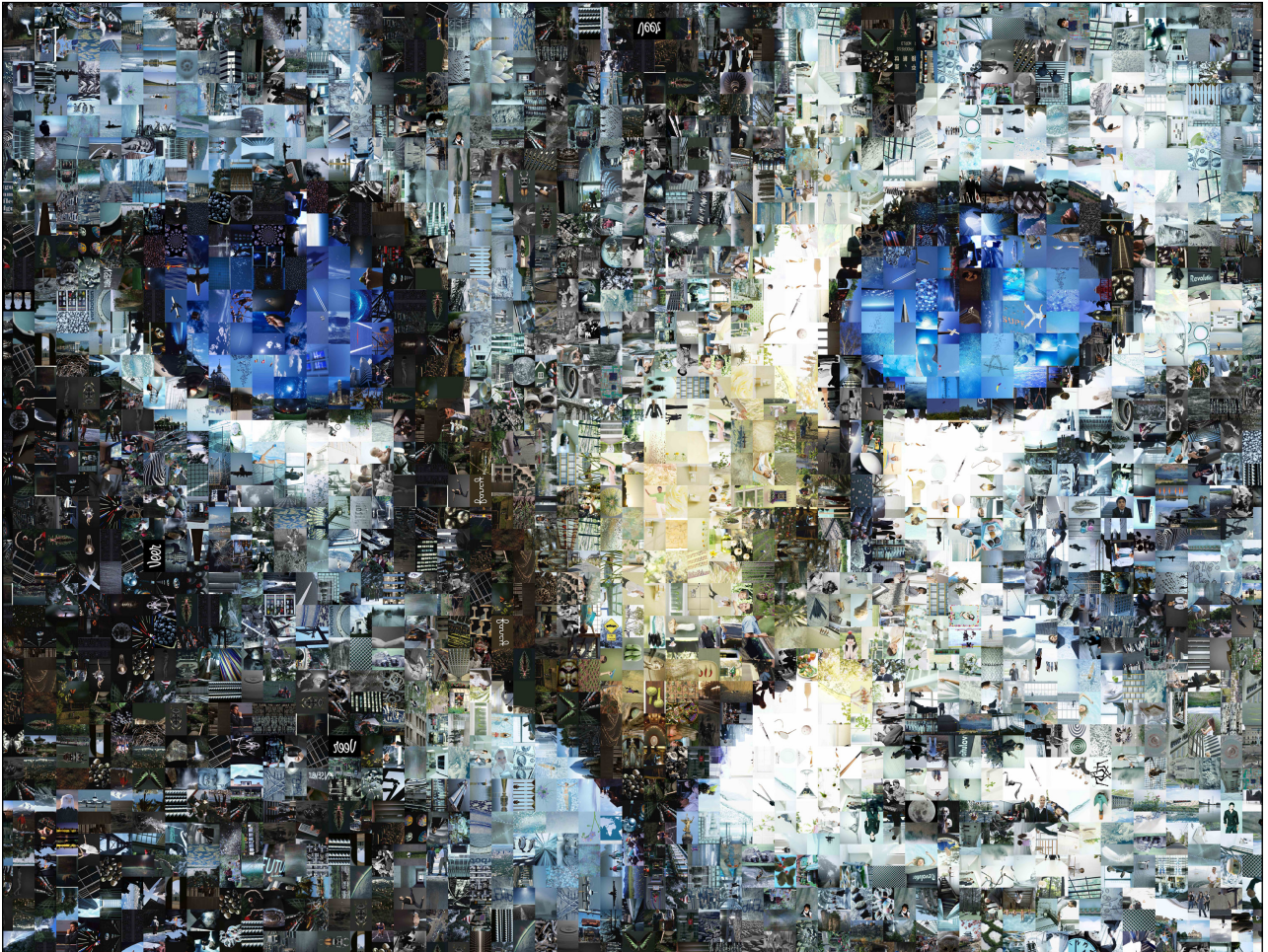

Photo Mosaic

Make a profile picture for your best friend.

0710006 盧可瑜 - 2019年1月17日



Overview & Motivation

What birthday gift will you sent to your best friend? A birthday card? A handmade cake? Next time, you may want to make something different and more interesting.

What I want to do is to make a photomosaic profile picture to sent to my best friend. For this reason, I'd like to make a project, a python program can generate my best friend's photomosaic profile picture. At first, it will ask the information of user's Facebook. Then it will analyze who is your best friend by Facebook activities, for example, who clicks like most frequently, who often tags you, and so on. Then, it will output his profile picture. What's more interesting, this picture would be made by "photomosaic". "Photomosaic" means that a picture is composed of a lot of small pictures, in fact, the picture would be divided into several small sections, and each of them would be replaced by the similar-color photo. When viewed at low magnifications, the individual pixels appears as the primary image.

I think this would be a fashionable project, and the users will have a lot of fun.

Plan

- **My code would be composed of the functions below.**
 - Input (Dec 25 ~ Dec 28)
 - ~~Ask user to input his Facebook access token.~~
 - Ask user to input some keywords.(What kind of photos will you use later?)
 - ~~FBBestFriend~~ (Dec 29 ~ Dec 30)
 - Use Facebook API to count the number of each friend's like of his posts.
 - Use Facebook API to count how many times did his friends tag him.
 - Find out who is his best friend, and then return his best friend's id.
 - ~~ProfilePicture~~ (Dec 31)
 - Use the id to get his best friend's profile picture.
 - PhotoDataset (Jan 1 ~ Jan 4)
 - Use the keywords inputted by user to search images.
 - Choose 500 most related images for each keyword.
 - Use BeautifulSoup to get the html, then download all these images.
 - Crop the images into the same size squares.
 - DividePicture (Jan 5)
 - Use OpenCV to divide the profile picture into pixels of the same sizes.
 - AverageRGB (Jan 5)
 - Average the RGB color value of each pixel and photo.
 - ClassifyData (Jan 6)
 - Classify the photos by their RGB values.
 - MatchPixel (Jan 6)
 - Search a photo with the most similar average RGB value of each pixel.
 - CombinePhoto (Jan 10)
 - Use OpenCV to combine the new profile picture with photos.
 - Delete data set by function in sys.
 - Output (Jan 10)
 - Use OpenCV to show the new profile picture through a window.
- **Test & Demo** (Jan 11 ~ Jan 14)
 - Ask some of my classmates to help me test my project. Then, demo it to TA.
- **The Web Data or Services I will use for this project.**
 - ~~Facebook Graph API~~
 - Use this API to get user's social graph.
 - Get user's friend list, numbers of like of the post, and tags.....
 - OpenCV (Open Source Computer Vision Library)
 - Use this API to deal with the images. Read, Resize, Crop, Combine, Show...

Update 1

- What I have done?
 - I can open an image file.
 - I can crop the image into the same size pixels.
 - I can show the images on a screen window.
- Changes
 - I can't not get the friend's user id from FB Graph API, so I'd like to change my plan.
 - The input will be changed to ask user to input a picture's path.
 - I will use a **Face API** to crop the face in the picture to make a photomosaic picture.
 - It is more free for user because they can choose which picture they want to use.
 - It is convenient because they don't need to change the size of picture on their own.

Update 2

- What I have done?
 - I can use **OpenCV** to finish all image modification, such as resize, crop, zoom.....
 - I can detect a face in the image and crop it.
- Changes
 - Use **face_recognition API** to detect face and crop it.
- Detail
 - User input, including file_path, folder_path, keywords.
 - Ask the user if he want to crop the face or not.
 - If yes, then use **face_recognition API** to detect the face's location.
 - Crop a rectangle of the face and save it as a new file.
 - Build the dataset by using the keywords to google search image.
 - Calculate the average RGB of the images in dataset.
 - Combine the photo
 - Resize the height and width of original file to the integer times of pixel_size.
 - Divide the photo into pixels.
 - Match each pixel to the image in the dataset with the closest RGB_value.
 - Save the photo file and output it though a window.

Test Result

- Large-size image is better.
- Small pixel size can make picture exquisite.
- More keywords, i.e. larger dataset, is better.

	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
Image	1	1	1	1	2	2
Original size	4928 × 3264	4928 × 3264	4928 × 3264	4928 × 3264	2176 × 1451	2176 × 1451
Crop size					385 × 385	385 × 385
keywords	cat, dog ,bird	cat, dog ,bird	cat, dog ,bird	cat	cat, dog ,bird	cat, dog ,bird
pixel size	24	32	16	24	24	16
crop_face	No	No	No	No	Yes	Yes



Image 1



Test 1



Test 2



Test 3



Test 4



Image 2



Crop



Test 5



Test 6

Run

- Install
 - pip install numpy
 - pip install cmake
 - pip install dlib
 - pip install face_recognition
 - pip install opencv-python
 - pip install beautifulsoup4
- Open the web
- Prepare a photo
- Run the code
- Be patient (It will run around five minutes)

```
Kellyde-Air:final_project kelly$ python Final.py
Which file do you want to use? Enter the path of the file:/Users/kelly/Desktop/final_project/IMG9590.jpg
Enter the keyword you want to use? Or enter "break" to end input:cat
Enter the keyword you want to use? Or enter "break" to end input:dog
Enter the keyword you want to use? Or enter "break" to end input:bird
Enter the keyword you want to use? Or enter "break" to end input:break
Enter a path of a folder which can save the output:/Users/kelly/Desktop/final_project
Do you want to crop the face from the image? Enter "Y"/"N":N
```

Timeline

Date	01/08	01/09	01/10	01/11	01/12	01/13	01/14	01/15	01/16	01/17
Week	Tue	Wen	Thu	Fri	Sat	Sun	Mon	Tue	Wen	Thu
Deadline			Demo1							Final
Doc-'Update2'+ 'Run'										
Code-Input										
Code-DividePicture										
Code-CropFace										
Code-PhotoDataset										
Code-AverageRGB										
Code-ClassifyData										
Code-MatchPixel										
Code-CombinePhoto										
Code-Output										
Test & Demo										