方法的简述 (50~100)

Open Image -> Preprocess image, reduce complexity -> Open color table & color json file, return matched colors -> Initialize models, fit data -> Iterate through image, stacks result

An image is first preprocessed into python dictionary object to reduce overall complexity, while color table is loaded and filtered. Preprocessed image is then processed by a model for its detailed color composition.

算法实现细节 (300+)

Opens image and converts it into RGB style while ignoring alpha channel.

If given any method, image will be processed by that method. The returning result of given method must be a 2-dimintion array, with the smallest element being RGB values. The array will then be iterated, each unique pixel will be recorded by its RGB value and reoccurrence. Result will be a Python dictionary object with RGB value as key, and it’s count as value.

The color table [1] is opened by pandas module with default argument, and the color json file is open by python internal function. Opened json file is then loaded by json module. If any given name has matched in json file, a filtered color-set will be returned. Else, the unedited raw table will be returned. If any element has failed matching (no such name), that row of result will instead be an empty row.

In this example, KNeighborsClassifier and LogisticRegressionClassifier are used. Both models are imported from sklearn module. The KNeighborsClassifier is initialized with parameters: result limited to one, and its algorithm set to brute. As for LogisticRegressionClassifier, it has parameters: max iteration set to element count of preprocessed image, and algorithm set to Liblinear. After initialization, RGB values and their Names from color-set are fitted into the module.

Each element in the preprocessed image is then iterated by a model. Each item will have its predicted name in color-set as result. Each result of pixel and its count is then added into the output. Final output has color name and it’s count as its structure.

[1] [Color Names | Kaggle](https://www.kaggle.com/datasets/avi1023/color-names)