Table of Contents

COUNTING VARIOUS OBJECTS

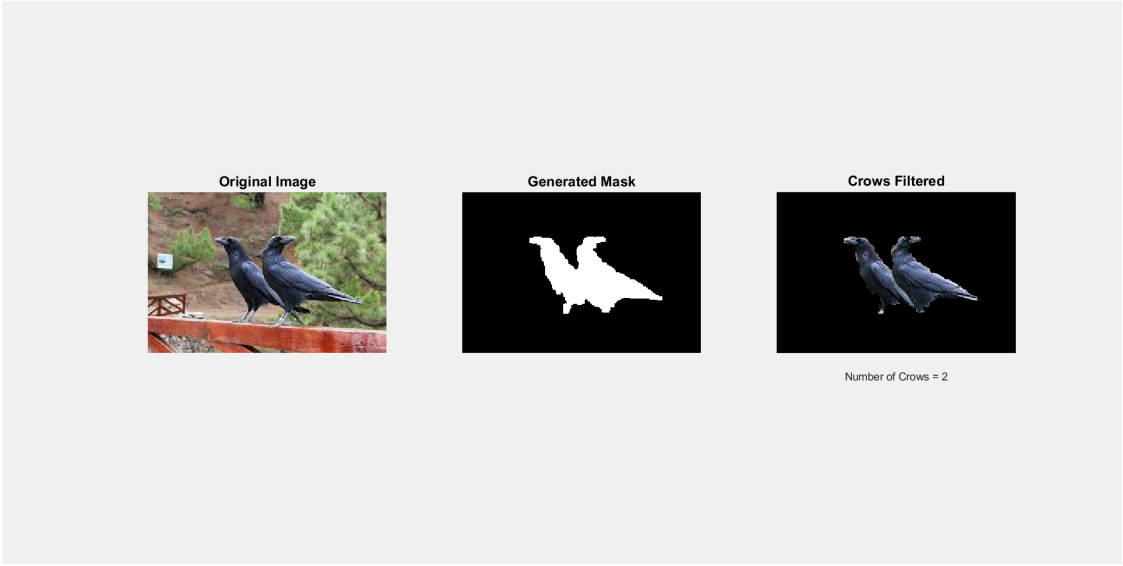
```
close all; clearvars; clc;

crows = imread('crows.jpg');
countCrows(crows);

eggs = imread('eggs.jpg');
countEggs(eggs);

bottles = imread('bottles.jpg');
countBottles(bottles);
```

Counting Crows

```
function countCrows(image)

% Reading the image and conversion to Double
image = im2double(image);

% Counting Crows
numCrows = num_of_crows(image);

% Displaying the result
display_crows(image,numCrows);

end

function numCrows = num_of_crows(image)

% Converting To Grayscale
grayImage = rgb2gray(image);

% Inverting Colors
grayImage = 1 - grayImage;

% Thresholding
grayImage(grayImage<0.6) = 0;
grayImage(grayImage>0.6) = 1;

% Creating binary mask
bwImage = imbinarize(grayImage);

% Removing very small connected components
```

```
bwImage = imerode(bwImage, strel('disk',1));

% Finding the connected components
CC = bwconncomp(bwImage);
n = CC.PixelIdxList;

% Storing the sizes of all connected components
num = zeros(length(n), 1);
for k = 1:length(n)
    num(k) = length(n{k});
end

% Finding the indices of the 5 biggest connected components
[~, sortedIndices] = sort(num, 'descend');
fiveLargestIndices = sortedIndices(1:5);

% Calculating the average number of pixels in the 5 largest connected
 components
totalPixels = 0;
for k = 1:5
    totalPixels = totalPixels + length(n{fiveLargestIndices(k)});
end

% Calculating average number of pixels of 5 largest CCs
averagePixels = round(totalPixels / 5);

% Filtering based on the calculated average
bwImage = bwareaopen(bwImage, averagePixels);

% Number of Crows
CC = bwconncomp(bwImage);
numCrows = CC.NumObjects;

end

function display_crows(image, numCrows)

% Converting to HSV
hsv_img = rgb2hsv(image);

% Extracting the 1st channel
Hue = hsv_img(:, :, 1);

% Thresholding
binaryImage = Hue > 0.5;  % 0.5 is a good number for pictures of crows

% Performing morphological operations
se = strel('disk', 3);
binaryImage = imopen(binaryImage, se);
binaryImage = imfill(binaryImage, 'holes');

% Finding the connected components
CC = bwconncomp(binaryImage);
n = CC.PixelIdxList;
```

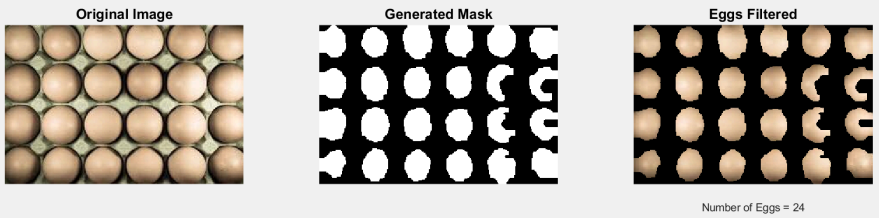
```
% Storing the sizes of all connected components
num = zeros(length(n), 1);
for k = 1:length(n)
    num(k) = length(n{k});
end

% Finding the indices of the 5 biggest connected components
[~, sortedIndices] = sort(num, 'descend');
fiveLargestIndices = sortedIndices(1:5);

% Calculate the difference in number of pixels between adjacently
 largest
% connected components
pixelDifferences = zeros(4, 1);
for i = 1:4
    indx1 = fiveLargestIndices(i);
    indx2 = fiveLargestIndices(i+1);
    pixelDiff = abs(num(indx1) - num(indx2));
    pixelDifferences(i) = pixelDiff;

    % For pixel difference values greater than 4200 (larger connected
    % components do not differ that much)
    if pixelDiff > 4200
        % Creating binary mask with only the connected components
 before
        % the exceeding difference value
        filteredImage = false(size(binaryImage));
        for k = 1:i
            filteredImage(n{fiveLargestIndices(k)}) = true;
        end

        % End the loop
        break;
    end
end

% Displaying the Results
newImage = image.*filteredImage;
figure('windowstate','maximized')
subplot(131), imshow(image), title('Original Image','fontsize',18)
subplot(132), imshow(filteredImage), title('Generated
 Mask','fontsize',18)
subplot(133), imshow(newImage), title('Crows Filtered','fontsize',18),
 xlabel(['Number of Crows = ',num2str(numCrows)],'fontsize', 14)

end
```



Counting Eggs

```
function countEggs(image)

% Reading the image and conversion to Double
image = im2double(image);

% Generating Binary Mask
binaryImage = eggsMask(image);

% Counting Eggs
numEggs = num_of_eggs(binaryImage);

% Displaying the result
display_eggs(image,binaryImage,numEggs);

end

function binaryImage = eggsMask(image)

ycbcrImage = rgb2ycbcr(image);

Cr = ycbcrImage(:,:,3);

binaryImage = Cr > 0.535;

se = strel('line',10,0);
binaryImage = imerode(binaryImage,se);
se = strel('disk',1);
binaryImage = imerode(binaryImage,se);
se = strel('disk',3);
binaryImage = imopen(binaryImage,se);
binaryImage = imfill(binaryImage, 'holes');
```

```
end

function numEggs = num_of_eggs(binaryImage)

CC = bwconncomp(binaryImage);
numEggs = CC.NumObjects;

end

function display_eggs(image, binaryImage, numEggs)

newImage = image.*binaryImage;
figure('windowstate', 'maximized')
subplot(131), imshow(image), title('Original Image','fontsize',18)
subplot(132), imshow(binaryImage), title('Generated
 Mask','fontsize',18)
subplot(133), imshow(newImage), title('Eggs Filtered','fontsize',18),
 xlabel(['Number of Eggs = ',num2str(numEggs)],'fontsize', 14)

end
```

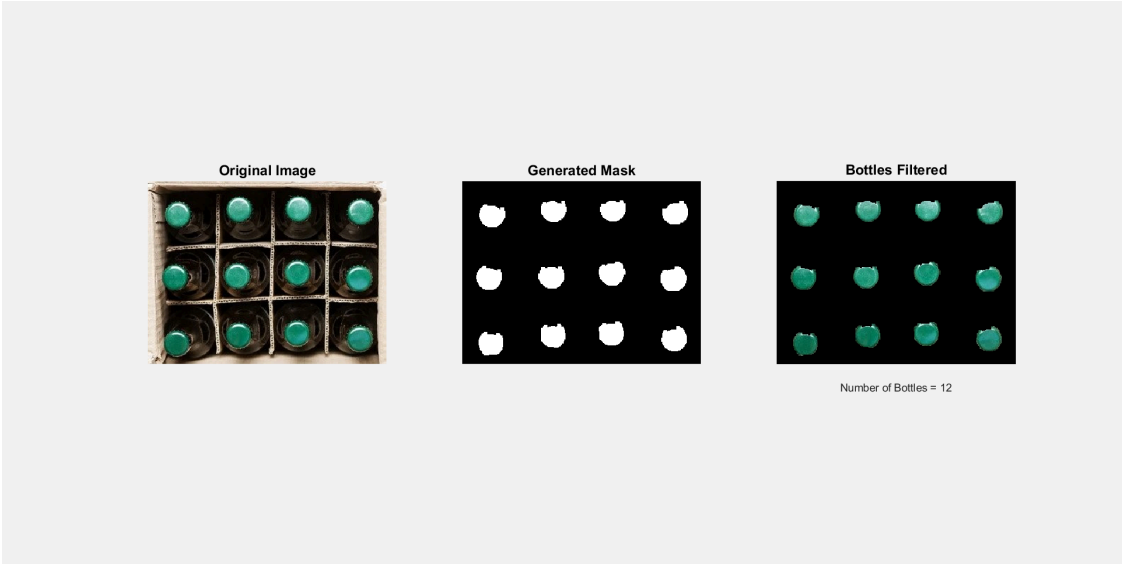


Counting Bottles

```
function countBottles(image)

% Reading the image and conversion to Double
image = im2double(image);

% Generating Binary Mask
binaryImage = bottlesMask(image);

% Counting Bottles
numBottles = num_of_bottles(binaryImage);
```

```
% Displaying the result
display_bottles(image,binaryImage,numBottles);

end

function binaryImage = bottlesMask(image)

% Converting to HSV
hsvImage = rgb2hsv(image);

% Extracting 1st Channel
hue = hsvImage(:,:,1);
saturation = hsvImage(:, :, 2);
value = hsvImage(:, :, 3);

% Defining thresholding range according to Green color
hue_low = 0.3529;   % 90
hue_high = 0.5850;  % 150
saturation_low = 0.2;
saturation_high = 1;
value_low = 0.2;
value_high = 1;

% Thresholding the image to extract the Green color
binaryImage = (hue >= hue_low) & (hue <= hue_high) & ...
              (saturation >= saturation_low) & (saturation <=
 saturation_high) & ...
              (value >= value_low) & (value <= value_high);

% Performing Morphological Opening to filter small Connected
 Components
se = strel('disk',3);
binaryImage = imopen(binaryImage, se);

end

function numBottles = num_of_bottles(binaryImage)

CC = bwconncomp(binaryImage);
numBottles  = CC.NumObjects;

end

function display_bottles(image, binaryImage, numBottles)

newImage = image.*binaryImage;
figure('windowstate', 'maximized')
subplot(131), imshow(image), title('Original Image','fontsize',18)
subplot(132), imshow(binaryImage), title('Generated
 Mask','fontsize',18)
subplot(133), imshow(newImage), title('Bottles
 Filtered','fontsize',18), xlabel(['Number of Bottles =
 ',num2str(numBottles)],'fontsize', 14)
```

```
end
```



Published with MATLAB® R2021a