

IMAGE SEGMENTATION WITH GRAB CUT ALGORITHM

Lâm Phát Tài

Phạm Nguyễn Thùy Trang

Nguyễn Xuân Tính

DESCRIPTION

Trong lĩnh vực xử lý ảnh, phân đoạn tiền cảnh và hậu cảnh là một vấn đề được mọi người hết sức quan tâm. Segmentation GrabCut giúp chúng ta có thể trích xuất được đối tượng chính và phụ trong bức ảnh của mình, nhờ đó người dùng và hơn nữa là máy tính có thể dễ dàng tiếp nhận được thông tin của bức ảnh. Một bức ảnh thông thường sẽ mang rất nhiều thông tin khác nhau, nhưng đa phần trong số đó chúng ta không sử dụng đến, do đó việc phân đoạn được tiền cảnh và hậu cảnh của bức ảnh sẽ nâng cao hiệu quả trong việc xử lý ảnh.

INTRODUCTION

Segmentation GrabCut là một phương thức giúp chúng ta thực hiện trích xuất, phân đoạn tiền cảnh và hậu cảnh của một bức ảnh một cách chính xác. GrabCut tiếp nhận một bức ảnh đầu vào từ người dùng với một bounding box dùng để xác định vị trí của đối tượng trong bức ảnh mà chúng ta muốn phân đoạn hoặc một mask gần đúng với phân đoạn. Trong đề tài này, chúng em sẽ thực hiện phân đoạn ảnh bằng phương thức sử dụng bounding box, chúng em hi vọng rằng sẽ trích xuất được đối tượng mà mình mong muốn bằng cách vẽ một hình chữ nhật bao quanh đối tượng mà chúng em muốn trích xuất.

MODEL

Gaussian Mixture Model (GMM) được sử dụng để mô hình hóa tiền cảnh và hậu cảnh. GMM học và gán nhãn (label) cho các điểm ảnh (pixel) không xác định và mỗi điểm ảnh được nhóm lại theo thống kê về màu sắc. Điều này thường được sử dụng để xây dựng một trường ngẫu nhiên Markov trên nhãn của điểm ảnh, với hàm năng lượng (energy function) thích hợp cho các vùng được kết

ALGORITHM

Thuật toán GrabCut được xây dựng dựa trên GraphCut. Thuật toán GraphCut phân đoạn đối tượng từ một hình ảnh được cho bằng cách giảm thiểu năng lượng (energy minimization). Đối với mỗi điểm ảnh (pixel) z_i trong hình ảnh, có một giá trị độ mờ (opacity) $a_i \in \{0, 1\}$ (với 0 cho nền sau và 1 cho tiền cảnh). θ_f, θ_b lần lượt mô tả sự phân bố hình ảnh tiền cảnh và hình ảnh nền sau, cũng là biểu đồ tần suất (histogram) của hai loại giá trị pixel (điểm ảnh tiền cảnh hoặc điểm ảnh nền).

$$\theta = \{h(z; a), a = 0, 1\}, \text{ trong đó } \int h(z, a) = 1$$

GraphCut định nghĩa một hàm năng lượng (energy function) E là hàm mất (loss) của nhiệm vụ phân đoạn bằng cách sử dụng hình thức năng lượng "Gibbs". Mục đích để giảm thiểu E .

$$E(a, \theta, z) = U(a, \theta, z) + V(a, z)$$

Điều kiện data U đánh giá mức độ phù hợp của a và z dựa trên biểu đồ tần suất (histogram) nhất định θ (θ_f hoặc θ_b).

$$U(a, \theta, z) = \sum_i \log h(a_i, z_i)$$

Điều kiện smoothness V đánh giá độ mượt của ranh giới phân đoạn. Trong đó $B(i, j)$ cao khi các giá trị của pixel i và pixel j gần nhau. Tham khảo [1] để biết định nghĩa chi tiết về $B(i, j)$.

$$V(a, z) = \gamma \sum_{(i,j) \in \text{img pixel}} [a_i \neq a_j] B_{(i,j)}$$

Nhiệm vụ phân đoạn sử dụng kỹ thuật Min Cut để tìm mức tối thiểu toàn cục nhằm giải quyết vấn đề tối ưu hóa.

$$\hat{a} = \operatorname{argmin} E(a, \theta, z)$$

GrabCut cải tiến hơn GraphCut chủ yếu ở hai khía cạnh. Đầu tiên, GrabCut thay thế biểu đồ bằng GMM (Mô hình Hỗn hợp Gaussian) trong điều kiện data U, GMM có thể được sử dụng cho không gian màu sắc. Thứ hai, tương tác với người dùng lặp đi lặp lại trong GrabCut giúp thuật toán đạt được hiệu quả tốt hơn. Có hai loại GMMs, một cho các điểm ảnh nền và một cho các điểm ảnh tiền cảnh, mỗi GMM có các thành phần K (thường là K = 5). Vì vậy:

1. Hàm năng lượng E trong GraphCut trở thành:

$$E(a, k, \theta, z) = U(a, k, \theta, z) + V(a, z), \text{ trong đó } k \in \{1, \dots, K\}.$$

2. Điều kiện data U (a, k, θ, z) trở thành:

$$\begin{aligned} D(a_i, k_i, \theta_i, z_i) &= -\log \pi(a_i, k_i) + \frac{1}{2} \log \det \Sigma(a_i, k_i) \\ &\quad + \frac{1}{2} [z_i - \mu(a_i, k_i)]^T \Sigma(a_i, k_i)^{-1} [z_i - \mu(a_i, k_i)] \\ U(a, k, \theta, z) &= \sum_i D(a_i, k_i, \theta_i, z_i) \end{aligned}$$

Trong đó $\mu(a_i, k_i)$ là trung bình của các giá trị và $\Sigma(a_i, k_i)$ là ma trận hiệp phương sai.

Điều kiện smoothness $V(a, z)$ giống như trong GraphCut. Sau khi khởi tạo, Thuật toán GrabCut được thực hiện với các bước sau:

1. Gán các thành phần GMM cho các điểm ảnh: Áp dụng cụm "k-means" cho các điểm ảnh tiền cảnh và nền được phân đoạn tương ứng bằng tay của người dùng. Số lượng các cụm là K. Với mỗi điểm ảnh z_i ở tiền cảnh và nền tương ứng, gán k_i với giá trị $D(a_i, k_i, \theta_i, z_i)$ tối thiểu.

$$k_i = \operatorname{argmin} D(a_i, k_i, \theta_i, z_i)$$

2. Học các parameter của GMM từ các điểm ảnh z. Đối với các điểm ảnh tiền cảnh được gán cho k_i , $\mu(a_i, k_i)$ và $\Sigma(a_i, k_i)$. Áp dụng cùng một phương pháp xử lý cho các điểm ảnh nền.

$$\theta = \operatorname{argmin}$$

3. Xây dựng đồ thị và tiến hành Min Cut.

$$\min E(a, k, \theta, z)$$

4. Tương tác với người dùng. Lặp lại các bước trên.

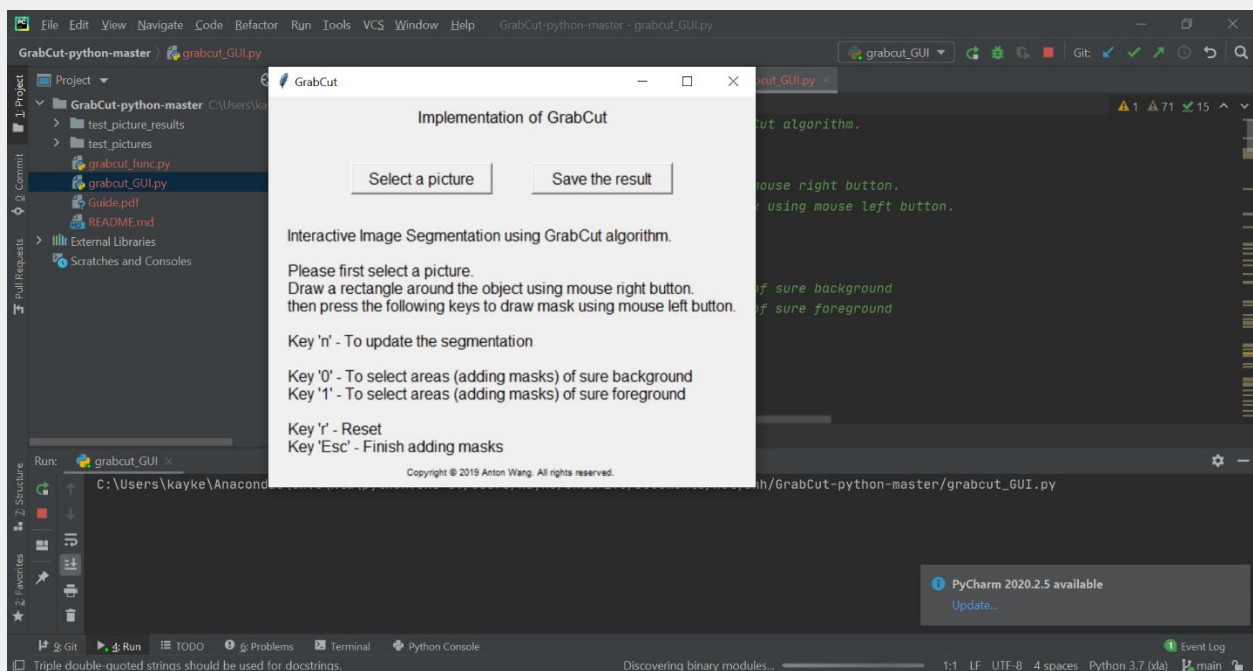
EXPERIMENT

System requirement: python 3.7.3, numpy, cv2, tkinter, igraph, sklearn

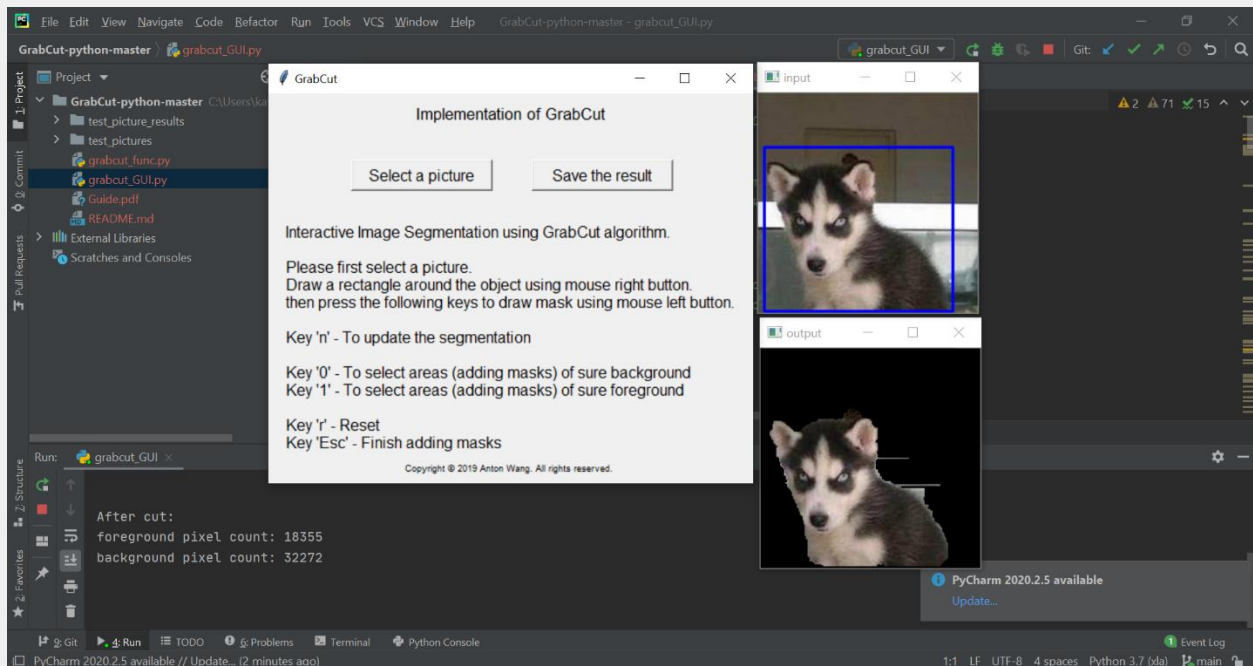
Các mã đã được thực hiện với các tài liệu tham khảo của mã nguồn mở trên github và package cv2.

Cách chạy code:

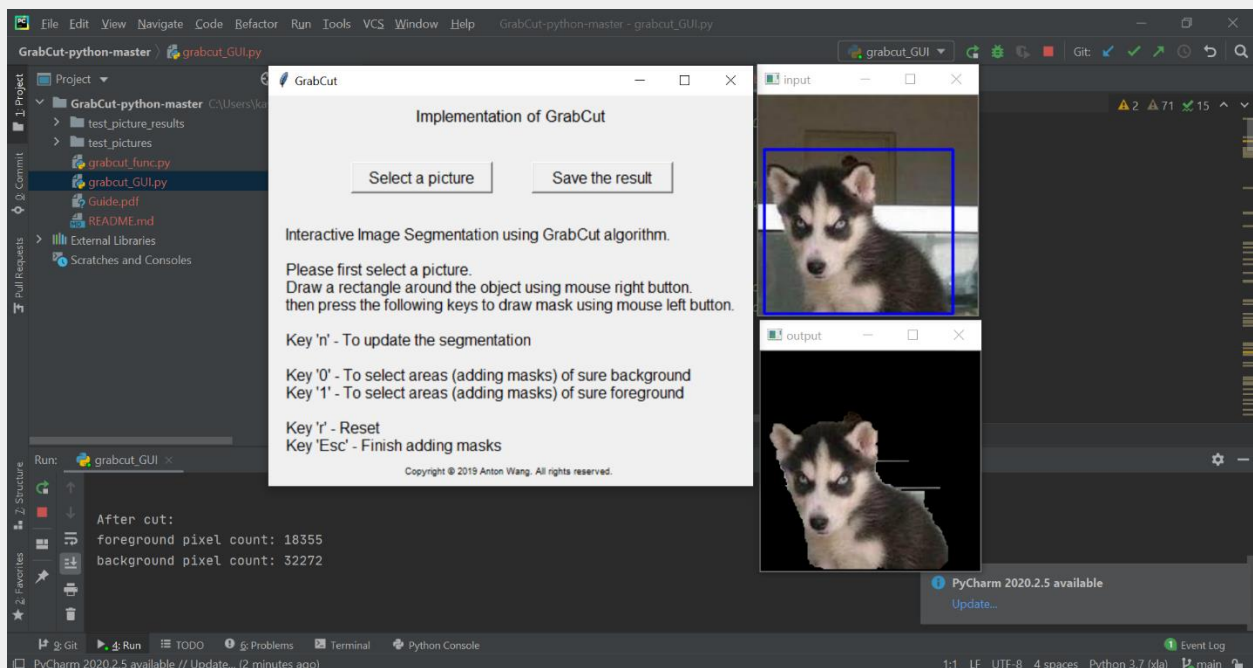
Chạy file `.\GrabCut\grabcut_GUI.py` bằng PyCharm. Sau đó, bạn sẽ thấy một giao diện được hiển thị dưới đây.



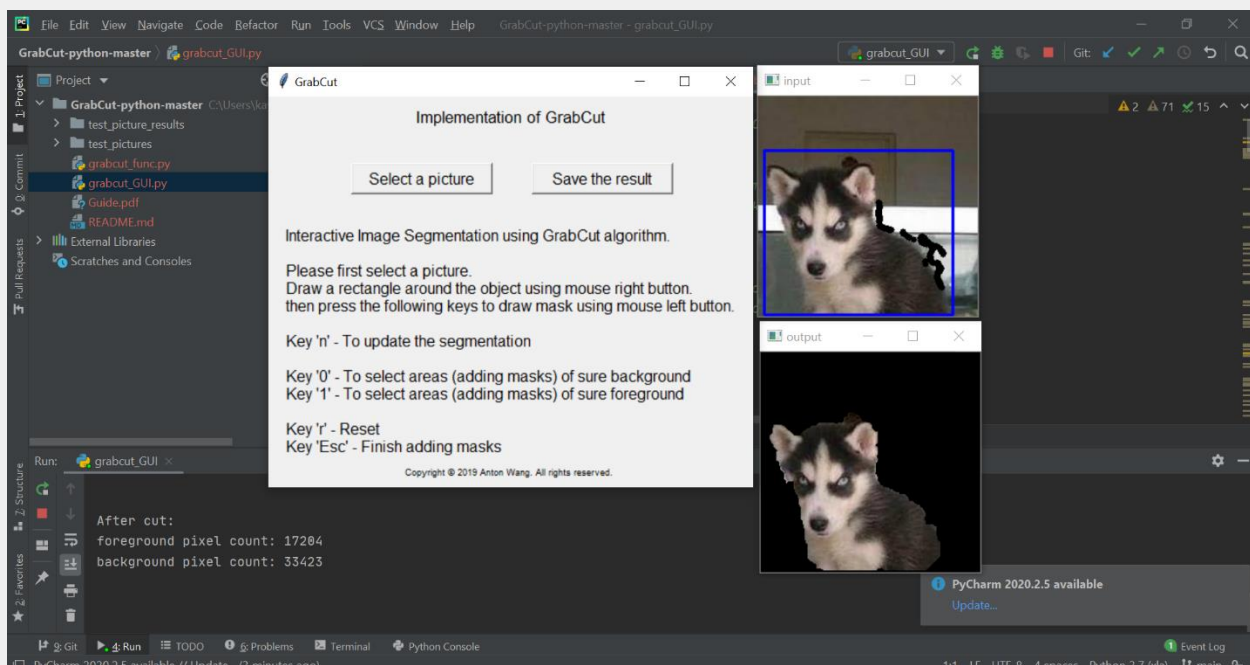
Trước tiên, nhấn nút “Select a picture” và chọn ảnh. Các hình ảnh mẫu được lưu trữ trong thư mục `.\GrabCut\test_pictures`. Ví dụ, chọn `.\GrabCut\test_picture\dog2.jpg`. Sau đó, hai cửa sổ sẽ bật lên hiển thị hình ảnh đầu vào và hình ảnh đầu ra.



Vẽ một hình chữ nhật xung quanh đối tượng bằng cách sử dụng nút chuột phải. Sau đó, một hộp thư sẽ bật lên. Làm theo hướng dẫn của hộp thư. Nhấn phím “n”, chờ vài giây và kết quả ban đầu sẽ được hiển thị trong cửa sổ “output”.



Tiếp theo, nhấn phím “0” để vẽ đường kẻ (màu đen) cho nền hoặc nhấn phím '1' để vẽ đường kẻ (trắng) cho tiền cảnh. Hãy nhớ nhấn phím “n” khi vẽ xong. Và chờ đợi trong vài giây. Ở đây, ví dụ chúng ta sẽ vẽ đường kẻ đen cho nền.



Nhấn phím “Esc” khi hoàn tất tất cả các bước. Kết quả có thể được lưu trong \GrabCut\grabcut_output.png bằng cách nhấn nút “Save the result” trên giao diện. Nhấn phím “r” để reset quá trình phân đoạn. Vui lòng tham khảo ghi chú chương trình của .\GrabCut\grabcut_GUI.py và .\GrabCut\grabcut_func.py để có thêm thông tin hướng dẫn thực hiện. File .\GrabCut\grabcut_GUI.py chứa code xây dựng giao diện và file .\GrabCut\grabcut_func.py chứa hai class grabcut và GMM, dùng để xây dựng thuật toán GrabCut.