ChromaShare: Advanced Photo Organization and Searching

Diao, Haoran
diode@ucla.edu
UID:205280122

Soykin, Elissa
elissasoykin@ucla.edu
UID:905415537

Anderson, Alexa
ack909@ucla.edu
UID:505518981

Zamparini, Simone
simozampae@ucla.edu
UID:---------

October 18, 2021

1 Introduction

Most photo album apps that exist today do not allow for only the most rudimentary forms of organization. In addition, most automatic organization uses arcane criteria that the end-user can not understand, resulting in unpredictable behavior and a poor UX.

Solution: ChromaShare, an app which allows users to organize and share their photos through *tags.* and automatically sorts photos into tags based off of their *color histogram* and other color composition information, or through metadata such as location, title, and time. In addition, color and metadata information can be used to recommend tags from others.

2 Features

- Allow users to upload files along with a title and textual notes attached
- Allow users to create *tags* which can be attached to any photo. Each photo can have multiple tags
- Automatically acquire the color histogram of photos, and use that, along with metadata such as title, location, and date to sort into tags consisting of photos with similar color histograms and metadata
- Show users the color histogram, as well as a palette of the most common colors in a photo

- Allow users to create accounts with usernames and passwords, as well as make their tags (along with the tagged photos) private or public to other users
- make suggestions to the user of (public) tags with color composition and metadata similar to the user's photos and tags
- Allow users to search through photos and tags with metadata and color histogram information.

3 Architecture

3.1 Backend

Our backend will be written with `python` because it is familiar to most of our group. Most heavy duty image processing will be done with the `OpenCV` library so as to avoid any performance issues arising from using python as a backend. For storage of user data, tag information, and photos, we will use `MongoDB`, which is a relational database that will make organizing our data easier. We will also be serving the webapp page ourselves with the `http.server` python built in library. Communications to the front end will be done through a JSON API also implemented with http.server.

3.2 Frontend

A webapp will be written with `React`, with requests to the server done with `Jquery`, which has built in JSON parsing functions.

3.3 Chart

```
[Server Written in Python           ]          [WebApp written in React ]
[Serving the webapp with http.server]          [                        ]
[->communicating with the webapp wi-] JSON API [                        ]
[th a json API                      ] -------> [Jquery communicates wit-]
[processing images with OpenCV      ]          [Server                  ]
[storing data in MongoDB            ]          [                        ]
```

4 Summary

This app will let people save their photos to another location. It offers more options for customization than default, already-installed photos apps on phones. Instead of putting a photo into an album on their phone, users can decide to tag a photo with as many relevant ideas as they want. This will let them easily access memories of events and locations they have visited; in addition, they can look for photos related to various themes.