

OREGON STATE UNIVERSITY

CS461 FALL 2017

GROUP 19

Aerolyzer Python Library Software Requirement Specification

Authors

Kin-Ho LAM

Logan WINGARD

Daniel ROSS

Supervisor

Dr. Kim WHITEHALL

Abstract

Monitoring atmospheric aerosols ¹ is important due to their effects on the atmosphere's chemical composition and radiation distribution. The presence of aerosols reduces air quality which can potentially lead to health complications such as bronchitis or respiratory inflammation. Unfortunately, existing methods to gather aerosol data such as satellites, planes, and ground-based instruments provide data that are too complex to be useful for the average person to understand or do not provide data fast enough.

Aerolyzer is a web application that uses weather information and acceptable images ² of the horizon to infer local atmospheric phenomena in the United States. The following document details the software requirement specifications for Aerolyzer Aerosol Detection API. The Aerosol Detection API will consist of python classifiers that shall analyze user submitted images, and remove unacceptable images.

10/25/17

I. TABLE OF CONTENTS

CONTENTS

I	Table of Contents	1
II	Introduction	2
II-A	Purpose	2
II-B	Scope	2
II-C	Definitions	2
II-D	Overview	2
III	Project Overview	3
III-A	Project Perspective	3
III-B	Project Functions	3
IV	Users & Functionality	4
IV-A	User Stories	4
IV-B	Expected user characteristics	4
IV-C	Input Constraints	4
IV-D	Assumptions and Dependencies	4
IV-E	External Interfaces	4
V	Functions	5
V-A	Classify photo	5
V-B	Call Weather API	5
V-C	Identify Aerosols	5
V-D	Extract EXIF	5
V-E	Web Application Interface	6
V-F	ZIP Code Submission	6
V-G	Upload Photo	6
V-H	Display data	6
VI	Performance	7
VI-A	Performance Requirements	7
VI-B	Stretch Goals	7
VI-C	Tentative Development Schedule	8

II. INTRODUCTION

A. Purpose

The Aeroalyzer Project serves as a new tool to infer local air quality using regional weather data and aerosol analysis. Aeroalyzer shall crowd-source images of the horizon to characterize aerosol 1 content in the atmosphere. Achieving this functionality calls for the creation of a Python library 8 that identifies and analyzes acceptable images 2, stores relevant information for trend analysis, and compiles weather information with aerosol data.

B. Scope

Bringing the Aeroalyzer Project to a mainstream audience calls for an open-source back-end Python API to perform image analysis. The Python library shall use image classification algorithms to characterize acceptable images. The Aeroalyzer project shall exclusively serve users in the United States. If time permits, the existing front-end UI shall be modified to present the Aeroalyzer library's weather and air quality inference based on analyzed images or by a requested location. The UI shall also be modified to collect user-submitted cell phone images of the horizon 3, sunset 5, or sunrise 4 to be fed to the library for analysis.

C. Definitions

- 1) Aerosol: Aerosols are tiny particles dispersed throughout the atmosphere. These particles originate from natural sources such as volcanic eruptions, and unnatural sources such as pollution. One can visibly see the effects of aerosols in the 'Rayleigh scattering effect' (2) which visibly reddens sunsets and sunrises.
- 2) Acceptable Image: An acceptable image is defined as an unedited image of the horizon 3, sunset 5, or sunrise 4. Images used in data collection must have valid EXIF meta data, while the images used in training classifiers only require relevant image content.
- 3) Horizon: The horizon is defined as the line at which the sky and earth's surface appear to meet.
- 4) Sunrise: A sunrise is defined as the colors and light visible in the sky produced by the sun's first appearance in the morning.
- 5) Sunset: A sunset is defined as the colors and light visible in the sky produced when the sun disappears in the evening.
- 6) EXIF: Exchangeable image file format.
- 7) Local: Local is defined as the circular area around an observer with the radius being the distance from an observer's position to the horizon. For an observer on the ground, this distance is approximately 2.9 miles (4.7 km) and for an observer standing at an elevation of 100 ft (30 m) above ground level this distance is approximately 12.2 miles (19.6 km).
- 8) Library: A library is defined a Python module. The terms "Python module" and "Python library" are used interchangeably.

REFERENCES

- [1] Wunderground documentation.
- [2] Bob Allen. Atmospheric aerosols: What are they, and why are they so important?, Apr 2015.

D. Overview

The following Project Overview details the functionality of the Aeroalyzer Library. This functionality shall be explored through its planned work flow, expected user behavior, and general implementation in a larger system. Finally, the Performance section details the expected state of the project when completed by the 2017-2018 team.

III. PROJECT OVERVIEW

A. Project Perspective

Aerolyzer is a web-based tool that provides weather and air quality information based on zip codes and submitted images. A back-end Python Library 8 shall provide the Aerolyzer project's functionality. Functions in the library shall take images of the horizon, sunrise, sunset or a user's ZIP code as input. Image classifiers shall be developed using tools such as OpenCV or Google Tensorflow. Human-selected acceptable images from public sources shall be used to train and verify the accuracy of the classifiers.

B. Project Functions

Aerolyzer's web interface shall take a user's location, either from a submitted image's EXIF 6 data or a provided ZIP code, as the primary input. If a user submits an acceptable image, the results of color analysis on that image are add to the application's data for their ZIP code. The library then calls on weather API for meteorological data for their ZIP code. The application then displays all data relevant to the user's ZIP code on their screen in an understandable format.

IV. USERS & FUNCTIONALITY

A. User Stories

Users shall either know their current ZIP code or have a mobile device that stores their location in an image's EXIF 6 data.

B. Expected user characteristics

- 1) The user is in the United States.
- 2) The user owns a smart phone or knows their ZIP code.
- 3) The user has a general idea of what aerosols are.
- 4) The user understands how to use their mobile device to capture an image of the horizon.
- 5) The user has a basic understanding of how to navigate a web browser in order to use the web application.
- 6) The user understands basic weather terminology such as temperature, wind speed, and humidity.

C. Input Constraints

- 1) The image must be taken on a device with location tracking enabled.
- 2) The image must be unedited and have no filters applied.
- 3) The image must have EXIF data.
- 4) The image must contain a horizon.
- 5) The image must be a sunset or sunrise.
- 6) The image must be taken from a location inside the United States.

D. Assumptions and Dependencies

- 1) Users submitting photos have given permission for having their image stored and used in further processing.
- 2) The user is connected to the Internet.
- 3) The user's web browser supports the application's interface.

E. External Interfaces

- 1) The library shall make external calls to the Weather Underground API (1) for current weather conditions at a certain ZIP code.
- 2) The library shall facilitate the storage of images onto an external database.
- 3) The library shall enable classifier task parallelization.

V. FUNCTIONS

A. *Classify photo*

Trigger: A new, unclassified photo has been added to the central database.

Precondition: The image and its JSON meta data are accessible to the classifier per use case Extract EXIF V-D.

Basic Path:

- 1) The image's latitude and longitude are read from the JSON meta data.
- 2) The latitude and longitude are used to identify the image's ZIP code.
- 3) The image is checked with the horizon classifier to confirm the presence of a horizon in the image.
- 4) If a horizon is confirmed, the image is checked using a sunset/sunrise classifier to confirm the presence of a sunset or sunrise.
- 5) The image is stored in the database with its content and ZIP code identifying it.

Alternative Paths: The image is checked with the horizon classifier and the image isn't confirmed to have the horizon in it. The user is notified, the image isn't stored. If case Extract EXIF V-D is properly performed on an image and a ZIP code is extracted, then the UI transitions to case Call Weather API V-B

If ZIP code cannot be retrieved from the EXIF data but the image is classified as an acceptable image, the use case transitions to Identify Aerosols V-C.

Postcondition: The user's image is stored and categorized for further use.

Exception Paths: The latitude and longitude in the image JSON data is not in the United States, the user is notified and the use case abandoned.

B. *Call Weather API*

Trigger: The user's ZIP code is received.

Precondition: The user submits their ZIP code per use case ZIP Code Submission V-F or the user has submitted an image that was successfully identified as per use case Classify Photo V-A.

Basic Path:

- 1) A request is made to the weather API for data on the received ZIP code.
- 2) The data received from the weather API is stored in the central database under the ZIP code it pertains to.

Alternative Paths: None

Postcondition: The central database stores weather data for the user's ZIP code.

Exception Paths: If data is unable to be retrieved from the weather API, an error message will be displayed and the use case abandoned.

C. *Identify Aerosols*

Trigger: The user's ZIP code and their submitted image is received.

Precondition: The user submitted an image that was successfully identified as per use case Call Weather API V-B.

Basic Path:

- 1) The colors in the image are analyzed.
- 2) The analysis returns the type of aerosol most likely to cause the colors in the image.
- 3) The type of aerosol and the current time are stored in the central database under the user's ZIP code.

Alternative Paths: None

Postcondition: The central database has a type of aerosol at a particular time for a certain ZIP code.

Exception Paths: None

D. *Extract EXIF*

Trigger: An image is passed to the EXIF function

Precondition: Photo has been successfully uploaded per use case Upload Photo V-G.

Basic Path:

- 1) Aerolyzer library receives image as input for extraction script.
- 2) Script outputs the image's EXIF data as a JSON file.
- 3) The JSON file is stored with the image prior to classification.

Alternative Paths: Images and their associated meta data can be added manually to the database.

Postcondition: The central database has access to the image and its meta data.

Exception Paths: None

E. Web Application Interface

Trigger: User navigates to the Application site using a Web Browser

Precondition: User has access to a web browser and the Internet.

Basic Path:

- 1) User opens web browser
- 2) User navigates to the Aerolyzer URL in the web browser
- 3) System serves start page to the user

Alternative Paths: Users can navigate to the Aerolyzer start page via a link

Postcondition: User has access to the Aerolyzer web interface.

Exception Paths: If the user isn't connected to the Internet or their web browser can't access the page, the use case is abandoned.

F. ZIP Code Submission

Trigger: User clicks submit on form for ZIP code

Precondition: User's web browser correctly displays the form for ZIP input.

Basic Path:

- 1) User types ZIP code in form
- 2) User clicks "Submit" button on form

Alternative Paths: Case Upload Photo V-G

Postcondition: User is navigated to output page.

Exception Paths: If the submitted ZIP code isn't a valid 5 digit ZIP, the user is notified and the use case is abandoned.

G. Upload Photo

Trigger: User clicks the 'Upload a Photo button

Precondition: Users have a picture from a mobile device and their web browser has access to this photo.

Basic Path:

- 1) User clicks the "Upload Photo" button on page.
- 2) User is prompted to select a photo from their device's storage.

Alternative Paths: Case ZIP Code Submission V-F

Postcondition: The image is passed to the server

Exception Paths: If some connection error or other issue occurs, the user is informed and the case is abandoned.

H. Display data

Trigger: Use case Identify Aerosols V-C is completed.

Precondition: The central database has completed any updates in progress on the user's requested ZIP code.

Basic Path:

- 1) All of the most recent data for the user's ZIP code is fetched from the central database.

Alternative Paths: None

Postcondition: The user is taken to a page to display the fetched data.

Exception Paths: None

VI. PERFORMANCE

A. Performance Requirements

This library shall identify horizons using Aerolyzer's classifier in a test-selection of images with a minimum accuracy of 66% certainty. Images of horizons that contain sunsets and sunrises shall be identified using Aerolyzer's classifier library with a minimum of 50% certainty. This certainty shall be verifiable through a series of regression tests to prove detection accuracy.

B. Stretch Goals

If time permits, the library shall be deployed onto the live Aerolyzer website.

C. Tentative Development Schedule

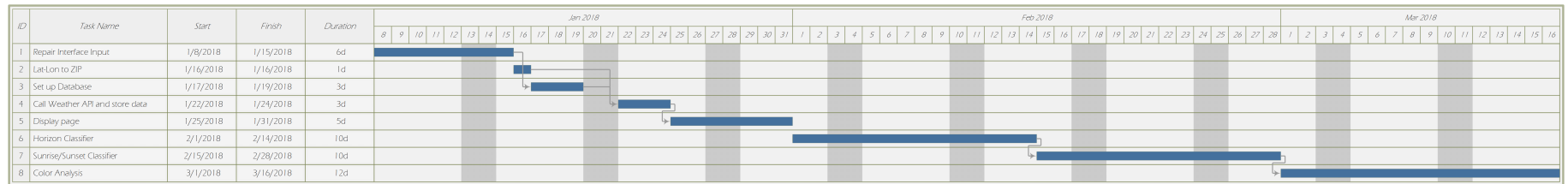


Fig. 1: Tentative Gantt Chart Schedule