BÁO CÁO PROJECT 1

1. THÔNG TIN SINH VIÊN

Trương Đức Tài – 21078321

Lâm Quang Phú – 21094601

2. THỐNG KÊ MỨC ĐỘ HOÀN THÀNH

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| STT | Các chức năng | Mức độ hoàn thành | Sinh viên thực hiện | |
| Trương Đức Tài | Lâm Quang Phú |
| 1 | Checkered Board | 100% | X |  |
| 2 | Color Correction | 100% |  | X |
| 3 | Rotate Image | 100% | X |  |
| 4 | Color Separation | 100% |  | X |
| 5 | Corner Line | 100% | X |  |
| 6 | Gradient | 100% |  | X |
| 7 | Letter B | 100% | X |  |
| 8 | Find Secret By Substract | 100% |  | X |

3. PHÂN TÍCH VÀ MÔ TẢ THUẬT TOÁN

a. Tạo ảnh bàn cờ

|  |
| --- |
| - Ý tưởng: Tạo một ảnh bàn cờ (checkerboard) với kích thước ô vuông cố định, sử dụng NumPy hoặc OpenCV để thực hiện việc này. |
| - Mô tả thuật toán:   * Sử dụng NumPy:   + Tạo một ma trận board\_size x board\_size với giá trị ban đầu là trắng (255).  + Duyệt qua các ô trên bàn cờ, thay đổi giá trị thành đen (0) cho các ô có chỉ số chẵn dựa trên công thức (i + j) % 2 == 0.   * Sử dụng OpenCV:   + Tạo một ma trận board\_size x board\_size với giá trị ban đầu là đen (0).  + Vẽ các hình chữ nhật trắng (255) vào vị trí các ô chẵn và các ô lẻ của bàn cờ bằng cách sử dụng hàm cv2.rectangle. |
| - Code: |
| - Kết quả: Ảnh bàn cờ được tạo ra với kích thước và số lượng ô theo tham số đầu vào. Sử dụng NumPy cho phép dễ dàng điều chỉnh và tạo ảnh bàn cờ với cấu trúc đơn giản, trong khi OpenCV cung cấp cách nhanh chóng để vẽ các hình chữ nhật. |

b. Hiệu chỉnh màu sắc

c. Xoay hình ảnh

|  |
| --- |
| - Ý tưởng:  + Xoay ảnh một góc nhất định mà không bị mất thông tin hoặc cắt góc bằng cách tính toán kích thước mới của ảnh và áp dụng phép biến đổi affine.  + Lật ảnh theo chiều dọc hoặc chiều ngang bằng cách thay đổi chỉ số của các pixel trong mảng NumPy. |
| - Mô tả thuật toán:   * Xoay ảnh   + Tạo ma trận xoay: Tính toán ma trận xoay từ góc xoay.  + Tạo ma trận dịch chuyển: Tạo ma trận dịch chuyển để căn chỉnh trung tâm của ảnh mới với trung tâm của ảnh gốc.  + Kết hợp các ma trận: Kết hợp ma trận xoay và ma trận dịch chuyển để tạo ma trận affine tổng hợp.  + Tính toán kích thước mới của ảnh: Tính toán kích thước mới của ảnh sau khi xoay dựa trên góc xoay và kích thước ảnh ban đầu.  + Áp dụng phép biến đổi: Duyệt qua từng pixel của ảnh mới, tính toán tọa độ pixel tương ứng trong ảnh gốc và sao chép giá trị pixel từ ảnh gốc vào ảnh mới.   * Lật ảnh   + Lật theo chiều dọc: Đảo ngược các hàng của mảng ảnh  + Lật theo chiều ngang: Đảo ngược các cột của mảng ảnh |
| - Code:   * Xoay ảnh          * Lật ảnh |
| - Kết quả:   * Xoay ảnh: Ảnh đầu ra là phiên bản xoay của ảnh đầu vào, với kích thước đủ lớn để không bị cắt góc.      * Lật ảnh |

d.

e. Tạo một đường màu đen ở góc trên bên trái

|  |
| --- |
| - Ý tưởng: Tạo một đường line qua trung tâm của ảnh, xoay theo một góc chỉ định, sau đó dịch đường line lên góc trên bên trái của ảnh. |
| - Mô tả thuật toán:  + Tạo đường line:   * Tạo một ảnh trắng có cùng kích thước với ảnh gốc. * Vẽ một đường line màu đen ngang qua trung tâm của ảnh trắng này.   + Xoay đường line:   * Sử dụng một ma trận xoay để xoay đường line theo góc chỉ định.   + Dịch đường line:   * Tính toán khoảng cách dịch chuyển dựa trên tỷ lệ phần trăm từ biên đến tâm ảnh. * Dịch đường line bằng cách sử dụng ma trận dịch chuyển.   + Kết hợp với ảnh gốc:   * Kết hợp đường line đã xoay và dịch với ảnh gốc bằng cách áp dụng phép toán "min" để tạo ảnh cuối cùng. |
| - Code: |
| - Kết quả: Ảnh có đường line màu đen được xoay và dịch lên góc trên bên trái. Đường line này đi qua trung tâm ảnh và có thể được điều chỉnh độ dày, góc xoay, và khoảng cách dịch chuyển bằng các tham số đầu vào. |

f.

g. Tạo chữ B

|  |
| --- |
| - Ý tưởng: Tạo chữ cái 'B' bằng cách vẽ chữ cái 'E' trước, sau đó thêm các hình chữ nhật để hoàn thiện chữ 'B'. |
| - Mô tả thuật toán:  **+ Tạo chữ 'E':** Vẽ các phần của chữ 'E' (dòng thẳng bên trái và các đường ngang) bằng cách thay đổi giá trị của các pixel trong ảnh.  + **Thêm các hình chữ nhật:** Vẽ các phần còn lại của chữ 'B' bằng cách thêm các hình chữ nhật nối tiếp vào các khu vực cần thiết để hoàn thiện chữ 'B'. |
| - Code: |
| - Kết quả: **Chữ cái 'B'** sẽ được hiển thị với kích thước và độ dày theo tham số đầu vào. |

h.

4. TÀI LIỆU THAM KHẢO

[1] Prateek Joshi (2015). *OpenCV with Python By Example*. Packt Publishing.

[2] NumPy Documentation. <https://numpy.org/doc/stable/reference/generated/numpy.array.html>

[3] OpenCV Documentation <https://docs.opencv.org/4.x/dc/d2e/tutorial_real_time_pose.html>