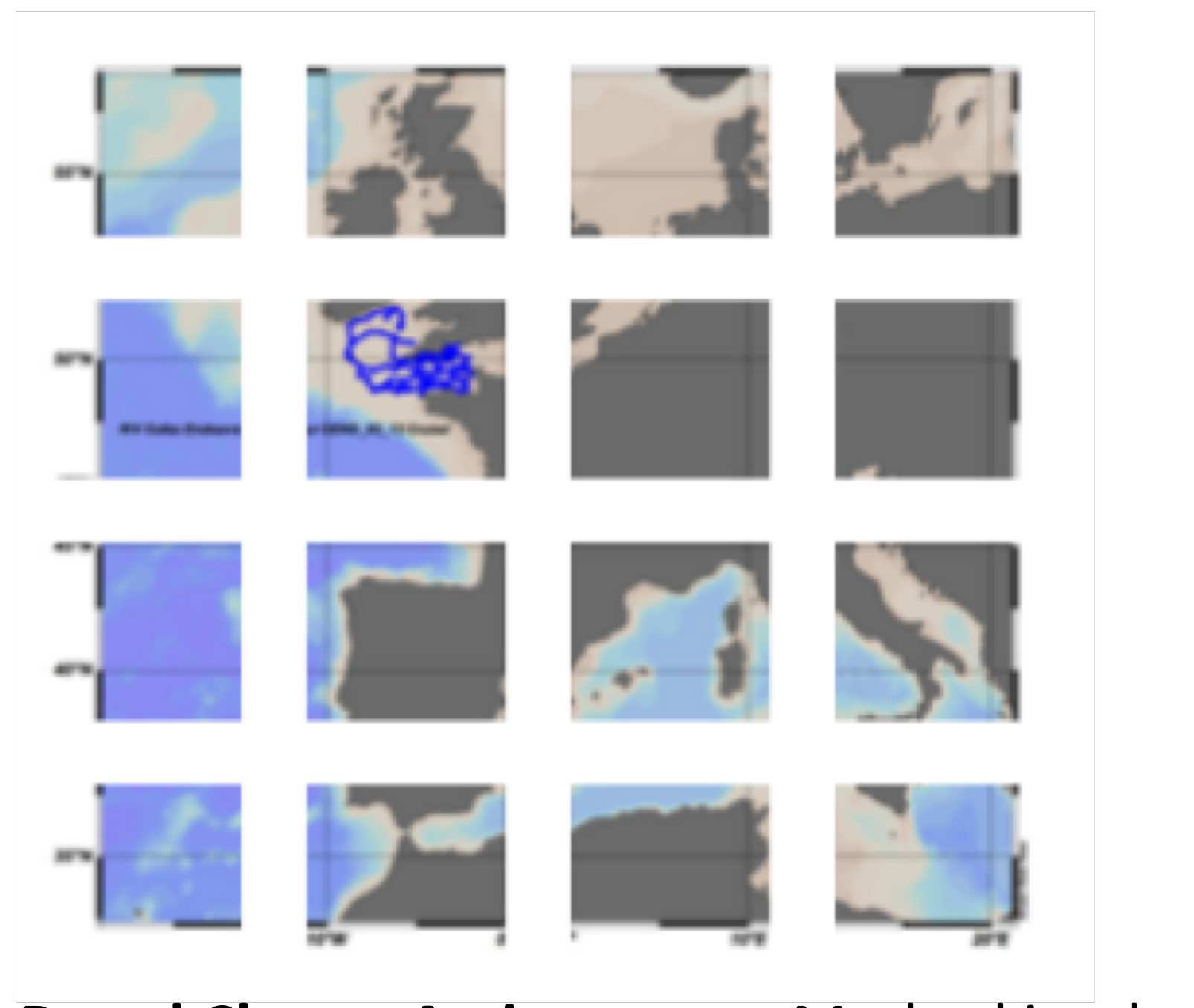


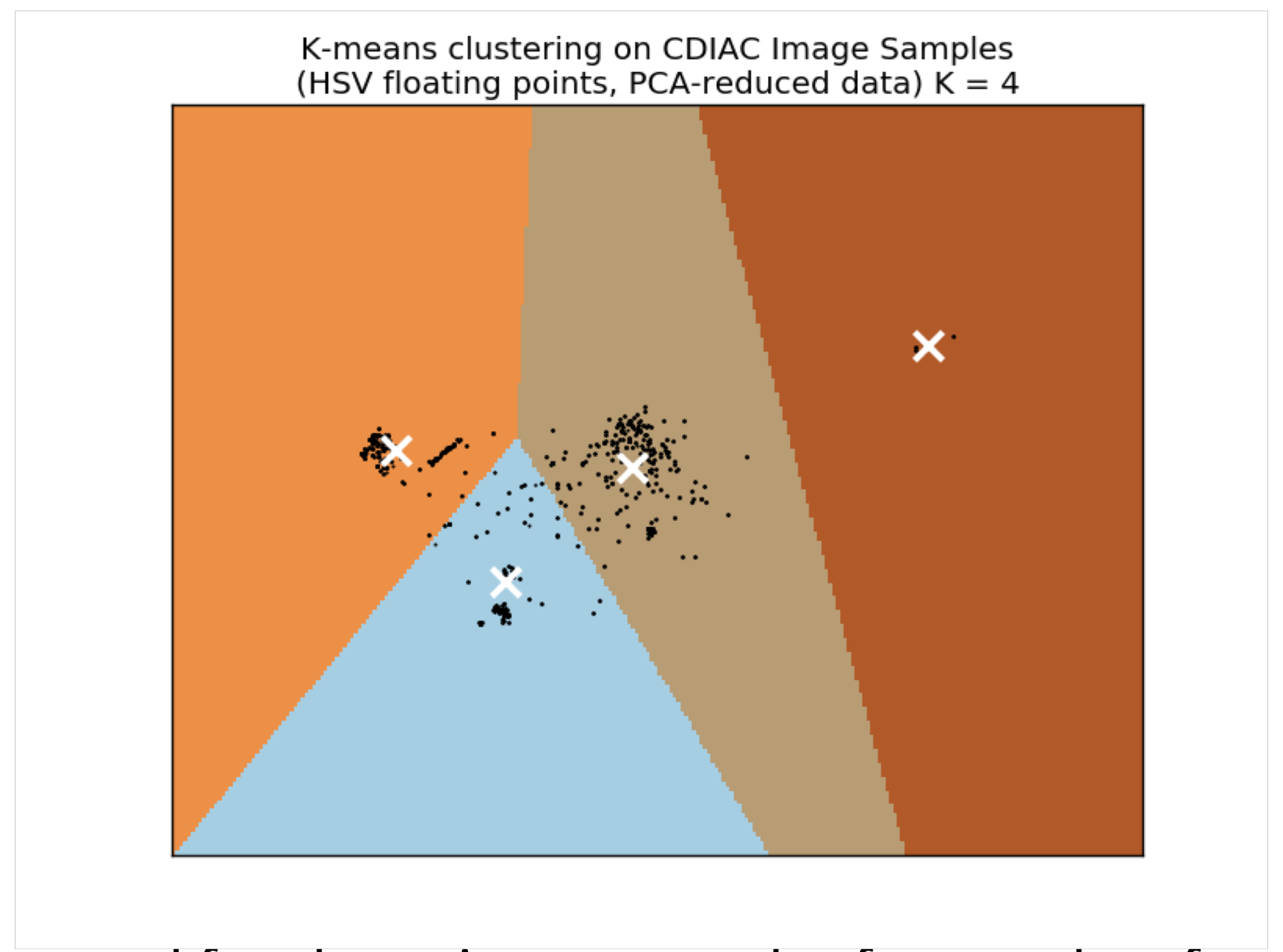
File System Data and Header Information

File System Data: Collected using Python's OS library. Included file system file names, paths, extensions, sizes. Separate function implemented to parse file names into searchable tags and keywords by separating file names into numbers, dates, character strings, and underscores.

Image Header Information: Accessed using the Python Image Library (PIL). Often includes details on image mode or format, resolution, encoding details, creation date, and software.

Clustering with K-Means

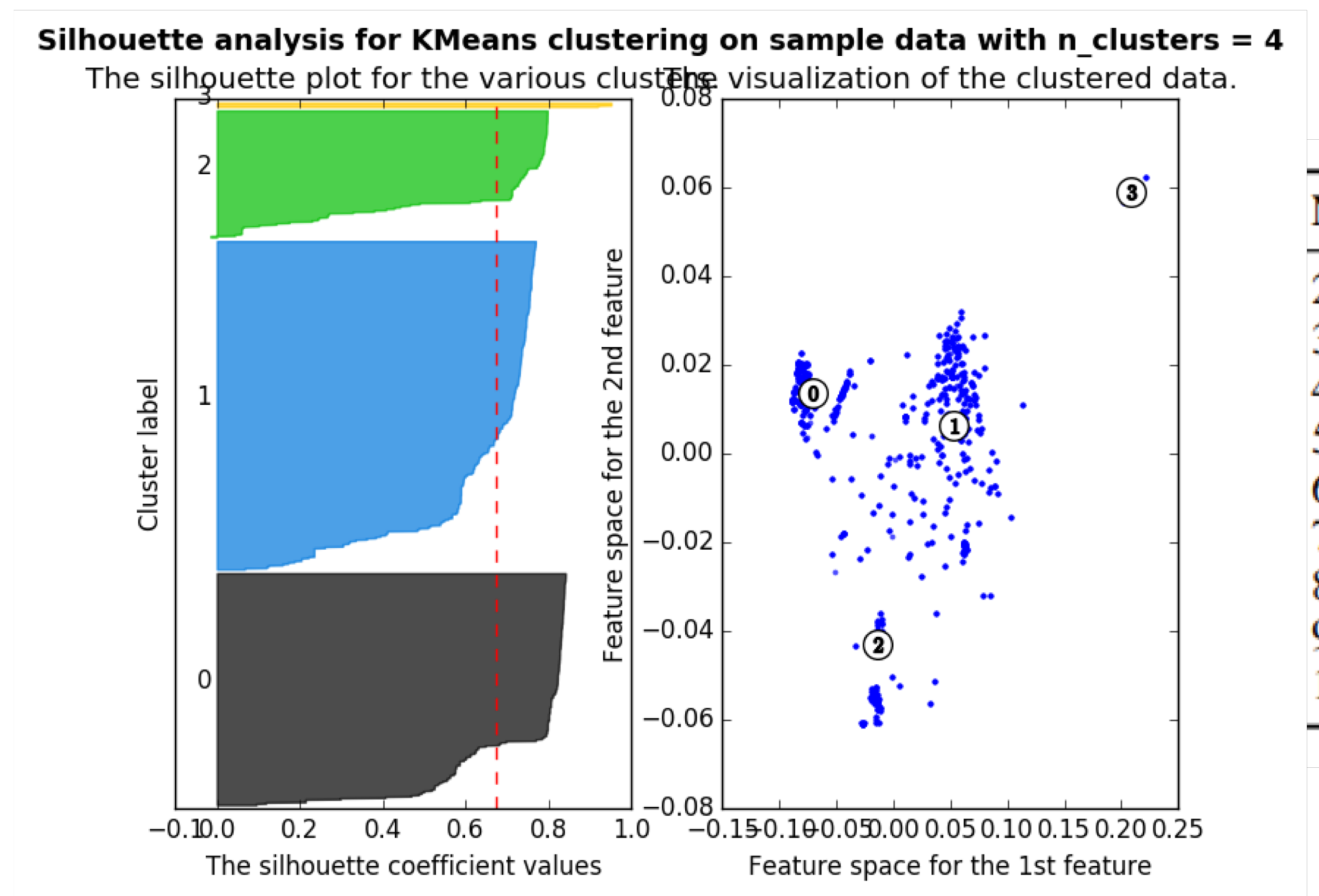



K-means clustering on CDIAC Image Samples
(HSV floating points, PCA-reduced data) K = 4

Color-Based Cluster Assignments: Method implemented for clustering mean color feature data from test images using K-Means, returning numerical cluster assignment for each image

- RGB and RGBA mode images resized, divided into 4 by 4 grids
- Mean floating point RGB values calculated grid sections, appended to feature vectors
- PCA-reduced feature vectors clustered using Scikit-Learn's K-Means clustering function.

Evaluating K-Means Clusters with Silhouette Analysis



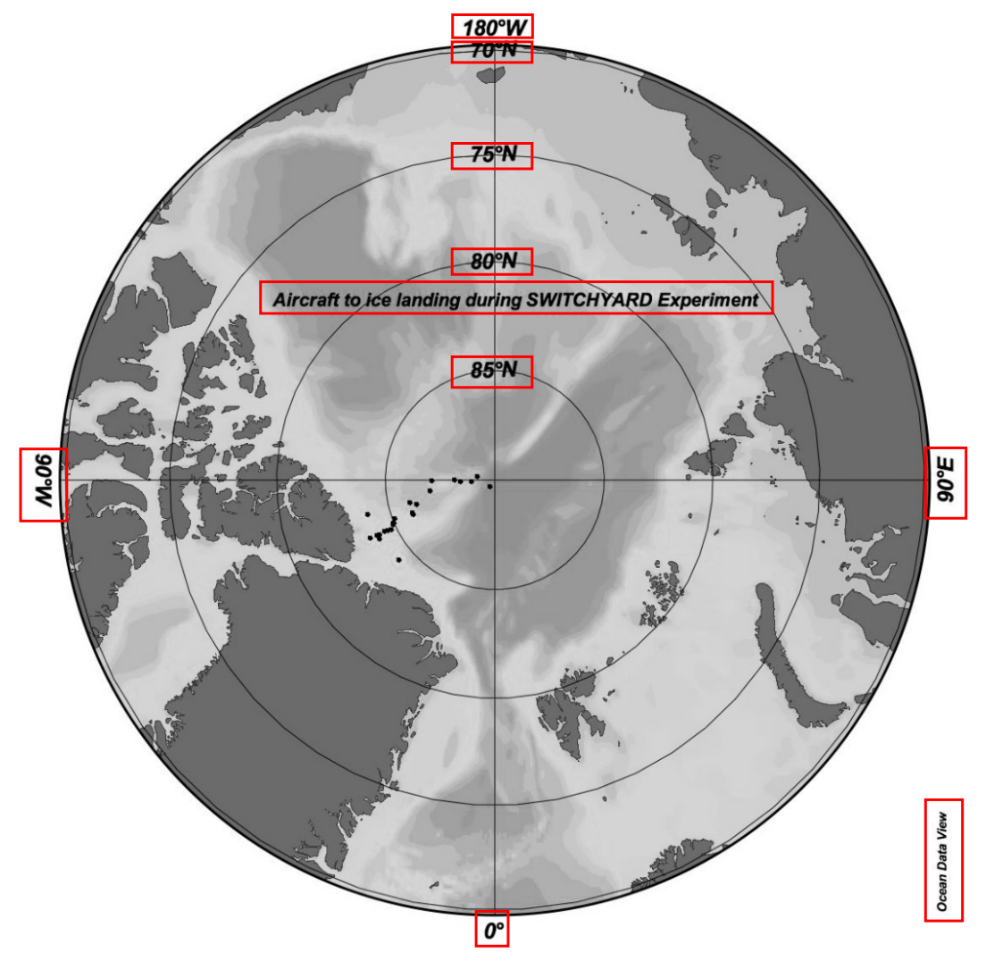
Number of KMeans Clusters, K	Mean Silhouette Score
2	0.62056
3	0.65953
4	0.67219
5	0.60345
6	0.59570
7	0.61031
8	0.59449
9	0.59441
10	0.58789

TABLE I

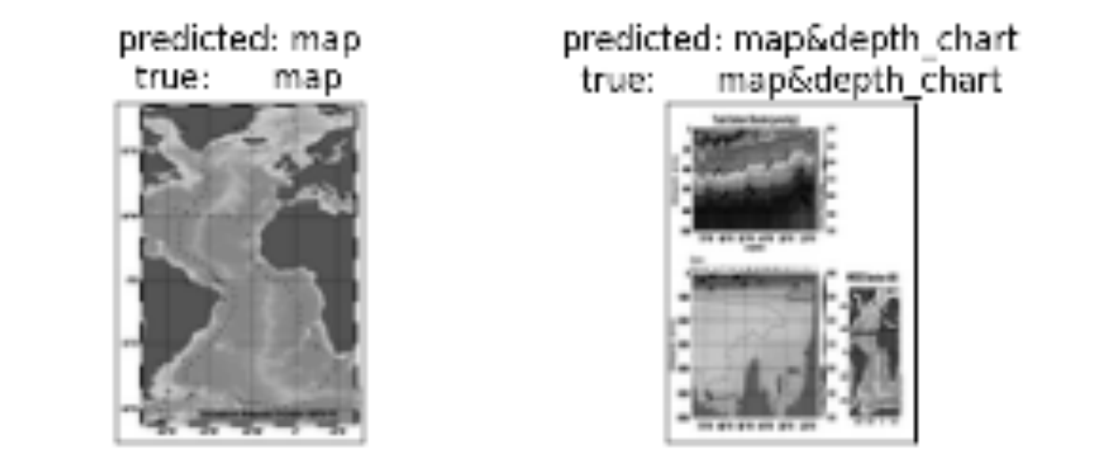
- Silhouette analysis used to select optimal number of clusters for mean color-based cluster assignments.
  - Measures average distances between neighboring clusters, scores ranging from -1 to 1, with 1 indicating clusters farther away from adjacent clusters
- Optimal k value of 4 selected based on the high mean silhouette score

Image Text Recognition

Text extracted from images using Python-tesseract's image to text function, an optical character recognition tool and wrapper for Google's Tesseract-OCR engine. Images are converted to grayscale with PIL and passed to the OCR function. Recognized text is returned as unicode strings which are parsed into metadata strings useful for searching images by text content. E.g. maps, longitude, latitude values in maps.



Training an SVM Classification Model



Class Label	Number Predicted in CDIAC
map	2877
map&depth_chart	22
map&plot	395
map&histogram	144
other	200
total	3638

TABLE IV

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

TABLE II

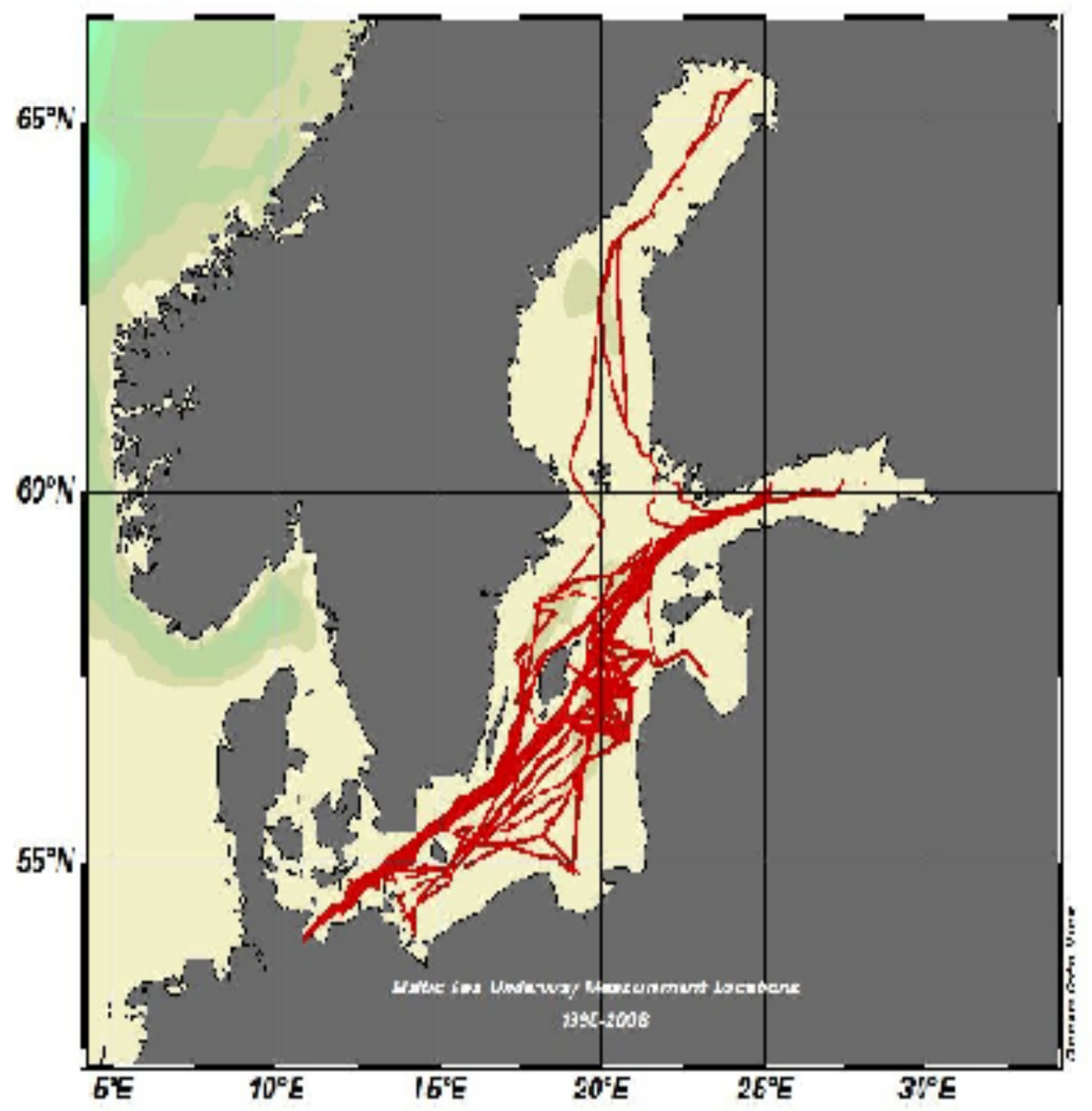
Class Label	Precision	Recall	f1 Score	Support
map	1.00	0.98	0.99	44
map&depth_chart	1.00	1.00	1.00	51
map&plot	1.00	1.00	1.00	2
map&histogram	0.75	1.00	0.86	3
avg / total	0.99	0.99	0.99	100

TABLE III

Support vector machine (SVM) model trained to classify and predict content of 3638 images transferred from CDIAC (see Table IV)

- RGB mode images given class assignment: identified by number, 0-4.
- Images, resized, converted to grayscale arrays
- Image arrays with class numbers randomized, split into separate arrays of image arrays and labels, and split into training and validation arrays.
- Training and validation image arrays dimensions reduced with PCA, used to train SVM model using Scikit-Learn's c-support classification model (SVC) function.
- Accuracy validated by comparing validation set labels predicted by the model to set true labels.
- Precision, recall, F-measure, support scores measured for 2:1 and 1:2 ratio split of test and training sets (see tables II & III).

Example Metadata String



```
{
'header': {
        'header_info': {
        'jfif': 257,
        'jfif_unit': 1,
        'exif': 'Exif\x00\x00MM\x00*\...',
        'jfif_version': (1, 1),
        'jfif_density': (96, 96),
        'dpi': (96, 96)},
    'image_mode': 'RGB',
    'dimensions': '930x825'},
  'color': {
        'mean_pixel_value': 144.818...,
        'extrema': ((0, 255), (0, 255), (0, 255)),
        'mode_pixel_value': 107.0,
        'median_pixel_value': 107.0,
        'std_dev_pixel_value': 69.529...},
   'system': {
        'path': '/media/emily/USB DISK1/images',
        'extension': 'jpg',
        'file': 'BS_Underway_Map.jpg',
        'size': 115811},
   'image_text': ['Baltic Sea Underway Measurement Locations', '1995-2008'],
   'name_tags': ['BSUnderwayMap'],
   'SVM_class_tags': ['map'],
   'mean_colors_cluster': 1}
```

Sample JSON string containing metadata extracted from BS_Underway_Map.jpg. Includes PIL header information, color statistics, file system data, text extracted from image, search key words extracted from title, tags from SVM classification, and color-based cluster assignments.

References

[1] P. Beckman, T. J. Skluzacek, K., Chard, I. Foster. Skluma: A Statistical Learning Pipeline for Taming Unkempt Data Repositories. Computation Institute, University of Chicago and Argonne National Laboratory, Chicago, IL. 2017.

[2] S. Hoffstaetter, et al. "pytesseract 0.1.7." Python Software Foundation. 1990-2017. https://pypi.python.org/pypi/pytesseract

[3] "Carbon Dioxide Information and Analysis Center." U.S. Department of Energy. Oak Ridge National Laboratory. 2017. ftp://cdiac.ornl.gov

Acknowledgements

This work is supported in part by the National Science Foundation grant NSF- 1461260 (BigDataX REU).