**Content-Based Image Retrieval System**

By Ethan Yee

CSS 484 | Wi

# Description

This is a simple CBIR system that uses a bag of words model to compare images. I created it from scratch using Python and Flask.

Here is documentation on setup and usage.

# Easy Viewing

! I have the application hosted on a server, so you can just go to https://cbir-flask.vercel.app/ to use the application quickly.

It should take a few milliseconds at first to fetch all the images, but after that, it should cache the website and run normally depending on your internet connection.

**Note: This version caches a list of intensity and color code bins from a database that was generated from the original app. It is not dynamic.**

# Prerequisites

* Python 3.10 or later
* Pip3

# To Run the Flask app manually (MacOS/Linux)

1. Open the base directory to this project in a \\*NIX-based OS

2. Load in a new python envrionment (Optional but best practice to mitigate conflicting deps) `python3 -m venv venv`

3. Source the new virtual environment (Optional but best practice to mitigate conflicting deps) `source venv/bin/activate`

4. Install requirements using `pip3 install -r requirements.txt`

5. To run the app, use `flask run` or `python3 run.py`

6. Make sure port 5000 is open

7. Open the browser and navigate to the link provided in the terminal, should be http://127.0.0.1:5000/

8. Wait for the application to load (it should take a few seconds, but after, it will cache all the images and their features to cookies, and it will run normally)

# Running the Flask app through setup.sh (MacOS/Linux)

1. Open the base directory to this project in a \\*NIX-based OS

2. Make sure the file has executable permissions `chmod +x setup.sh`

3. Make sure port 5000 is open

4. Run the script `./setup.sh` or `bash setup.sh`

5. Source the virtual environment: `source venv/bin/activate`

6. Run the app: `flask run` or `python3 run.py`

7. Open the browser and navigate to the link provided in the terminal, should be http://127.0.0.1:5000/

8. Wait for the application to load (it should take a few seconds, but after, it will cache all the images and their features to cookies, and it will run normally)

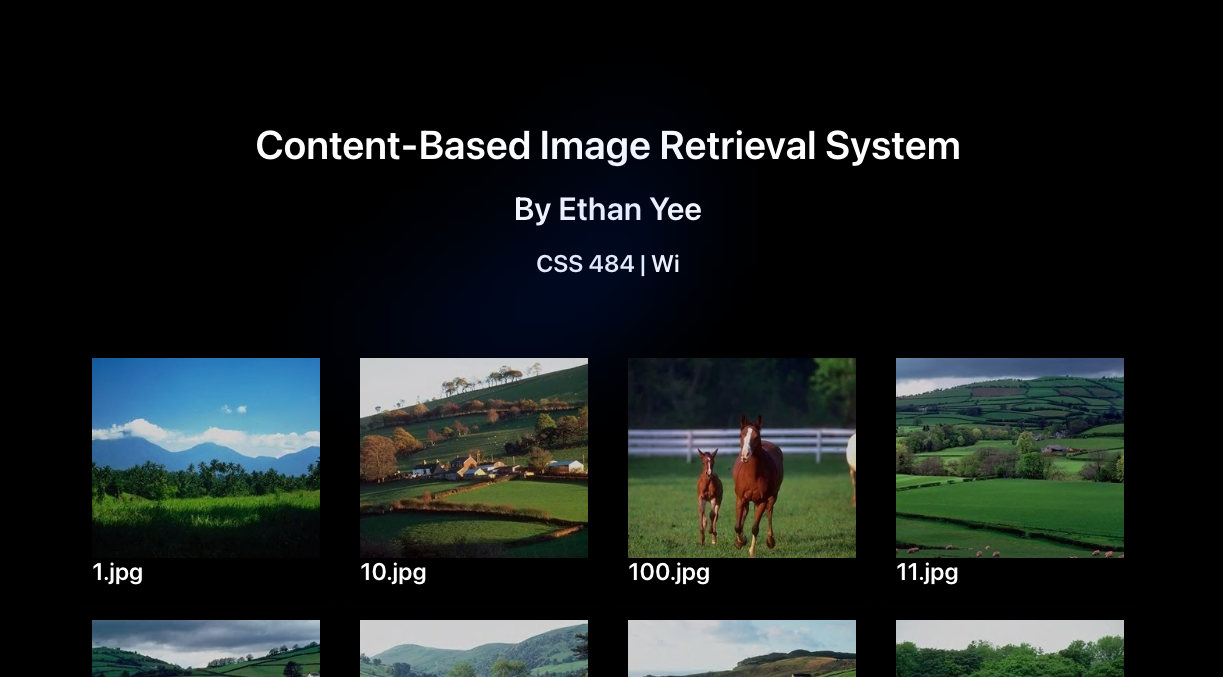
# Having Trouble?

Refer to the Documentation on Flask: <https://flask.palletsprojects.com/en/2.2.x/>

You can check all the requirements in this project, housed in requirements.txt.

If you don’t know what to do to install and set up the web app, you can also refer to setup.sh, which has all the executable commands that you would need to set everything up.

# Using the application

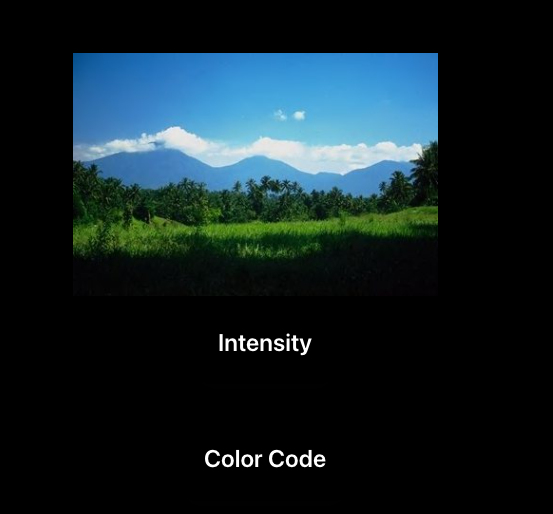


When the page loads, you will get a grid of 4x5 images ordered from left to right, top to bottom.

Graphical user interface, application

Description automatically generated

You can navigate to the next page to view different images in the database of images.



Clicking on an image will take you to the viewing page of the image. You can also choose your sort type, either Intensity or Color Code.

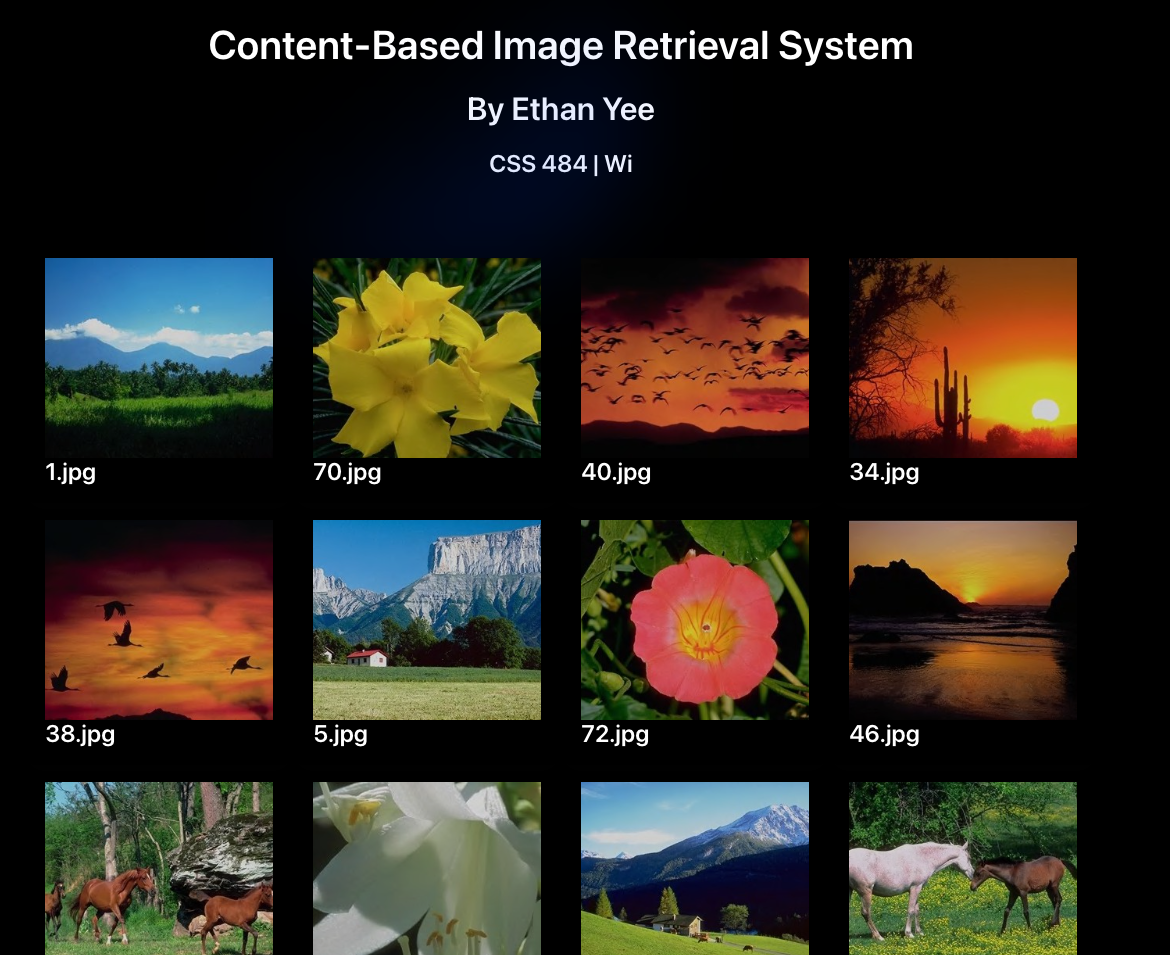
Graphical user interface, application

Description automatically generated

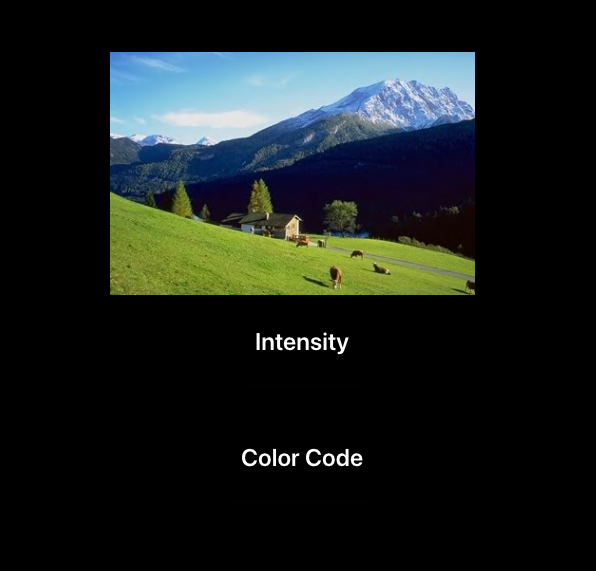
A picture containing dark

Description automatically generated

You can right click on the image to open the image in a new tab to inspect the image.



When you click intensity or color code, it will filter all the images closest to the target image’s intensity/color code, ordered from left to right, top to bottom, 1st page to nth page.



You can click the image to navigate back to the index route, or just go back in the web browser.