ISSN: 2249-7137 Vol. 10 Issue 12, December 2020 Impact Factor: SJIF 2020 = 7.13

ACADEMICIA
An International Multidisciplinary Research Journal
(Double Blind Refereed & Peer Reviewed Journal)

DOI: 10.5958/2249-7137.2020.01735.8

IMAGE SEGMENTATION IN OPEN CV AND PYTHON

Xasanov Dilmurod Rasul ogli*; Tojiyev Maruf Ruzikulovich*; Primqulov Oybek Dilmurot Ogli*

*Jizzakh branch of National University,
UZBEKISTAN
Email id: tatusf2015@gmail.com

ABSTRACT

Object detection and face recognition in image are essential major in this century. Because, every part of industries need security system, service system has to do comfortable their services. To detect object applies some methods of computer vision. Computer vision is one most important part of artificial intelligence. Segmentation considers basic method of computer vision. Segmantation and filtring images change interface of image, also separate objects in image. Below all method of segmantation image showed in this article.

KEYWORDS: *Image Segmantation, Open Source Computer Vision, Frames, Color Space Model, AI.*

INTRODUCTION

Now, perhaps the era of deep learning and big data has arrived, where complex algorithms analyze images based on showing them millions of examples, but color spaces are still great for analysis. Simple techniques remain extremely powerful. In this article, you will learn how to easily segment an object from an image based on color using Python and OpenCV. A popular computer vision library written in C / C ++ and related to Python, OpenCV, offers easy ways to manipulate color spaces.

The main part

In the most common color space, RGB (Red Green Blue) colors are represented in terms of their constituent red, green, and blue. More technically, RGB describes a color as a tuple of three components. Each of them can be a value from 0 to 255, where (0, 0, 0) represents black and (255, 255, 255) represents white. RGB is considered an & quot; additive & quot; color space, where colors are the product of red, blue, and green light against a black background.

ISSN: 2249-7137 Vol. 10 Issue 12, December 2020 Impact Factor: SJIF 2020 = 7.13

Here are some more examples of RGB colors:

TABLE-1 RGB COLORS

Colour	RGB value
Red	255,0,0
Orange	255,128,0
Pink	255,153,255

RGB is one of five major color space models, each of which has its own ramifications. There are so many of them, because each space is used for different purposes. In publishing, CMYK is used because it describes the combinations that are required to produce color on a white background. If in RGB tuple 0 is black, then in CMYK it is white. Therefore, printers contain cyan, magenta, yellow, and black cartridges. In certain areas of medicine, glass slides containing stained tissue samples are scanned. They are saved as images. These are analyzed in the HED space - representation of the saturation of the types of paint - Hematoxylin, Eosin and Diaminobenzidine. HSV and HSL are descriptions of hue, saturation, and brightness. They are suitable for detecting contrast in images. These spaces are often used for color selection in web design software.

In fact, color is a continuous phenomenon, which means that there are an infinite number of them. Color spaces, however, depict colors as discrete structures (a fixed integer), which is quite acceptable, because the perception of the human eye is also limited. Color spaces are capable of rendering all colors that we distinguish. Now that everything is clear with color spaces, you can move on to describing how to use them in Open CV.

Simple segmentation with color spaces

To demonstrate the segmentation technique based on color spaces, we will use a set of clown fish images. They are easily recognizable due to their bright orange color, so we will look for Nemo in the picture. The key libraries required are the advanced scientific computing package NumPy , the Matplotlib plotting tool, and of course OpenCV. This example uses OpenCV 3.2.0, NumPy 1.12.1 and Matplotlib 2.0.2 versions. But small differences in versions should not greatly affect the understanding of the very concept of work.

Color spaces and reading images in Open CV

First of all, you need to set up the space. Python 3.x should already be installed on your system. Please note that despite using OpenCV 3.x, you still need to

Import cv2:

Import cv2

If the library is not already installed on your computer, the import will not work.

Here you can view the installation guide for different operating systems. After import, you can see all the color space transformations that are available in Open CV, and save them all to a variable:

Flags = [i for i in dir(cv2) if i. starts with('COLOR_')]

The list and number of flags may differ slightly depending on the version of

ISSN: 2249-7137 Vol. 10 Issue 12, December 2020 Impact Factor: SJIF 2020 = 7.13

Open CV, but there will be many in any case. Just look at the total:

Len (flags)

258

flags [40]

'COLOR_BGR2RGB'

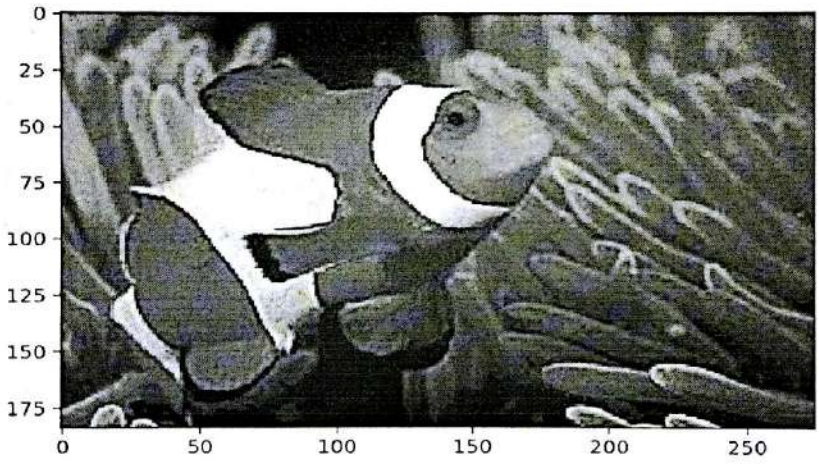
The first characters after the COLOR original color space, and the characters after the 2target. This flag represents the conversion from BGR (Blue, Green, Red) to RGB. These spaces are very similar. They only have different values of the first and last channels. To view images you will need Matplotlib .pyplotNumPy as well. If they are not installed, use the command pip3 install matplotliband pip3 install numpyprior to import:

Archive with images:

Import matplotlib.pyplot as plt

Import numpy as np

Now you can download and explore the image. Please note that if you are working from the command line or terminal, the images will appear in a pop-up window. If it is Jupyter Notebook or something else, then they will be displayed below. Regardless of the setting, you will see the image generated by the command show(): nemo=cv2.imread('./images/image_1.jpg') plt.imshow(nemo) plt.show()



Figure–1

HSV is a great choice for color segmentation, but to find out why this is, you need to compare RGB and HSV to see the color distribution of their pixels. The 3D graph demonstrates this perfectly. Each axis here represents one of the color space channels. Here is a plot of the color scattering of a Nemo image in RGB

(Fig. 1.).

Here you can see that the orange pixels are spread across all the red, green and blue values. So segmenting Nemo in RGB space based on RGB values is not an easy task at all.

ISSN: 2249-7137 Vol. 10 Issue 12, December 2020 Impact Factor: SJIF 2020 = 7.13

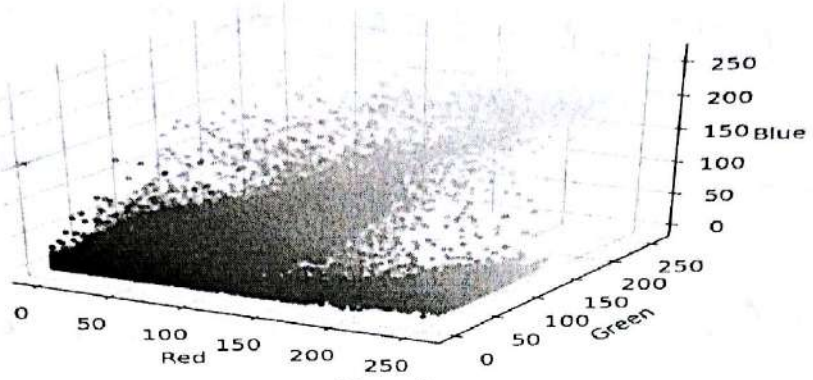


Figure-2.

Rendering Nemo in HSV color space

Now let's see how Nemo will look in HSV space and compare the results. As mentioned, HSV is Hue, Saturation, and Value (hue, saturation, and brightness). This is a cylindrical color space. Colors, or shades, change as you move around the cylinder. The vertical axis is responsible for brightness: from dark (0 at the bottom) to light at the top. The third axis, saturation, determines the shade shadows as we move from center to edge along the radius of the cylinder (from less to more saturated)

To convert from RGB to HSV you can use cvtColor():

hsv_nemo = cv2.cvtColor(nemo, cv2.COLOR_RGB2HSV)

Now hsv_nemostores Nemo's representation in HSV. Using the already familiar technique, you can look at the graph of the image in HSV (Fig. 3.):

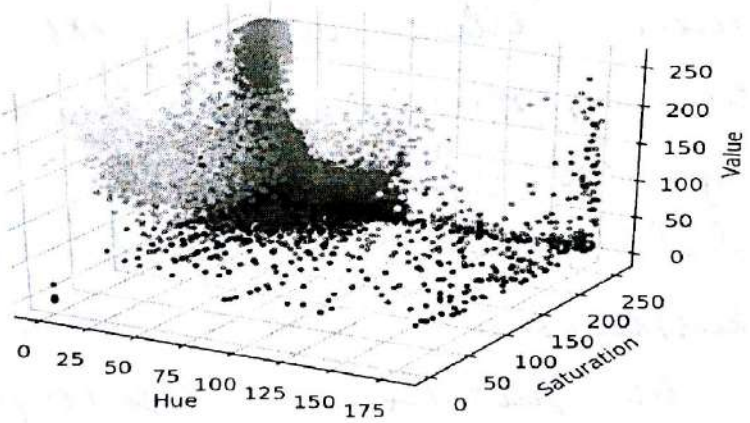


Figure-3

In HSV space, Nemo's orange color is more localized and visually distinguishable. The saturation and values of orange vary, but they have a small range along the hue axis. This is a key point that can be used for segmentation.

ISSN: 2249-7137 Vol. 10 Issue 12, December 2020 Impact Factor: SJIF 2020 = 7.13

REFERENCES:

1. Gonzalez Rafael C.,Richard Woods E. Digital Image Processing – 2nd edition.- 2012 – pg. 11-28.

2. Anil Jain K., Fundamentals of digital image processing. 2008, pg. 8 – 112.

3. James F. Peters, "Foundations of Computer Vision". 2017, pg. 1 – 7.

4. Richard Szeliski, "Computer vision: Algorithms and applications".