Program Description of Emagine

jiahe.zhang 20205722

November 2022

1 Introduction

This is an "NFT-like community" that integrates image processing and image asset sharing transactions. Here users can upload images of their own creation or rework them with a high degree of freedom of tooling. The program guarantees that all image assets within the community are unique and tradable. Each asset has one owner, who holds the unique key to that asset and has the right to trade it or exhibit it in the community.

2 Progress during Alpha

1. Implemented login registration and related functions 2. Implement basic image style rendering based on matrix operation without external libraries (Basic Image Process) 3. Implement a series of image processing methods based on OpenCV with high degrees of freedom 4. Realized reading local storage images and uploading images 5. Realized the homepage flow display 6. Enables browsing of owned image assets 7. Implemented different screen size adaptability for all page layouts 8. Implemented LitePal database related operations 9. Initially realized real-time image rendering photography based on OpenCv Camera 10. Preliminary implementation of the image asset details page

3 Description

3.1 Dependencies(Those are not things that I wrote myself)

LitePal 3.0.0: database

OpenCV for Android SDK 4.4.0: For high-end image processing and real-time rendering cameras, etc. (only refer to the official relevant instructions, code writing and method implementation for your own implementation)

MagicIndicator 1.6.0: Optimized the horizontal slide function box I wrote

Facebook Fresco 0.14.1: Optimize the appearance of the cards I write

3.2 Main Activities

3.2.1 Basic Image Process

Basic image processing interface written in pure Java code. Users can pick up a local image, render it, and preview it. Rendering includes the ability to slide sliders to adjust hue, saturation, and grayscale. After clicking Upload to save, the system will compare the image hash in the database, and if there are no similar image assets, the user will be allowed to upload the image.

3.2.2 Advanced Image Process on OpenCV

Advanced image processing interface based on openCV. Users can select different rendering modes (gray, dilate, median blur, gaussian, canny scan, corrosion, binary) for rendering by clicking to upload the image. Users can perform facial recognition, and can use the ROI box to select parts of the image and render them individually (this function is under development). After OK, click Upload to save and the system will compare the image hash in the database, and allow users to upload images if there are no similar image assets.

3.2.3 Login & Register

The user can log in to an existing account, and the system will check if it exists and if the password matches. If you don't have an account, you can register, and the registration system can check whether the username already exists, whether the password is the same twice, etc. Users can choose to remember users when they log in, and the system will automatically display the user name of the last login when they log in the next time, and only need to enter the password. Status information is saved after login.

3.2.4 Front Page

The first page you go to after logging in. At the top is the function marquee, and below is the entire image of the waterfall.

3.2.5 Art Info

Click on the image display card on the homepage to enter the details. Details are displayed in the details, and options such as following the author, applying for transactions, and adding to favorites will be added in the future.

3.2.6 OpenCV Camera

Real-time rendering camera that supports several modes.

3.2.7 Cards

User collections displayed as dynamically swipeable stacked cards.

3.3 Main Tools

3.3.1 Img Transfer Tools

ImgHelper & Bytes2Bitmap

In order to perform operations such as display, rendering, and processing images in different situations, the method of image type conversion is integrated. This includes, but is not limited to, conversion between byte, byte[], drawable, Bitmap, etc. types.

3.3.2 Hash Tools

HashUtil

In order to ensure the uniqueness of the image, the Hash Distance algorithm is implemented to calculate the similarity of the image.

3.3.3 Basic Image Templates Matrices

BeautyUtil

It mainly encapsulates the matrices corresponding to various rendering templates, as well as methods for adjusting hue, saturation, and grayscale (which will be called when the user drags the slider is detected).

3.3.4 Card Swipe Tools

RoundImageView & CardConfig & CardItemTouchHelperCallback & CardLayoutManager & OnSwipeListener(Interface)

In order to allow users to have a better experience when browsing their own image assets, the touch response of the user's finger has been carefully rewritten, and the animation of card movement, floating logic, etc. have been designed to achieve the effect of users sliding the card left and right to switch.

3.3.5 Horizontal Swiped Topic(TaoBao Liked)

TopicActivity & HomeTopicPagerAdapter & TopicAdapter & TopicBean

Inspired by the function box at the top of the Taobao mobile app homepage, it imitates the style of the item, the logic of sliding left and right, etc., and then improves it.

3.3.6 PopUp Window Tools

Pop-up box, call in the context of user upload or other subsequent needs, and make improvements later.

3.4 Using of OpenCV

The program uses the relatively stable OpenCV 4.4.0 version. A series of models and methods provided in the library were learned, and image rendering methods such as gray, dilate, median blur, gaussian, canny scan, corrosion, and binary were implemented successively.

The ROI is initially realized, and the user's local selection and local rendering of the selected image will be realized on the basis of the ROI, so as to increase the degree of freedom.

At the same time, it can also be facial recognition based on the official European face dataset.

In the future, we will continue to use the related functions provided by OpenCV for camera development.