

Image Processing in OpenCV

Code Repository :- <https://github.com/Shiven-saini/OpenCV-Samples>

This section is further divided into many subsections :-

- Changing Colorspaces
- Geometric Transformations of Images
- Image Thresholding
- Smoothing Images
- Morphological Transformations
- Image Gradients
- Canny Edge Detection
- Image Pyramids
- Contours
- Histograms
- Image Transforms
- Template Matching
- Hough Line Transform
- Hough Circle Transform
- Image segmentation with Watershed Algorithm
- Foreground Extraction using GrabCut Algorithm

Changing Color-space

Multiple color-space conversion methods are inbuilt in opencv library. The most frequently used are BGR <-> HSV & BGR <-> Gray

Why need to change color-space ?

Because a lot of times, it is easier to use an algorithm in a specific color-space.

`cv.cvtColor(input_source, flag)` : used for color conversion. The first parameter is the image source file, grabbed from `imread`. The second parameter flag suggests the type of conversion.

BGR -> Gray : `COLOR_BGR2GRAY`

BGR -> HSV : `COLOR_BGR2HSV`

- To List out all the available conversion methods available in the library

```
flags = [i for i in dir(cv) if i.startswith('COLOR_')]
```

- Note about HSV usage in OpenCV

For HSV, Hue range is [0, 179], saturation range is [0, 255] and value range is [0, 255]. Different software use different scales. Like Photoshop uses [0, 100] for saturation and value range. So choose and map accordingly.

Trivial usage : Object Tracking (Simplest)

We can use the color-conversion method in our program that is supposed to extract a colored object out of a given image source. In HSV, it is easier to represent color than in BGR. Because in contrast to BGR, we only need Hue and Saturation range to detect a color.

![[Pasted image 20240903191514.png]]

****Approach :-****

- Take each frame of the video.
- Convert BGR to HSV Color-Space
- Threshold the image for a range of desired color.
- Extract the desired color alone, do any operation you intended to do.

****Source Code :-****

```
```python
import numpy as np
import cv2 as cv

cap = cv.VideoCapture(0)
if not cap.isOpened() :
 print("Unable to launch camera!")
 exit()

while 1:
 _, frame_BGR = cap.read()
 frame_HSV = cv.cvtColor(frame_BGR, cv.COLOR_BGR2HSV)

 # Defining HSV Bounds to gather.
 lower_bound = np.array([170, 150, 50])
 upper_bound = np.array([180, 255, 255])

 # frame data only in the given bound
 mask = cv.inRange(frame_HSV, lower_bound, upper_bound)
 result = cv.bitwise_and(frame_BGR, frame_BGR, mask=mask)

 cv.imshow("Video raw", frame_BGR)
 cv.imshow("Mask", mask)
 cv.imshow("Result", result)

 if cv.waitKey(1) == ord('q'):
 break

cap.release()
cv.destroyAllWindows()
```

**Code Analysis :-**

```
frame_HSV = cv.cvtColor(frame_BGR, cv.COLOR_BGR2HSV)
```

To convert BGR Color-space captured frame to HSV.

```
lower_bound = np.array([170, 150, 50])
upper_bound = np.array([180, 255, 255])

mask = cv.inRange(frame_HSV, lower_bound, upper_bound)
```

To filter out lower and upper bound red color only.

```
result = cv.bitwise_and(frame_BGR, frame_BGR, mask=mask)
```

To overlay the red part only over original image.

## References :-

Python OpenCV Docs : [https://docs.opencv.org/4.x/df/d9d/tutorial\\_py\\_colorspaces.html](https://docs.opencv.org/4.x/df/d9d/tutorial_py_colorspaces.html)