

**COMSATS UNIVERSITY ISLAMABAD, ABBOTTABAD**

**Course: Digital Image Processing (CSC331)**

**Semester Project:**

Color-Based Object Recognition on a Map Image

***Submitted by:***

Laiba binte tahir FA21-BSE-019

**Instructor: Dr. Mubashir Ahmad**

Project Overview: Color-Based Object Recognition on a Map Image

# Objective:

The primary goal of this project is to recognize and segment various regions in a map image based on their color. By leveraging color-based segmentation techniques, the project aims to identify and label different continents and countries.

# Techniques Used:

1. **Color Space Conversion:**
   * The RGB color space, commonly used in digital images, is converted to the HSV (Hue, Saturation, Value) color space. This transformation simplifies the process of color segmentation, as HSV separates chromatic content (hue) from intensity (value), making it easier to define and manipulate specific colors.
2. **Thresholding:**
   * Thresholding involves setting minimum and maximum values for the hue, saturation, and value channels to create a binary mask. This mask identifies pixels in the image that fall within the specified color range. Thresholding effectively isolates regions of interest based on their colors.
3. **Morphological Operations:**
   * Morphological operations like hole filling (imfill) and small object removal (bwareaopen) refine the binary mask. These operations clean up the segmented regions by removing noise and filling gaps, leading to more accurate and coherent region identification.
4. **Labeling and Display:**
   * Connected components in the binary mask are labeled, and their properties (such as bounding boxes and centroids) are identified using regionprops. The original image is then displayed with bounding boxes and centroids overlaid to highlight and label the segmented objects.

# Importance of the Project:

* **Automated Analysis:**
  + The project demonstrates an automated approach to analyzing images based on color. This technique is applicable in various fields such as geographic information systems (GIS), medical imaging, satellite imagery analysis, and object detection in computer vision.
* **Color-Based Segmentation:**
  + Segmenting images based on color is a fundamental technique in image processing. It can be used in applications like traffic sign recognition, agricultural monitoring, and content-based image retrieval. In this project, it enables the identification of different regions on a map based on their color.
* **Educational Tool:**
  + This project serves as a practical example for understanding color space transformations, thresholding, and morphological operations. It's a valuable teaching tool for courses in image processing and computer vision, illustrating the practical implementation of these techniques.

Finding Continents on a Map

By applying specific color thresholds, the project can segment different regions on a map based on their colors. Here are the regions segmented and their corresponding colors:

* North America: Green
* South America: Yellow
* Europe: Orange
* Africa: Red
* Asia: Blue
* Australia: Purple
* Antarctica: Not present in this image

**Example of Segmentation and Labeling:**

1. **Green Regions (e.g., North America, South America, Australia):**
   * The script identifies and labels green regions on the map, potentially corresponding to continents like North America, South America, and Australia.
2. **Yellow Regions (e.g., Europe):**
   * Europe is segmented based on its yellow color. The script isolates and labels this region, highlighting it on the map.
3. **Blue Regions (e.g., Africa):**
   * Africa is segmented based on its blue color. The script identifies and labels this region, highlighting it on the map.
4. **Purple Regions (e.g., Asia):**
   * Asia is segmented based on its purple color. The script identifies and labels this region, highlighting it on the map.
5. **Red and Orange Regions (e.g., China, India):**
   * China and India are segmented based on their respective red and orange colors. The script identifies and labels these regions, highlighting them on the map.

# Conclusion:

This project demonstrates the practical application of image processing techniques to recognize and segment objects based on color. By converting the image to HSV, applying color thresholds, and refining the segmentation with morphological operations, we can effectively isolate and label different regions in an image. This approach is versatile and can be adapted to various applications where color is a distinguishing feature.

# Code

% Clear console and variables

clc;

clear;

close all;

% Load the uploaded image

image = imread('map2.jfif'); % Ensure the path is correct

% Convert image to double precision for processing

image = im2double(image);

% Convert the image from RGB to HSV color space

hsvImage = rgb2hsv(image);

% Define a cell array of color thresholds and labels

colors = {

'North America', 0.2, 0.4, 0.2, 1.0, 0.2, 1.0; % Green

'South America', 0.12, 0.18, 0.4, 1.0, 0.4, 1.0; % Yellow

'Europe', 0.05, 0.10, 0.4, 1.0, 0.4, 1.0; % Orange

'Africa', 0.95, 1.0, 0.4, 1.0, 0.4, 1.0; % Red

'Asia', 0.55, 0.65, 0.4, 1.0, 0.4, 1.0; % Blue

'Australia', 0.70, 0.85, 0.4, 1.0, 0.4, 1.0; % Purple

};

% Create a figure to display results

figure;

% Display the original image in the first subplot

subplot(1, 2, 1);

imshow(image);

title('Original Image');

% Display the segmented image in the second subplot

subplot(1, 2, 2);

imshow(image);

title('Image with Color Segmentation');

hold on;

% Loop through each color and segment the image

for i = 1:size(colors, 1)

% Get the color thresholds

label = colors{i, 1};

channel1Min = colors{i, 2};

channel1Max = colors{i, 3};

channel2Min = colors{i, 4};

channel2Max = colors{i, 5};

channel3Min = colors{i, 6};

channel3Max = colors{i, 7};

% Create mask based on chosen histogram thresholds

binaryMask = (hsvImage(:,:,1) >= channel1Min) & (hsvImage(:,:,1) <= channel1Max) & ...

(hsvImage(:,:,2) >= channel2Min) & (hsvImage(:,:,2) <= channel2Max) & ...

(hsvImage(:,:,3) >= channel3Min) & (hsvImage(:,:,3) <= channel3Max);

% Use morphological operations to refine the segmentation

binaryMask = imfill(binaryMask, 'holes');

binaryMask = bwareaopen(binaryMask, 100);

% Label connected components

labeledImage = bwlabel(binaryMask);

stats = regionprops(labeledImage, 'BoundingBox', 'Centroid');

% Display bounding boxes and centroids

for k = 1:length(stats)

bbox = stats(k).BoundingBox;

centroid = stats(k).Centroid;

rectangle('Position', bbox, 'EdgeColor', 'r', 'LineWidth', 2);

plot(centroid(1), centroid(2), 'b\*');

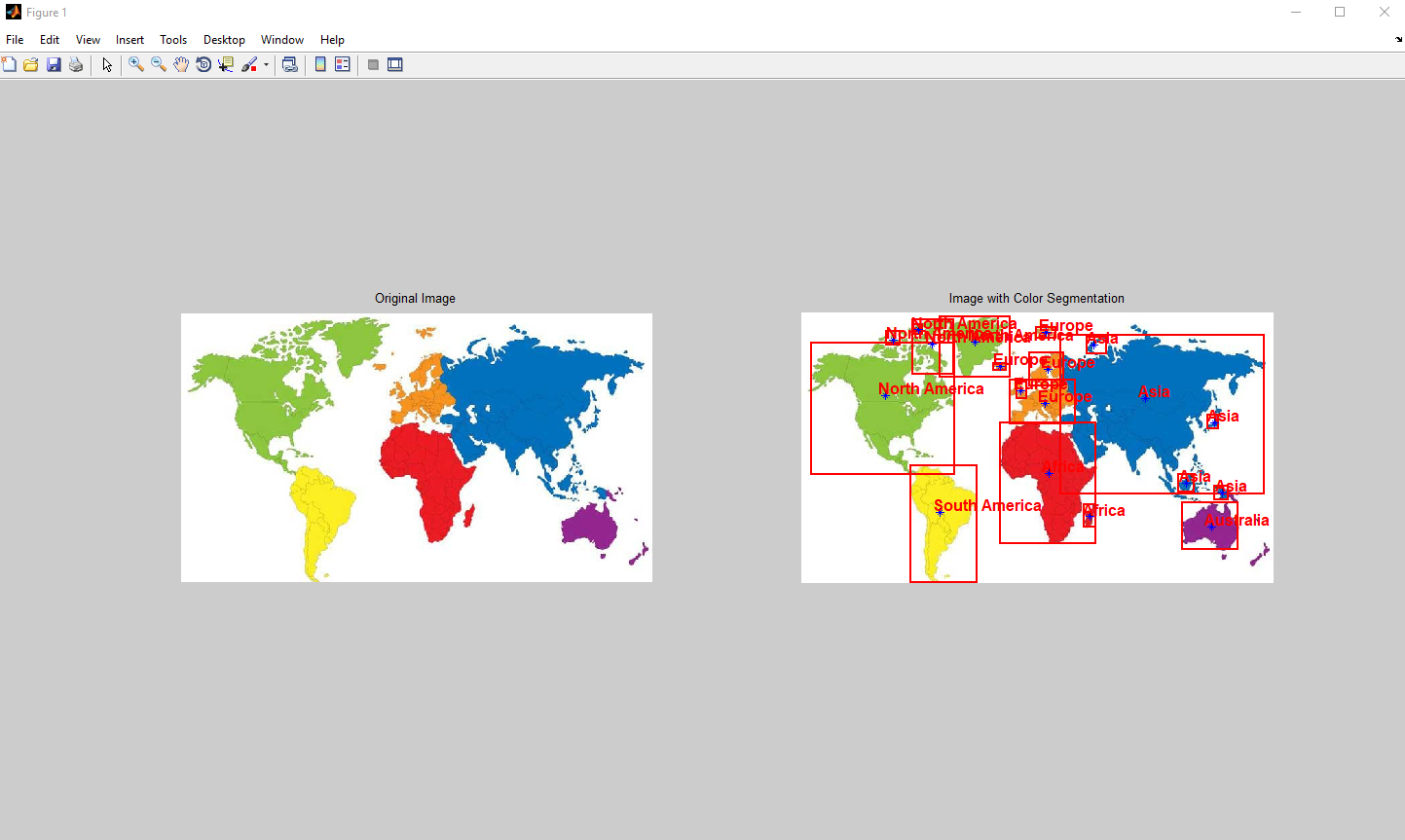
text(centroid(1) - 10, centroid(2) - 10, sprintf('%s', label), 'Color', 'red', 'FontSize', 12, 'FontWeight', 'bold');

end

end

hold off;

# output



A screenshot of a computer

Description automatically generated