```
function varargout = GUI(varargin)
% GUI MATLAB code for GUI.fig
%      GUI, by itself, creates a new GUI or raises the existing
%      singleton*.
%
%      H = GUI returns the handle to a new GUI or the handle to
%      the existing singleton*.
%
%      GUI('CALLBACK',hObject,eventData,handles,...) calls the local
%      function named CALLBACK in GUI.M with the given input
 arguments.
%
%      GUI('Property','Value',...) creates a new GUI or raises the
%      existing singleton*.  Starting from the left, property value
 pairs are
%      applied to the GUI before GUI_OpeningFcn gets called.  An
%      unrecognized property name or invalid value makes property
 application
%      stop.  All inputs are passed to GUI_OpeningFcn via varargin.
%
%      *See GUI Options on GUIDE's Tools menu.  Choose "GUI allows
 only one
%      instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help GUI

% Last Modified by GUIDE v2.5 13-Dec-2019 19:57:38

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                   'gui_Singleton',  gui_Singleton, ...
                   'gui_OpeningFcn', @GUI_OpeningFcn, ...
                   'gui_OutputFcn',  @GUI_OutputFcn, ...
                   'gui_LayoutFcn',  [] , ...
                   'gui_Callback',   []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT


% --- Executes just before GUI is made visible.
function GUI_OpeningFcn(hObject, eventdata, handles, varargin)
```

```
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to GUI (see VARARGIN)

% Choose default command line output for GUI
handles.output = hObject;
image = imread('./data/1.jpg');
axes(handles.axes1);imshow(image);
% Update handles structure
guidata(hObject, handles);
global Path L N foregroundInd backgroundInd BW choosefore chooseback;
Path = './data/1.jpg';
L = 0;
N = 0;
foregroundInd = 0;
backgroundInd = 0;
BW = 0;
choosefore = 0;
chooseback = 0;
guidata(hObject, handles);
% UIWAIT makes GUI wait for user response (see UIRESUME)
% uiwait(handles.figure1);


% --- Outputs from this function are returned to the command line.
function varargout = GUI_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;



% --- Executes on slider movement.
function Kslider_Callback(hObject, eventdata, handles)
% hObject    handle to Kslider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    K = round(get(handles.Kslider, 'Value'));
    set(handles.Klabel,'String', 'K='+string(K));
% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
 of slider


% --- Executes during object creation, after setting all properties.
function Kslider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Kslider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
```

```
% handles    empty - handles not created until after all CreateFcns
 called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
 get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on slider movement.
function Mslider_Callback(hObject, eventdata, handles)
% hObject    handle to Mslider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    M = round(get(handles.Mslider, 'Value'));
    set(handles.Mlabel,'String', 'M='+string(M));
% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range
 of slider


% --- Executes during object creation, after setting all properties.
function Mslider_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Mslider (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
 called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'),
 get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end


% --- Executes on button press in Foreground.
function Foreground_Callback(hObject, eventdata, handles)
% hObject    handle to Foreground (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    global I L foregroundInd backgroundInd BW choosefore chooseback;
    I = lab2rgb(I);
    axes(handles.axes1);imshow(I);
    [x,y] = getpts();
    x = round(x);
    y = round(y);
    h1 = impoly(gca,[x,y],'Closed',false);
    foresub = getPosition(h1);
    foreground = sub2ind(size(rgb2gray(I)),foresub(:,2),foresub(:,1));
    if choosefore == 0
        foregroundInd = foreground;
    else
        foregroundInd = [foregroundInd;foreground];
```

```
        end
        if chooseback ~= 0
            BW = lazysnapping(I,L,foregroundInd,backgroundInd);
            maskedImage = I;
            maskedImage(repmat(~BW,[1 1 3])) = 0;
            axes(handles.axes3);imshow(maskedImage);
        else
            if choosefore ~= 0
                BW1 = lazysnapping(I,L,foreground,backgroundInd);
                BW = BW | BW1;
                maskedImage = I;
                maskedImage(repmat(~BW,[1 1 3])) = 0;
                axes(handles.axes3);imshow(maskedImage);
            end
        end
        choosefore = 1;
        I = rgb2lab(I);
    % --- Executes on button press in Background.
    function Background_Callback(hObject, eventdata, handles)
    % hObject    handle to Background (see GCBO)
    % eventdata  reserved - to be defined in a future version of MATLAB
    % handles    structure with handles and user data (see GUIDATA)
        global I L foregroundInd backgroundInd BW choosefore chooseback;
        I = lab2rgb(I);
        axes(handles.axes1);imshow(I);
        [x,y] = getpts();
        x = round(x);
        y = round(y);
        h1 = impoly(gca,[x,y],'Closed',false);
        backsub = getPosition(h1);
        background = sub2ind(size(rgb2gray(I)),backsub(:,2),backsub(:,1));
        if chooseback == 0
            backgroundInd = background;
        else
            backgroundInd = [backgroundInd;background];
        end
        if choosefore ~= 0
            BW = lazysnapping(I,L,foregroundInd,backgroundInd);
            maskedImage = I;
            maskedImage(repmat(~BW,[1 1 3])) = 0;
            axes(handles.axes3);imshow(maskedImage);
        else
            if chooseback ~= 0
                BW1 = lazysnapping(I,L,foregroundInd,background);
                BW = BW & BW1;
                maskedImage = I;
                maskedImage(repmat(~BW,[1 1 3])) = 0;
                axes(handles.axes3);imshow(maskedImage);
            end
        end
        chooseback = 1;
        I = rgb2lab(I);

    % --- Executes on selection change in listbox1.
```

```
function listbox1_Callback(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    global Path chooseback choosefore;
    chooseback = 0;
    choosefore = 0;
    Path = './data/';
    picture = get(handles.listbox1, 'Value');
    switch picture
        case 1
            Path = [Path '1.jpg'];
        case 2
            Path = [Path '2.jpg'];
        case 3
            Path = [Path '3.jpg'];
    end
    image = imread(Path);
    axes(handles.axes1);imshow(image);
% Hints: contents = cellstr(get(hObject,'String')) returns listbox1
 contents as cell array
%        contents{get(hObject,'Value')} returns selected item from
 listbox1


% --- Executes during object creation, after setting all properties.
function listbox1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to listbox1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns
 called

% Hint: listbox controls usually have a white background on Windows.
%       See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
 get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end


% --- Executes on button press in SLIC.
function SLIC_Callback(hObject, eventdata, handles)
% hObject    handle to SLIC (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    K = round(get(handles.Kslider, 'Value'));
    M = round(get(handles.Mslider, 'Value'));
    global Path;
    image = imread(Path);
    I_lab = rgb2lab(image);
    I_gray = rgb2gray(image);
    global I;
    I = I_lab;
    display = 1;
```

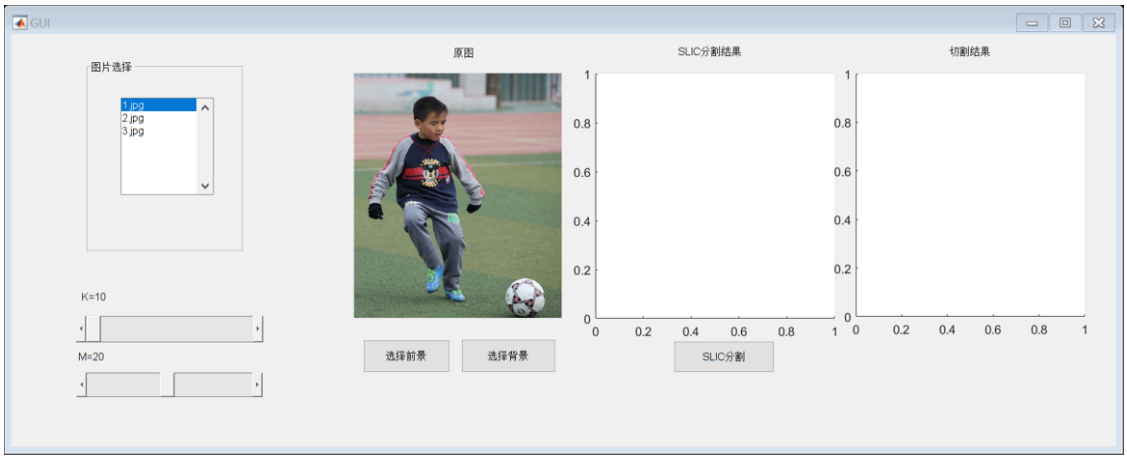
```
% SLIC
[m,n] = size(I_gray);
% ##S
S = round(sqrt(m*n/K));
% ###Label#Distance
Label = -1 * ones(m,n);
Distance = Inf * ones(m,n);
[H,W] = meshgrid(round(0.5 * S):S:m,round(0.5 * S):S:n);
H = reshape(H, 1, []);
W = reshape(W, 1, []);
Narray = size(H);
global N;
N = Narray(2);
% ###Center
C = zeros(N,5);
for i = 1:N
    % #3*3##############
    if H(i)+1 > m
        img = I(H(i)-2:H(i),W(i)-1:W(i)+1,1);
        [Fx, Fy] = gradient(img);
        F = sqrt(Fx .^ 2 + Fy .^ 2);
        [h,w] = ind2sub([3,3],find(F == min(min(F))));
        C(i,1) = I(H(i)-3+h(1),W(i)-2+w(1),1);
        C(i,2) = I(H(i)-3+h(1),W(i)-2+w(1),2);
        C(i,3) = I(H(i)-3+h(1),W(i)-2+w(1),3);
        C(i,4) = H(i)-2+h(1);
        C(i,5) = W(i)-2+w(1);
    else
        if W(i)+1 > n
            img = I(H(i)-1:H(i)+1,W(i)-2:W(i),1);
            [Fx, Fy] = gradient(img);
            F = sqrt(Fx .^ 2 + Fy .^ 2);
            [h,w] = ind2sub([3,3],find(F == min(min(F))));
            C(i,1) = I(H(i)-2+h(1),W(i)-3+w(1),1);
            C(i,2) = I(H(i)-2+h(1),W(i)-3+w(1),2);
            C(i,3) = I(H(i)-2+h(1),W(i)-3+w(1),3);
            C(i,4) = H(i)-2+h(1);
            C(i,5) = W(i)-2+w(1);
        else
            img = I(H(i)-1:H(i)+1,W(i)-1:W(i)+1,1);
            [Fx, Fy] = gradient(img);
            F = sqrt(Fx .^ 2 + Fy .^ 2);
            [h,w] = ind2sub([3,3],find(F == min(min(F))));
            C(i,1) = I(H(i)-2+h(1),W(i)-2+w(1),1);
            C(i,2) = I(H(i)-2+h(1),W(i)-2+w(1),2);
            C(i,3) = I(H(i)-2+h(1),W(i)-2+w(1),3);
            C(i,4) = H(i)-2+h(1);
            C(i,5) = W(i)-2+w(1);
        end
    end
end

for i=1:10
    for j = 1:N
```

```
            range = [max(C(j,4)-2*S,1), min(C(j,4)+2*S,m),
 max(C(j,5)-2*S,1), min(C(j,5)+2*S,n)];
            for h = range(1):range(2)
                for w = range(3):range(4)
                    dc = sqrt((I(h,w,1)-C(j,1))^2 + (I(h,w,2)-
C(j,2))^2 + (I(h,w,3)-C(j,3))^2);
                    ds = sqrt((h-C(j,4))^2 + (w-C(j,5))^2);
                    d = sqrt(dc^2 + ((ds/S)^2)*(M^2));
                    if d < Distance(h,w)
                        Distance(h,w) = d;
                        Label(h,w) = j;
                    end
                end
            end
        end
        for j = 1:N
            index = find(Label == j);
            [h,w] = ind2sub([m,n], index);
            Sizeh = size(h);
            sum = zeros(1,5);
            for k = 1:Sizeh(1)
                sum = sum + [I(h(k),w(k),1), I(h(k),w(k),2),
 I(h(k),w(k),3), h(k), w(k)];
            end
            sum = sum / Sizeh(1);
            sum(4) = round(sum(4));
            sum(5) = round(sum(5));
            C(j,:) = sum;
        end
        if display == 1
            bw = boundarymask(Label);

 axes(handles.axes2);imshow(imoverlay(lab2rgb(I),bw,'cyan'),'InitialMagnification'
            hold on;
            x = C(:,4);
            y = C(:,5);
            plot(y,x,'.');
            hold off;
        end
    end
    global L;
    L = Label;
    bw = boundarymask(L);

 axes(handles.axes2);imshow(imoverlay(lab2rgb(I),bw,'cyan'),'InitialMagnification'
    guidata(hObject, handles);
```



Published with MATLAB® R2018b