DSP MATLAB Project Report

Image Color Filtering

2019 년 12 월 10 일

32151671 박민혁
32153180 이상민
32155068 홍승기

1. Project 개요

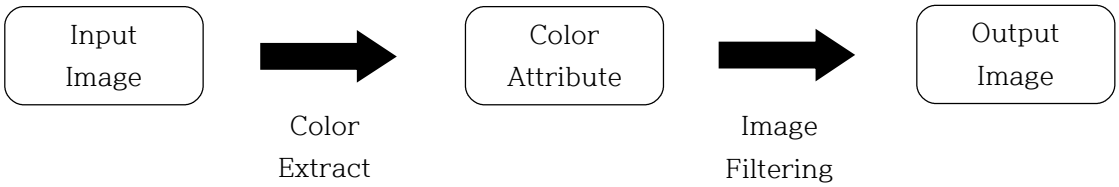
1.1 Project 개발 필요성

- 이미지는 여러 색상의 집합인데 우리는 각 이미지들이 얼마나 많은 색상들로 이루었는지를 알고 싶었다. 육안으로는 정확한 색을 구분하는 데에 한계가 있다. 따라서 필터링을 통해 이미지 내사용 빈도가 높은 색상부터 나타내고, 더 나아가 하나의 색상만을 추출해 이미지를 재구성한다.

1.2 Project 개발 배경 또는 동향

- 인터넷에 이미지를 등록하면 색상을 추출해주는 사이트는 상당히 많다. 하지만 우리는 단순히 색상을 얻는 것으로 끝나는 것이 아니라 한 가지 색상만을 사용해 이미지를 재구성하고 싶었다.

1.3 기본적인 동작 및 기능



1.4 Project 개발 방법

a. 참고 MATLAB project 출처

https://kr.mathworks.com/matlabcentral/fileexchange/49898-image-color-filtering

https://kr.mathworks.com/matlabcentral/fileexchange/69538-image2palette-simple-k-means-color-clustering

b. 수정 및 개선 필요성, 범위 및 방법

- 색상 범주를 조금 더 구체화할 필요가 있다. 현재는 6가지로 나뉘어 있는데, 보다 정확한 색상을 추출하기 위해서는 더 많은 색상들로, 더욱 좁은 범위를 사용해야 한다.

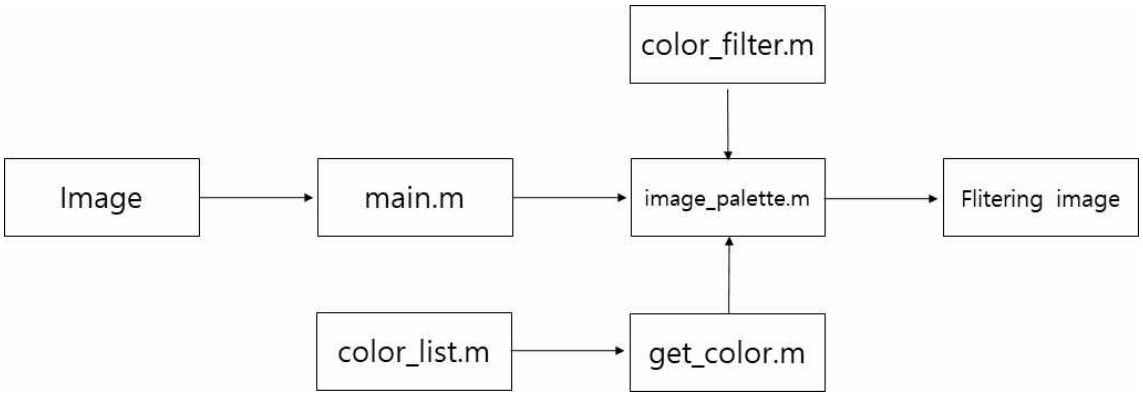
c. Project에서 본인들의 기여도 %로 측정 및 근거 제시

이름	역할	근거	기여도
박민혁	알고리즘 분석	알고리즘 및 블록다이어그램 기술	33%
이상민	코드 분석	각 함수 기능 설명	33%
홍승기	코드 구현	색상 범주별 추출 코드 작성	33%

2. Project 수행 내용

2.1 알고리즘 설명

1. 지정된 이미지 호출
2. 이미지 RGB 요소 LAB로 변환
3. 이미지의 색상 구성 confirm
4. 지정한 HSV 색상 영역만큼 필터링 된 이미지 confirm

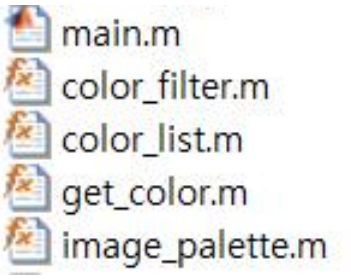


위 그림을 토대로 설명하면, main에서 이미지를 읽어와 image_palette 함수에서 이미지를 구성하고 있는 색상 요소들을 구별한다. color_filter 함수에서는 이미지를 특정한 색상으로 필터링하여 재구성된 모습을 새로운 창에 표시해준다.

get_color 함수와, color_list 함수는 서로 상호 작용하여 추출된 색상을 RGB color list와 일치하는 색상의 이름을 보여주는 데 사용된다. 알고리즘의 구성이 단순하게 되어있기 때문에 코드를 처음 보는 사람도 쉽게 이해할 수 있었다.

2.2 MATLAB 구현

- 함수별 설명



main.m : 이미지를 읽어 image_palette.m 함수 호출
color_filter.m : HSV 범위를 이용하여 특정 색상 필터링
color_list.m : RGB 표에 의한 색상 정보
get_color.m : color_list.m으로부터 색상 이름 전달받음
Image_palette.m : main.m에서 이미지를 받아온 후 색상 추출

- 예상 출력값

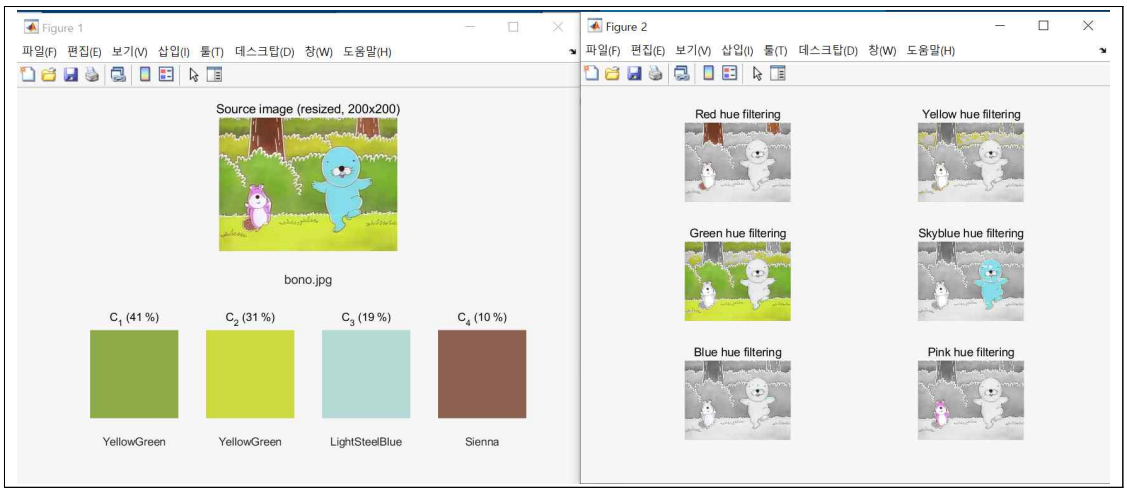
① 지정된 이미지의 컬러 구성에 맞는 색상들이 추출
② 미리 지정한 HSV의 H 영역 범위를 이용, 특정 색상을 제외한 나머지 색상은 필터링하여 이미지 재구성

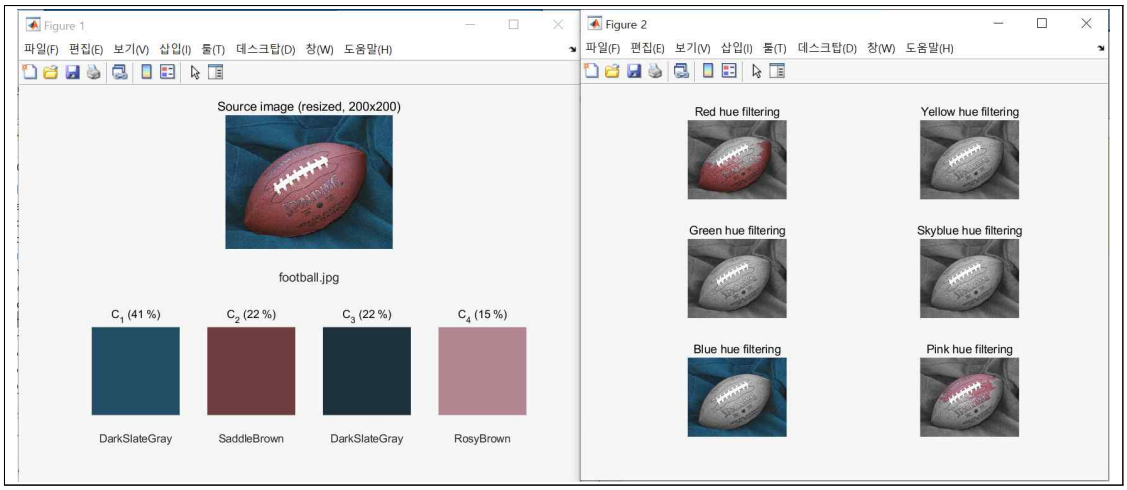
하지만 HSV와 RGB는 색 구성 방법부터 다르기 때문에 양쪽 색상의 호환 문제가 중요할 것 같다.

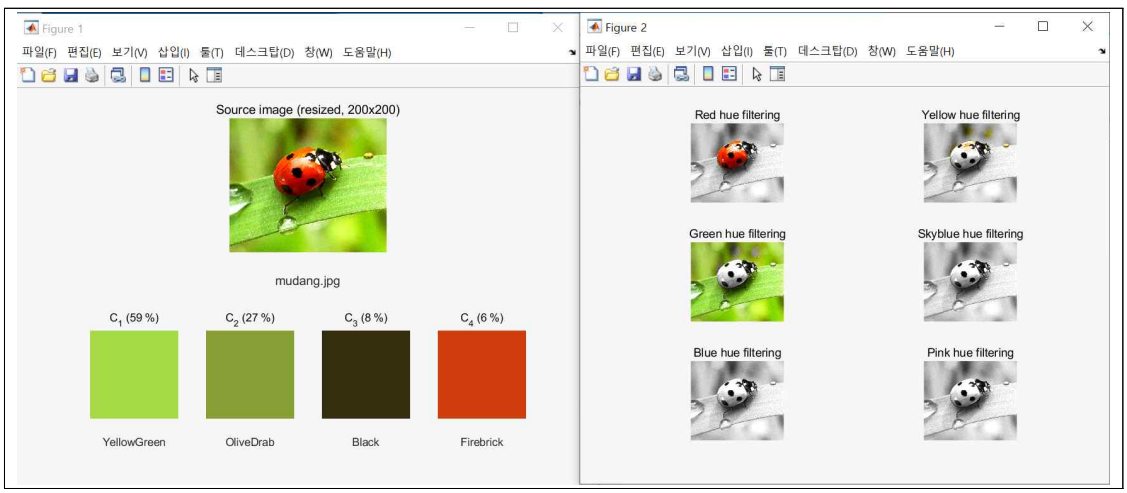
2.3 입력 및 출력

- 입력에 대한 결과물 설명

1. 그림과 실제 사진 비교







2. 색 구분이 뚜렷한 경우

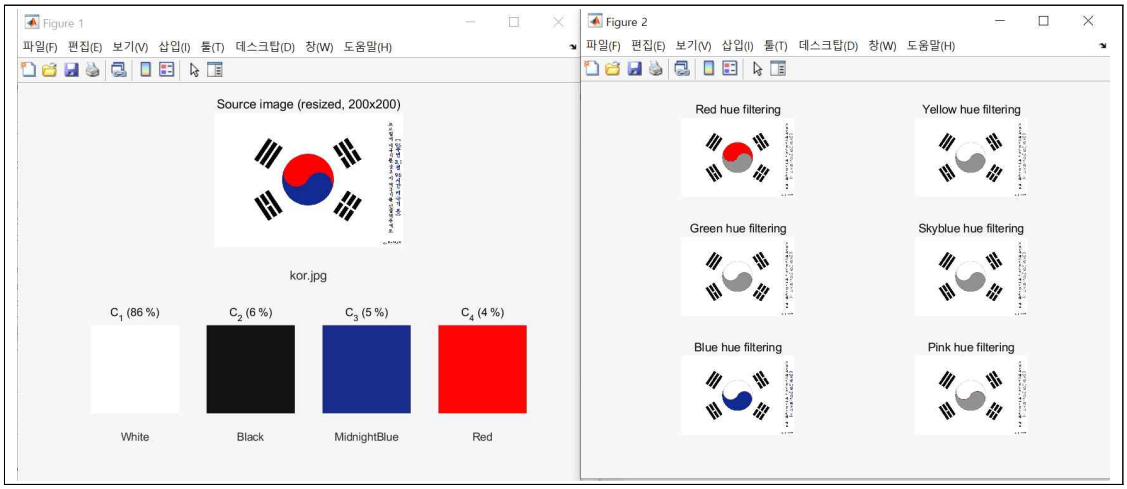


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Source image (resized, 200x200)
kor.jpg
C1 (86 %)
C2 (6 %)
C3 (5 %)
C4 (4 %)
White
Black
MidnightBlue
Red
Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Red hue filtering
Yellow hue filtering
Green hue filtering
Skyblue hue filtering
Blue hue filtering
Pink hue filtering

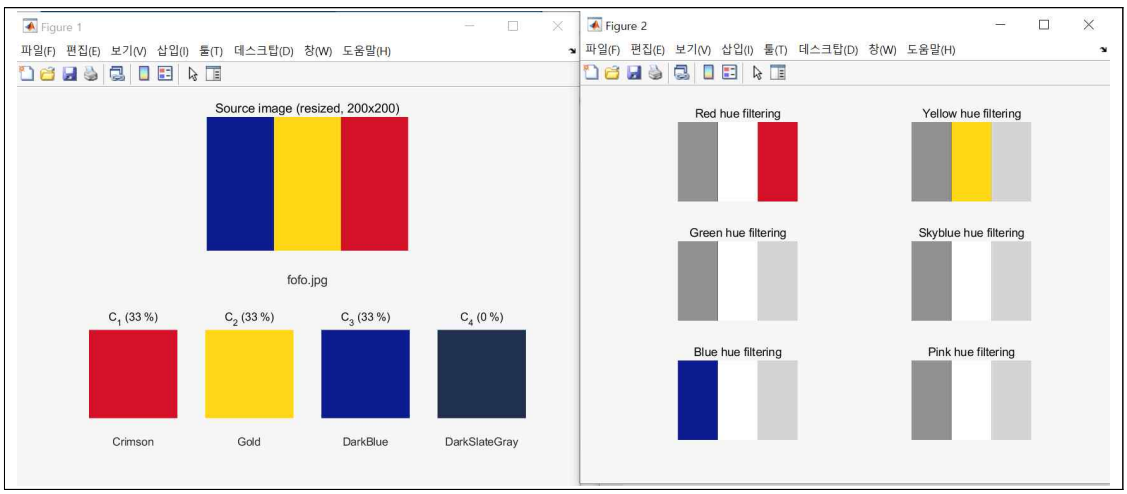


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Source image (resized, 200x200)
fofo.jpg
C1 (33 %)
C2 (33 %)
C3 (33 %)
C4 (0 %)
Crimson
Gold
DarkBlue
DarkSlateGray
Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Red hue filtering
Yellow hue filtering
Green hue filtering
Skyblue hue filtering
Blue hue filtering
Pink hue filtering

3. 색 구분이 복잡한 경우

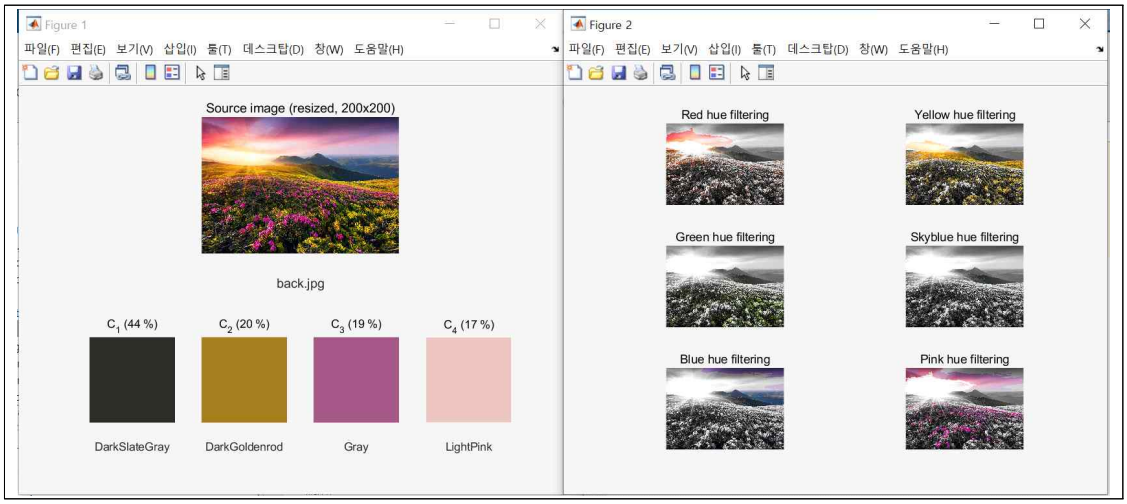


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Source image (resized, 200x200)
back.jpg
C1 (44 %)
C2 (20 %)
C3 (19 %)
C4 (17 %)
DarkSlateGray
DarkGoldenrod
Gray
LightPink
Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Red hue filtering
Yellow hue filtering
Green hue filtering
Skyblue hue filtering
Blue hue filtering
Pink hue filtering

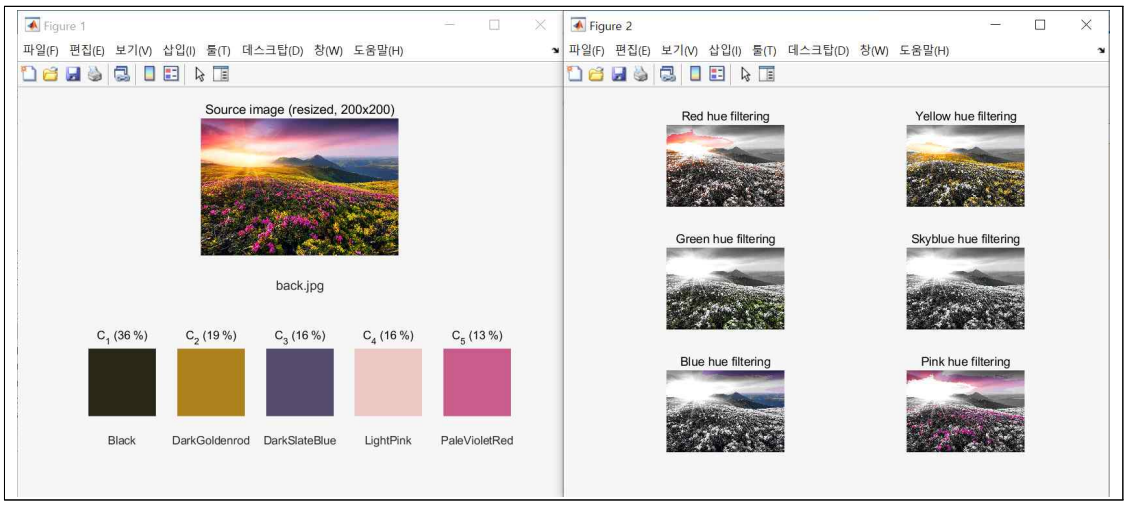


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Source image (resized, 200x200)
back.jpg
C1 (36 %)
C2 (19 %)
C3 (16 %)
C4 (16 %)
C5 (13 %)
Black
DarkGoldenrod
DarkSlateBlue
LightPink
PaleVioletRed
Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Red hue filtering
Yellow hue filtering
Green hue filtering
Skyblue hue filtering
Blue hue filtering
Pink hue filtering

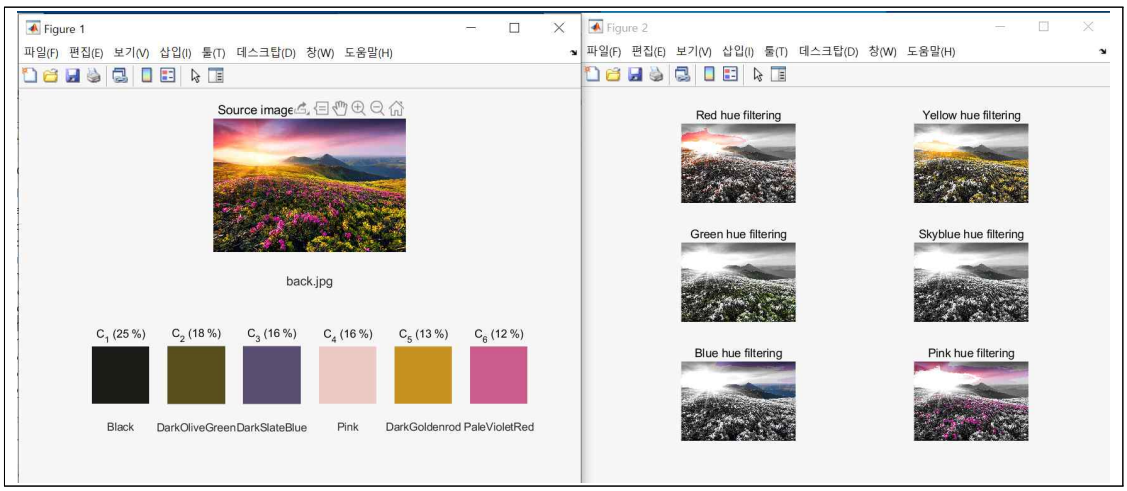


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Source image
back.jpg
C1 (25 %)
C2 (18 %)
C3 (16 %)
C4 (16 %)
C5 (13 %)
C6 (12 %)
Black
DarkOliveGreen
DarkSlateBlue
Pink
DarkGoldenrod
PaleVioletRed
Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Red hue filtering
Yellow hue filtering
Green hue filtering
Skyblue hue filtering
Blue hue filtering
Pink hue filtering

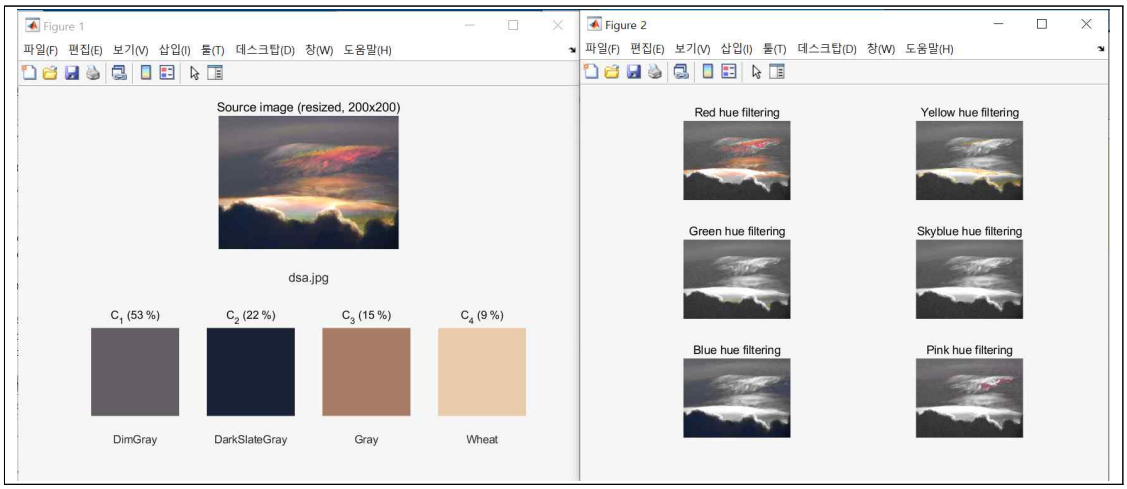


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Source image (resized, 200x200)
dsa.jpg
C1 (53 %)
C2 (22 %)
C3 (15 %)
C4 (9 %)
DimGray
DarkSlateGray
Gray
Wheat
Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Red hue filtering
Yellow hue filtering
Green hue filtering
Skyblue hue filtering
Blue hue filtering
Pink hue filtering

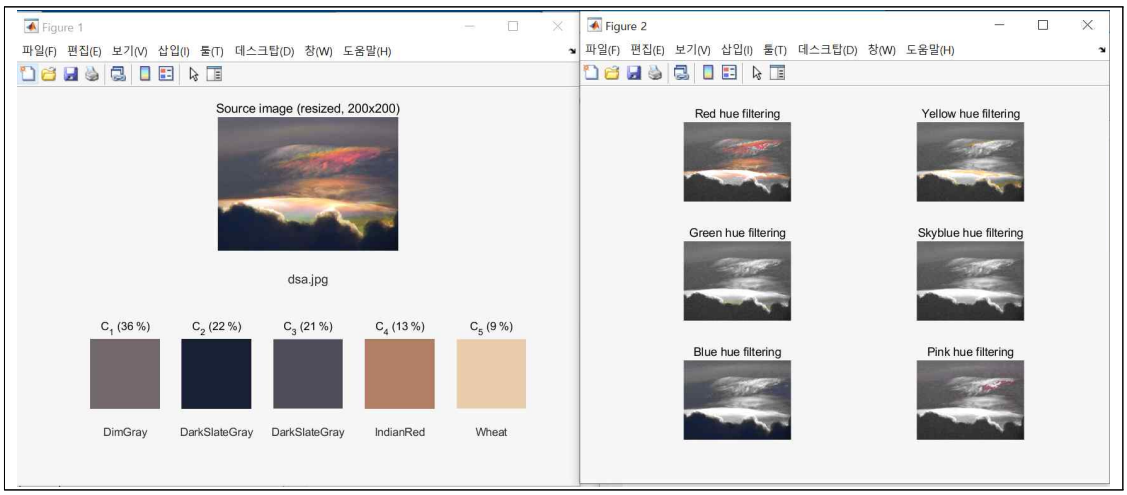


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Source image (resized, 200x200)
dsa.jpg
C1 (36 %)
C2 (22 %)
C3 (21 %)
C4 (13 %)
C5 (9 %)
DimGray
DarkSlateGray
DarkSlateGray
IndianRed
Wheat
Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Red hue filtering
Yellow hue filtering
Green hue filtering
Skyblue hue filtering
Blue hue filtering
Pink hue filtering

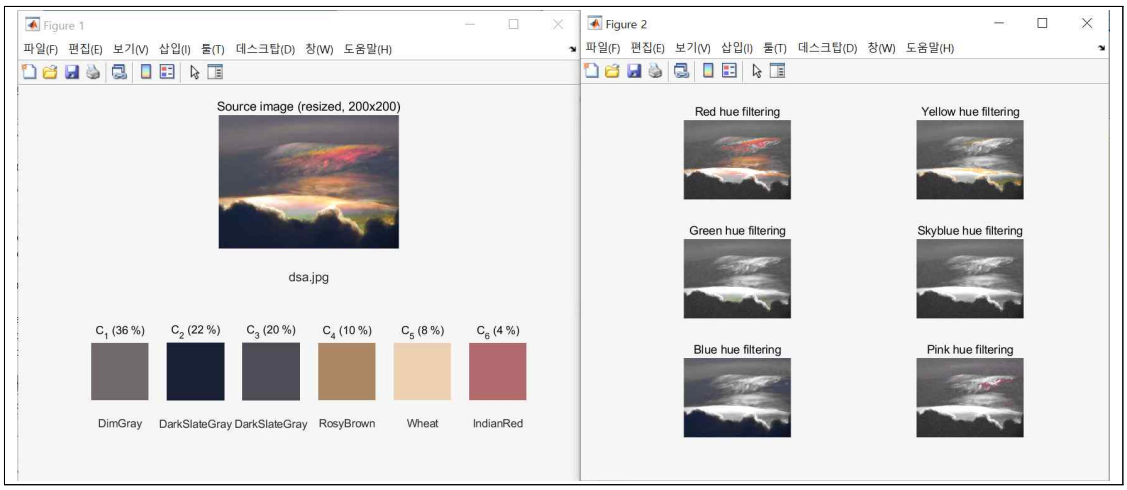


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Source image (resized, 200x200)
dsa.jpg
C1 (36 %)
C2 (22 %)
C3 (20 %)
C4 (10 %)
C5 (8 %)
C6 (4 %)
DimGray
DarkSlateGray
DarkSlateGray
RosyBrown
Wheat
IndianRed
Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Red hue filtering
Yellow hue filtering
Green hue filtering
Skyblue hue filtering
Blue hue filtering
Pink hue filtering

- 결과물을 비교하여 알고리즘 성능 분석

결과물은 위의 표에서 볼 수 있듯이 3가지 기준을 갖고 분류했다.

① 그림과 실제 사진을 비교한 경우

그림과 실제 사진을 이용해 필터링을 해본 결과 실제 사진에서는 결과가 잘 나오지만 그림에서는 결과 값이 제대로 나오지 않는 것을 볼 수 있다. 따라서 그림보다는 선명한 색감을 지닌 실제 사진을 이용해 필터링을 하는 것이 정확도가 높다는 것을 알 수 있었다.

② 색 구분이 뚜렷한 경우

색 구분이 뚜렷한 사진을 이용해 필터링을 해본 결과 국기처럼 색이 잘 구분되어 있는 사진일수록 정확도가 높았다. 두 번째 국기 사진의 경우 3가지 색을 사용했는데 결과 값으로 4가지의 색이 나왔다. 하지만 마지막 색상인 DarkSlateGray 성분이 0% 들어있다는 것을 알 수 있다. 따라서 cluster 수가 사진에 사용된 색상 수보다 많을 경우 임의의 값이 들어간다는 것을 알 수 있었다.

③ 여러 색들이 모여 있어 색 구분이 복잡한 경우

마지막으로 색 구분이 복잡할 경우의 사진을 이용해 필터링을 해보았다. 이러한 사진들은 색 구분이 어려워 색상을 성공적으로 분석하지 못할 것이라고 생각했다. 우선 cluster를 4개부터 시작해서 5개, 6개 점점 개수를 증가시키면서 비교했다. 4개보단 5개, 5개보단 6개의 cluster를 사용했을 때가 결과 값이 잘 나왔다. 따라서 색 구분이 복잡한 경우엔 cluster 수를 증가시킴에 따라 정확도가 증가하는 것을 볼 수 있다.

세 가지를 종합해보면 색 구분이 뚜렷하고, 그림보다는 사진을 이용하며 cluster 수를 증가시킨다면 성능이 좋아질 것이다.

3. 참여 인원별 역할 및 수행 소감

- **박민혁** : 이번 학기에만 팀 과제를 세 개나 수행하는 중인데 가장 어려움을 느꼈습니다. MATLAB이 익숙하지 않아서였던 것 같습니다. 그래도 팀원들과 함께 역할을 나누고 자신이 맡은 역할에 대해 책임감 있게 해서 만족스러운 결과물을 만들어 낸 것 같습니다. 또한 MATLAB이라는 프로그램을 알게 되어서 좋은 경험을 했다고 생각합니다.

- **이상민** : 팀 프로젝트가 주어졌을 때 MATLAB이라는 프로그램을 사용하는 데 두려움이 많았다. 교수님께서 직관적인 코딩을 할 수 있다고 거듭 강조하셨는데, 아무래도 처음 접하는 프로그램이었기 때문에 그랬던 것 같다. 우선 Mathwork 사이트에서 방대한 양의 예제들을 접했다. 처음에는 얼굴 인식을 주제로 진행하려고 했었는데, 실제로 코드를 본 결과 굉장히 난이도가 높았다. 난이도가 너무 높은 주제를 삼으면 정확히 이해하지 못하고 그저 따라하는 수준으로 끝날 것 같아 주제를 재선정하였다.
분석한 코드를 간략하게 설명하면 다음과 같다. 우선 rgb로 되어있는 이미지를 lab로 바꿔 저장을 한다. 그리고 K-means clustering을 하는데 이것은 대표적인 분리형 군집화 알고리즘이다. 쉽게 말해 각 개체들은 가장 가까운 중심에 할당되고, 같은 중심에 할당된 개체들이 모여 군집을 형성한다. 이를 통해 이미지를 이루고 있는 색상 성분들의 할당량을 %로 알 수 있었다.
그리고 색상을 추출하여 이미지를 재구성하는 코드는 다음과 같다. rgb로 되어있는 이미지를 hsv로 바꿔 특정 범위를 주면, 마스크를 씌워 범위 외의 색은 흑백처리를 해주는 알고리즘인데, 우리는 6개의 범위를 적용했다. 범위를 더욱 좁게 설정하면 보다 정확한 결과를 도출해낼 수 있었을 것 같다.

- **홍승기** : 대학교에 와서 Visual studio, Android Studio, pycharm 등 프로그래밍 관련 개발환경은 많이 접했었는데 신호처리라는 과목을 들으면서 정말 처음으로 MATLAB이라는 환경을 배웠는데 생각 이상으로 좋은 프로그램이라고 프로젝트를 하는 동안 느꼈고, Mathwork라는 좋은 사이트도 알게 되었다. 프로젝트에서 나의 역할이 코드를 재구성하는 것 이여서, 대강 프로젝트의 기반이 된 코드를 봤을 땐 코드가 엄청 어려운 것 같다고 생각이 들었었지만, 확실히 교수님 말씀대로 Matlab 자체가 굉장히 직관적인 언어이고, 구현되어있는 함수가 많다고 느낀 것이, 어려울 것 같다고 생각한 대부분 코드가 읽으면 무슨 역할을 하는 코드인지 바로 해석이 되고, 처음 보는 함수가 있으면 help라는 명령어를 통해서 무슨 역할을 하는 함수인지

바로 확인할 수 있었다. 그래도 이해가 안 되는 함수는 함수사용 예제를 통해서 학습 하면서 익힐 수 있었다. 그래서 빠르게 어떤 코드가 어떠한 역할을 하는지 파악할 수 있었고, 우리가 하는 프로젝트에 어떤 코드가 필요 없는지 외부함수로 어떤 코드들을 옮겨야 하는지 쉽게 진행할 수 있었다. 그리고 Matlab 자체에 구현되어 있는 함수가 굉장히 다양하다고 확실히 느꼈고, 왜 교수님이 C언어에 비해 쓰는 사람이 쉽게 배울만한 언어라고 하셨는지 이해가 됐다. 이전에는 수업이나, 프로젝트를 하면서 항상 C언어나 Java 등만 사용했었는데, 이번 Matlab 프로젝트를 통해서 정말 좋은 경험을 한 것 같고, 앞으로 다른 프로젝트나 과제를 할 때 유용하게 쓸 수 있을 것 같다.

4. Project 후기

- 대학을 다니면서 MATLAB을 처음 접했다. 기본 사용법만 배우고 프로젝트를 진행하는 것이 우리에겐 상당히 어려운 일이었다. 그래서 다소 긴장한 상태에서 프로젝트를 시작했다. 처음에는 face recognition을 주제로 프로젝트를 진행하려고 했다. 하지만 코딩 이해에 어려움이 있어 우리 팀은 많은 고민 끝에 image filtering으로 주제를 바꾸었다.

주제를 바꾸고 처음 접한 것이 특정 색을 제외한 나머지 색은 흑백으로 처리하는 알고리즘을 접했다. 그렇게 어려운 난이도가 아니기에, 다른 image processing 알고리즘과 접목시키고 싶었다. Mathwork 사이트에서 계속 검색해본 결과 사진에서 사용된 색을 추출하여 추출한 색을 가지고 image filtering을 하자는 의견이 나왔다. 이것을 토대로 시작을 해서 현재의 결과물을 만들게 되었다. 의견은 좋았지만 두 개의 코딩을 합치는 것 자체가 어려웠고 계속적인 시행착오가 있었다. 하지만 팀원들끼리 합심하여 성공했고 처음 접한 프로그램에서 이정도의 결과물을 만들었다는 사실에 굉장히 만족스러웠다. 또한 MATLAB은 오픈소스가 다양하고, 이를 사용할 경우 다른 프로그램에 비해 상대적으로 오류가 적었다. 따라서 처음 접하는 프로그램임에도 불구하고 조금 더 효율적으로 프로젝트를 성공시킬 수 있었다.

5. Appendix (Project MATLAB source code)

main.m

```
%% 이미지를 읽어 image_palette 함수 호출
nCluster = 4;               % 추출하고 싶은 색 성분 개수
fname ='dsa.jpg';          % 이미지 파일

image_palette( nCluster, fname );
```

get_color.m

```
%% color_list.m으로부터 색상 이름 전달받음
function [matching_label, min_error] = get_color( input )

if size(input,1)>size(input,2)
    input=input';
end
[rgb, labels] = color_list();
input_rep = repmat( input, [size(rgb,1), 1]);
[min_error,matchIdx] = min( mean((input_rep-rgb).^2, 2) );
matching_label = labels{ matchIdx };
end
```

color_filter.m

```
%% 지정한 HSV의 H 범위를 이용하여 특정 색상 외의 나머지 색 필터링
function I = color_filter(image, range)
    % RGB를 HSV로 변환
    I = rgb2hsv(image);

    % 0 ~ 1 사이 범위로 normalization
    range = range./360;

    % 마스크 생성
    if(size(range,1) > 1), error('Error. Range matriz has too many rows.');
end
    if(size(range,2) > 2), error('Error. Range matriz has too many
columns.'); end

    % 범위에 따라 마스크 부여
    if(range(1) > range(2))
        mask = (I(:,:,1)>range(1) & (I(:,:,1)<=1)) + (I(:,:,1)<range(2) &
(I(:,:,1)>=0));
    else
        mask = (I(:,:,1)>range(1)) & (I(:,:,1)<range(2));
    end

    I(:,:,2) = mask .* I(:,:,2);

    % HSV를 다시 RGB로 변환
    I = hsv2rgb(I);
end
```

color_list.m

```
%% RGB 표에 의한 색상 정보
function [rgb, labels] = color_list()
labels = {...
    'Pink'
    'LightPink'
    'HotPink'
    'DeepPink'
    'PaleVioletRed'
    'MediumVioletRed'
    'LightSalmon'
    'Salmon'
    'DarkSalmon'
    'LightCoral'
    'IndianRed'
    'Crimson'
    'Firebrick'
    'DarkRed'
    'Red'
    'OrangeRed'
    'Tomato'
    'Coral'
    'DarkOrange'
    'Orange'
    'Yellow'
    'LightYellow'
    'LemonChiffon'
    'LightGoldenrodYellow'
    'PapayaWhip'
    'Moccasin'
    'PeachPuff'
    'PaleGoldenrod'
    'Khaki'
    'DarkKhaki'
    'Gold'
    'Cornsilk'
    'BlanchedAlmond'
    'Bisque'
    'NavajoWhite'
    'Wheat'
    'Burlywood'
    'Tan'
    'RosyBrown'
    'SandyBrown'
    'Goldenrod'
    'DarkGoldenrod'
    'Peru'
    'Chocolate'
    'SaddleBrown'
    'Sienna'
    'Brown'
```

```
                    'Turquoise',
'Maroon',           'MediumTurquoise',  'Fuchsia',
'DarkOliveGreen',   'DarkTurquoise',    'Magenta',
'Olive',            'LightSeaGreen',    'MediumOrchid',
'OliveDrab',        'CadetBlue',        'MediumPurple',
'YellowGreen',      'DarkCyan',         'BlueViolet',
'LimeGreen',        'Teal',             'DarkViolet',
'Lime',             'LightSteelBlue',   'DarkOrchid',
'LawnGreen',        'PowderBlue',       'DarkMagenta',
'Chartreuse',       'LightBlue',        'Purple',
'GreenYellow',      'SkyBlue',          'Indigo',
'SpringGreen',      'LightSkyBlue',     'DarkSlateBlue',
'MediumSpringGreen','DeepSkyBlue',      'SlateBlue',
'LightGreen',       'DodgerBlue',       'MediumSlateBlue',
'PaleGreen',        'CornflowerBlue',   'White',
'DarkSeaGreen',     'SteelBlue',        'Snow',
'MediumAquamarine', 'RoyalBlue',        'Honeydew',
'MediumSeaGreen',   'Blue',             'MintCream',
'SeaGreen',         'MediumBlue',       'Azure',
'ForestGreen',      'DarkBlue',         'AliceBlue',
'Green',            'Navy',             'GhostWhite',
'DarkGreen',        'MidnightBlue',     'WhiteSmoke',
'Aqua',             'Lavender',         'Seashell',
'Cyan',             'Thistle',          'Beige',
'LightCyan',        'Plum',             'OldLace',
'PaleTurquoise',    'Violet',           'FloralWhite',
'Aquamarine',       'Orchid',           'Ivory',
```

```
    'AntiqueWhite'      220 20  60      210 180 140
    'Linen'             178 34  34      188 143 143
    'LavenderBlush'     139 0   0       244 164 96
    'MistyRose'         255 0   0       218 165 32
    'Gainsboro'         255 69  0       184 134 11
    'LightGray'         255 99  71      205 133 63
    'Silver'            255 127 80      210 105 30
    'DarkGray'          255 140 0       139 69  19
    'Gray'              255 165 0       160 82  45
    'DimGray'           255 255 0       165 42  42
    'LightSlateGray'    255 255 224     128 0   0
    'SlateGray'         255 250 205     85  107 47
    'DarkSlateGray'     250 250 210     128 128 0
    'Black'             255 239 213     107 142 35
    };                  255 228 181     154 205 50
rgb = [255  192 203     255 218 185     50  205 50
    255 182 193         238 232 170     0   255 0
    255 105 180         240 230 140     124 252 0
    255 20  147         189 183 107     127 255 0
    219 112 147         255 215 0       173 255 47
    199 21  133         255 248 220     0   255 127
    255 160 122         255 235 205     0   250 154
    250 128 114         255 228 196     144 238 144
    233 150 122         255 222 173     152 251 152
    240 128 128         245 222 179     143 188 143
    205 92  92          222 184 135     102 205 170
```

```
60  179 113
46  139 87
34  139 34
0   128 0
0   100 0
0   255 255
0   255 255
224 255 255
175 238 238
127 255 212
64  224 208
72  209 204
0   206 209
32  178 170
95  158 160
0   139 139
0   128 128
176 196 222
176 224 230
173 216 230
135 206 235
135 206 250
0   191 255
30  144 255
100 149 237
70  130 180
65  105 225
0   0   255
0   0   205
0   0   139
0   0   128
25  25  112
230 230 250
216 191 216
221 160 221
238 130 238
218 112 214
255 0   255
255 0   255
186 85  211
147 112 219
138 43  226
148 0   211
153 50  204
139 0   139
128 0   128
75  0   130
72  61  139
106 90  205
123 104 238
    245 245 220
    253 245 230
    255 250 240
    255 255 240
    250 235 215
    250 240 230
    255 240 245
    255 228 225
    220 220 220
    211 211 211
    192 192 193
    169 169 169
    128 128 128
    105 105 105
    119 136 153
    112 128 144
    47  79  79
    0   0   0
    ];
end
```

image_palette.m

```
%% main.m에서 받은 이미지를 바탕으로 색상 추출
function image_palette(nCluster, fname, plotOption)

  % 함수 인자가 3개 미만이면 plotOption 값 true
if nargin<3, plotOption = true; end
sz = [200 200];                          % 사진 크기 초기화

%% Load image file
img = rgb2lab(imread(fname));          % 이미지 파일 읽어오기
img = single(img);
L = img(:,:,1);                          % 이미지의 lab 값을 각각 변수에 저장
a = img(:,:,2);
b = img(:,:,3);

%% Visualize image
if plotOption                            % 불러온 이미지 표시
    figure(1);
    subplot(2,1,1);
    imshow( lab2rgb( img ) );
    title(['Source image (resized, ' num2str(sz(1)) 'x' num2str(sz(2)) ')'])
    xlabel(fname);
end

%% Parse and visualize Lab values
img_parse = [];
img_parse(1,:) = L(:);
img_parse(2,:) = a(:);
img_parse(3,:) = b(:);
```

```
%% K-means Clustering
[G] = kmeans( img_parse', nCluster);    % k-평균 군집화 수행
p = [];
for clusterIdx = 1:nCluster                    % 퍼센트 할당
    p(clusterIdx) = 100*length(find(G==clusterIdx)) / length(G);
end
[~,order]=sortrows(p'); order=order(end:-1:1);

% Palette visualization
for clusterIdx = 1:nCluster                    % 값이 작은 것부터 정렬해서 저장
    id = G==order(clusterIdx);
    LAB = ( img_parse(:, id) );
    mean_LAB = reshape( nanmean(LAB,2)', [1, 1, 3]);    % LAB 색상 저장

     % lab 색상을 rgb 색상표에 따라 이름 부여
    [color_label] = get_color(255*lab2rgb(mean(LAB,2)'));
    if plotOption
        subplot(2,nCluster,nCluster+clusterIdx);
        % LAB색상을 rgb 색상으로 변환후 표시
        imshow( lab2rgb( mean_LAB ) );
        title(['C_{' num2str(clusterIdx) '}
(' num2str(round(100*p(order(clusterIdx))/100)) ' %)' ]);
        xlabel(color_label);
    end
end
```

```
%% 6가지 색상별로 추출하여 이미지 재구성
if plotOption
    figure(2)
            I = imread(fname);
            I = im2double(I);
            I_m = color_filter(I,[350 30]);
            subplot(321)
            imshow(I_m,[]);
            title('Red hue filtering');
            I_m2 = color_filter(I, [31 60]);
            subplot(322)
            imshow(I_m2,[]);
            title('Yellow hue filtering');
            I_m3 = color_filter(I, [61, 130]);
            subplot(323)
            imshow(I_m3,[]);
            title('Green hue filtering');
            I_m4 = color_filter(I, [131 190]);
            subplot(324)
            imshow(I_m4,[]);
            title('Skyblue hue filtering');
            I_m5 = color_filter(I, [191 280]);
            subplot(325)
            imshow(I_m5,[]);
            title('Blue hue filtering');
            I_m6 = color_filter(I, [281 349]);
            subplot(326)
            imshow(I_m6,[]);
            title('Pink hue filtering');
end
end
```