Ali Ilham Syahriansya

1918118/A

Computer Vision

Ekstraksi Citra Cabai

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| NO | Gambar | red | green | blue |
| 1 |  | 237.694 | 231.157 | 219.766 |
| 2 |  | 244.435 | 230.834 | 222.173 |
| 3 |  | 245.405 | 234.669 | 218.791 |
| 4 |  | 245.451 | 237.636 | 227.108 |
| 5 |  | 240.571 | 232.474 | 223.122 |
| 6 |  | 244.777 | 242.328 | 221.036 |
| 7 |  | 245.248 | 242.596 | 221.618 |
| 8 |  | 248.446 | 236.181 | 224.135 |
| 9 |  | 244.576 | 240.013 | 222.273 |
| 10 |  | 244.096 | 232.393 | 221.154 |
| 11 |  | 243.78 | 225.88 | 216.247 |
| 12 |  | 245.037 | 240.566 | 223.748 |
| 13 |  | 247.005 | 233.413 | 224.206 |
| 14 |  | 247.401 | 235.401 | 220.172 |
| 15 |  | 248.683 | 236.051 | 222.211 |
| 16 |  | 244.713 | 236.068 | 222.265 |
| 17 |  | 246.518 | 243.048 | 224.834 |
| 18 |  | 247.049 | 242.969 | 224.964 |
| 19 |  | 248.201 | 238.34 | 213.047 |
| 20 |  | 244.999 | 225.904 | 215.703 |

*Source Code* :

|  |
| --- |
| function pushbutton1\_Callback(hObject, eventdata, handles)  % hObject handle to pushbutton1 (see GCBO)  % eventdata reserved - to be defined in a future version of MATLAB  % handles structure with handles and user data (see GUIDATA)  [nama\_file1, nama\_path1]=uigetfile('\*,\*.jpg,\*.png,\*.jpeg');    if ~isequal (nama\_file1, 0);  %membaca ctra rgb  citra=imread(fullfile(nama\_path1,nama\_file1));  %menampikan citra di axes  axes(handles.axes1)  imshow(citra)  title('citra yang di olah')  %simpan variabel i d dalam handles agar bisa di simpan  handles.citra=citra;  guidata(hObject,handles)  else  return  end      % --- Executes on button press in pushbutton2.  function pushbutton2\_Callback(hObject, eventdata, handles)  % hObject handle to pushbutton2 (see GCBO)  % eventdata reserved - to be defined in a future version of MATLAB  % handles structure with handles and user data (see GUIDATA)  ekstrak\_citra=handles.citra;  rgb=ekstrak\_citra;  nilai\_hasil = uint8(zeros(size(rgb,1), size(rgb,2), size(rgb,3)));    %R,G,B components of the input image  R = rgb(:,:,1);  G = rgb(:,:,2);  B = rgb(:,:,3);    %Inverse of the Avg values of the R,G,B  mR = 1/(mean(mean(R)));  mG = 1/(mean(mean(G)));  mB = 1/(mean(mean(B)));    %Smallest Avg Value (MAX because we are dealing with the inverses)  maxRGB = max(max(mR, mG), mB);    %Calculate the scaling factors  mR = mR/maxRGB;  mG = mG/maxRGB;  mB = mB/maxRGB;    %Scale the values  nilai\_hasil(:,:,1) = R\*mR;  nilai\_hasil(:,:,2) = G\*mG;  nilai\_hasil(:,:,3) = B\*mB;    %Convert the image from RGB to YCbCr  img\_ycbcr = rgb2ycbcr(nilai\_hasil);  Cb = img\_ycbcr(:,:,2);  Cr = img\_ycbcr(:,:,3);    %Detect Skin  [r,c,v] = find(Cb>=77 & Cb<=127 & Cr>=133 & Cr<=173);  numind = size(r,1);    bin = false(size(rgb,1), size(rgb,2));  %Mark Skin Pixels  for i=1:numind  bin(r(i),c(i)) = 1;  end    bin = imfill(bin,'holes');  R(~bin) = 0;  G(~bin) = 0;  B(~bin) = 0;  nilai\_hasil = cat(3,R,G,B);  axes(handles.axes2)  imshow(nilai\_hasil)      red=mean(mean(rgb(:,:,1)));  green=mean(mean(rgb(:,:,2)));  blue=mean(mean(rgb(:,:,3)));    set(handles.edit1,'string',red);  set(handles.edit2,'string',green);  set(handles.edit3,'string',blue); |