Aerolyzer

Problem Statement

Daniel H. Ross

CS 460

Fall 2017

Abstract

Aerolyzer is an app development project that aims to provide users with near real-time updates on the air quality and conditions in their area. Aerolyzer's main purpose is to provide the users with information about the aerosol content in the air. The project is going to be focused on creating a python library capable of retrieving data from weather API and using image recognition software to derive relevant information from digital images.The amount of information that can be gleamed from a single picture could be potentially limitless so our development will be limited to shots of the horizon.

Problem Description

Atmospheric air quality isn't normally provided in an average weather service. US citizens that are particularly sensitive to variations in air quality or have a desire to learn more about atmospheric conditions for any reason can find it's difficult information to find. With the increase in the amount of cameras available to developers, it appears that digital images could be provide a wealth of data on atmospheric conditions. Digital images typically include meta-data that could also provide vital information relevant to the image itself. The Aerolyzer project is aimed to develop a python library that's capable of taking digital images as input and providing atmospheric conditions as output. The secondary goal of the Aerolyzer project is the development of a web-based application that fully utilizes the python library. The atmospheric analysis should be presented to the user in a clear set of visualizations and values.

Proposed Solution

For this library to provide an accurate analysis of the aerosol content of an area, the primary input is going to be the user's location. The user will provide this by two methods, the first being the input of a zip code, and the second being the taking of a picture on a device that has location services. Once we have the user's location we can send queries to multiple weather API, like wunderground, for their relevant data. The secondary input for the library is going to be digital images of horizons. Using machine learning software, such as OpenCV or Tensor flow, the library will be able to identify certain weather conditions and phenomena. Identifying sunsets, sunrises, fog, clear skies, and hazy skies can be accomplished by feature analysis alone. The library hopes to gather additional information on aerosols by analyzing the color of the sky. The color of the sky when the sun is close to the horizon is a large identifier of the aerosols present in the atmosphere in the image. This is due to the refraction of the sun's light as it passes through more of the atmosphere than it does at midday. The Aerolyzer library will compare the colors present in the digital images to data sets from atmospheric studies done by NASA and other legitimate resources. The Aerolyzer app that will utilize the python library has been developed in Django and will have to be expanded in order to properly display the atmospheric data in a appealing manner.

Performance

Given the nature of machine learning and the analysis of digital images as a whole, we're aiming to provide a library that can identify sunsets, sunrises, fog, clear skies, and hazy skies with a 5% tolerance. The library will be able to provide general weather data amassed from public API regardless of whether the user provides a image or not. The color analysis of user input images will only be available on images that are identified as sunsets or sunrises. Color analysis, once performed, will return the primary aerosol present with a tolerance of 5%. The color analysis may also return secondary and tertiary aerosol's with a 15% tolerance for error. The final out put will provide the user with the primary aerosol, secondary and tertiary aerosols only if the color analysis confirms them with 85% certainty, temperature, humidity, and average wind speed.