



Images and Videos

 학습목표

학습목표

1. Image를 file로부터 읽고 window에 출력한다.
2. Image의 구조를 이해한다.
3. Video를 file로부터 읽고 window에 출력한다.
4. Camera로부터 image를 capture한다.

00 Load and Display Images

Download Sample Images



<https://github.com/opencv/opencv/blob/master/samples/data/messi5.jpg>

Load and Display Images

```
import cv2

# Load an image
img = cv2.imread('messi5.jpg')

# Display the image in a window
cv2.imshow('Lionel Messi', img)

# Wait for a key to be pressed
cv2.waitKey(0)

# Destroy all windows
cv2.destroyAllWindows()
```

- 터미널에서 해당 경로로 이동한 뒤 코드 실행

```
$ cd [path/to/the/file]
$ python [filename]
```

https://docs.opencv.org/4.4.0/db/deb/tutorial_display_image.html

Load and Display Images

```
import cv2
import os

# Get the path to the current file
cwd = os.path.dirname(os.path.abspath(__file__))

# Change the working directory
os.chdir(cwd)

# Load an image
img = cv2.imread('messi5.jpg')

# Display the image in a window
cv2.imshow('Lionel Messi', img)

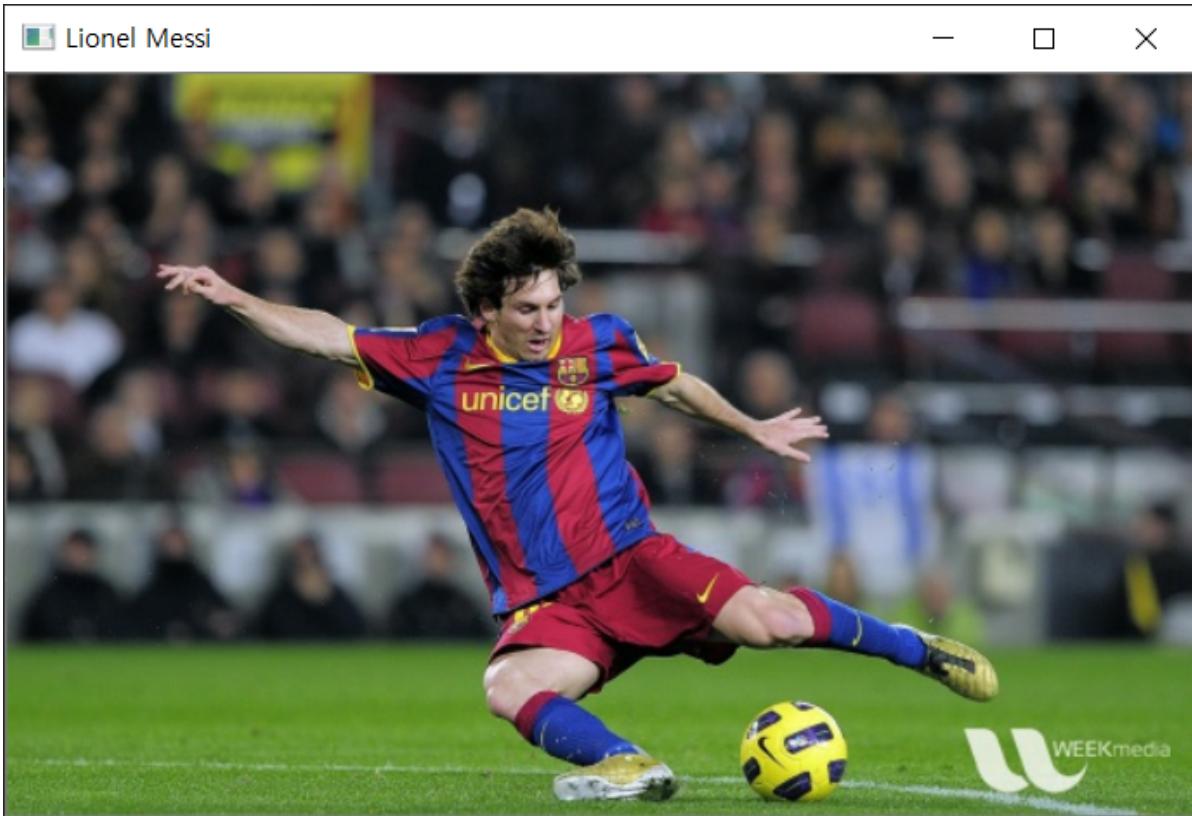
# Wait for a key to be pressed
cv2.waitKey(0)

# Destroy all windows
cv2.destroyAllWindows()
```

- VS CODE에서 Run Python File in Terminal 클릭

https://docs.opencv.org/4.4.0/db/deb/tutorial_display_image.html

Load and Display Images



Load and Display Images in Grayscale

```
import cv2

# Load an image in a grayscale
img = cv2.imread('messi5.jpg', cv2.IMREAD_GRAYSCALE)

# Display the image in a window
cv2.imshow('Lionel Messi in Grayscale', img)

# Wait for a key to be pressed
cv2.waitKey(0)

# Destroy all windows
cv2.destroyAllWindows()
```

Load and Display Images in Grayscale



cv.imread()¹

```
retval = cv.imread(filename[, flags])
```

- File로부터 image를 읽어들인다.
 - filename: image path
 - retval: loaded image, NumPy ndarray

flag ²	영상 읽기 모드
cv.IMREAD_UNCHANGED	원본 그대로 사용
cv.IMREAD_GRAYSCALE	1채널 그레이스케일로 변환
cv.IMREAD_COLOR	3채널 BGR로 변환 (기본값)

1. https://docs.opencv.org/4.4.0/d4/da8/group_imgcodecs.html#ga288b8b3da0892bd651fce07b3bbd3a56
2. https://docs.opencv.org/4.4.0/d4/da8/group_imgcodecs.html#ga61d9b0126a3e57d9277ac48327799c80

cv.imshow()¹

```
None = cv.imshow(winname, mat)
```

- Image를 고유한 window에 display한다.
 - winname: window title
 - mat: image to be displayed, NumPy ndarray

1. https://docs.opencv.org/4.4.0/df/d24/group_highgui_opengl.html#gaae7e90aa3415c68dba22a5ff2cef25d
2. <https://076923.github.io/posts/Python-opencv-3/>

cv.waitKey()¹

```
retval = cv.waitKey([, delay])
```

- Keyboard 입력을 기다린다.
 - `delay`: 대기 시간(ms), 0보다 작거나 같으면 무한히 대기
 - `retval`: 입력된 key 값, 대기 시간동안 입력이 없으면 -1

1. https://docs.opencv.org/4.4.0/d7/dfc/group_highgui.html#ga5628525ad33f52eab17feebcfba38bd7

cv.destroyWindows()¹

```
None = cv.destroyAllWindows()
None = cv.destroyWindow(winname)
```

- 모든 HighGUI window를 닫는다.
- 특정 이름의 window를 닫는다.
 - winname: name of the window to be destroyed

1. https://docs.opencv.org/4.4.0/d7/dfc/group_highgui.html#ga6b7fc1c1a8960438156912027b38f481

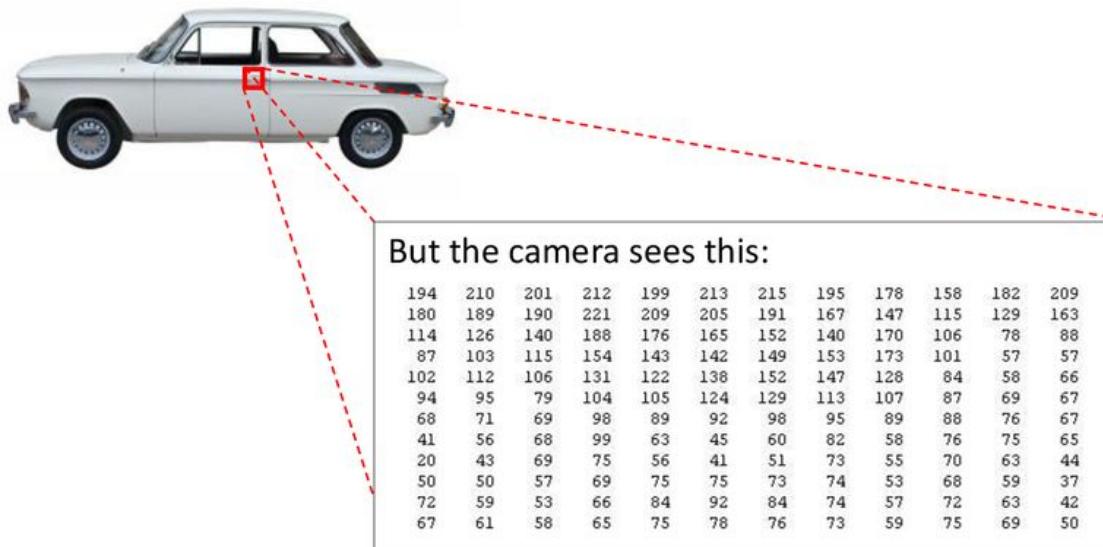


Image Data Structure

Image Data

What is this?

You see this:



Andrew Ng

<https://slideplayer.com/slide/14708730/>

Image Properties

```
import cv2

# Load an image
img = cv2.imread('messi5.jpg')

# Image properties: type
print(type(img))      # <class 'numpy.ndarray'>

# Image properties: number of dimensions
print(img.ndim)       # 3

# Image properties: matrix shape = (rows, cols, channels)
print(img.shape)       # (342, 548, 3)

# Image properties: number of pixels
print(img.size)        # 562248 = 342 x 548 x 3

# Image properties: image data type
print(img.dtype)        # uint8: image depth

# Image properties: image data type
```

https://opencv-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_core/py_basic_ops/py_basic_ops.html#basic-ops

Image Array

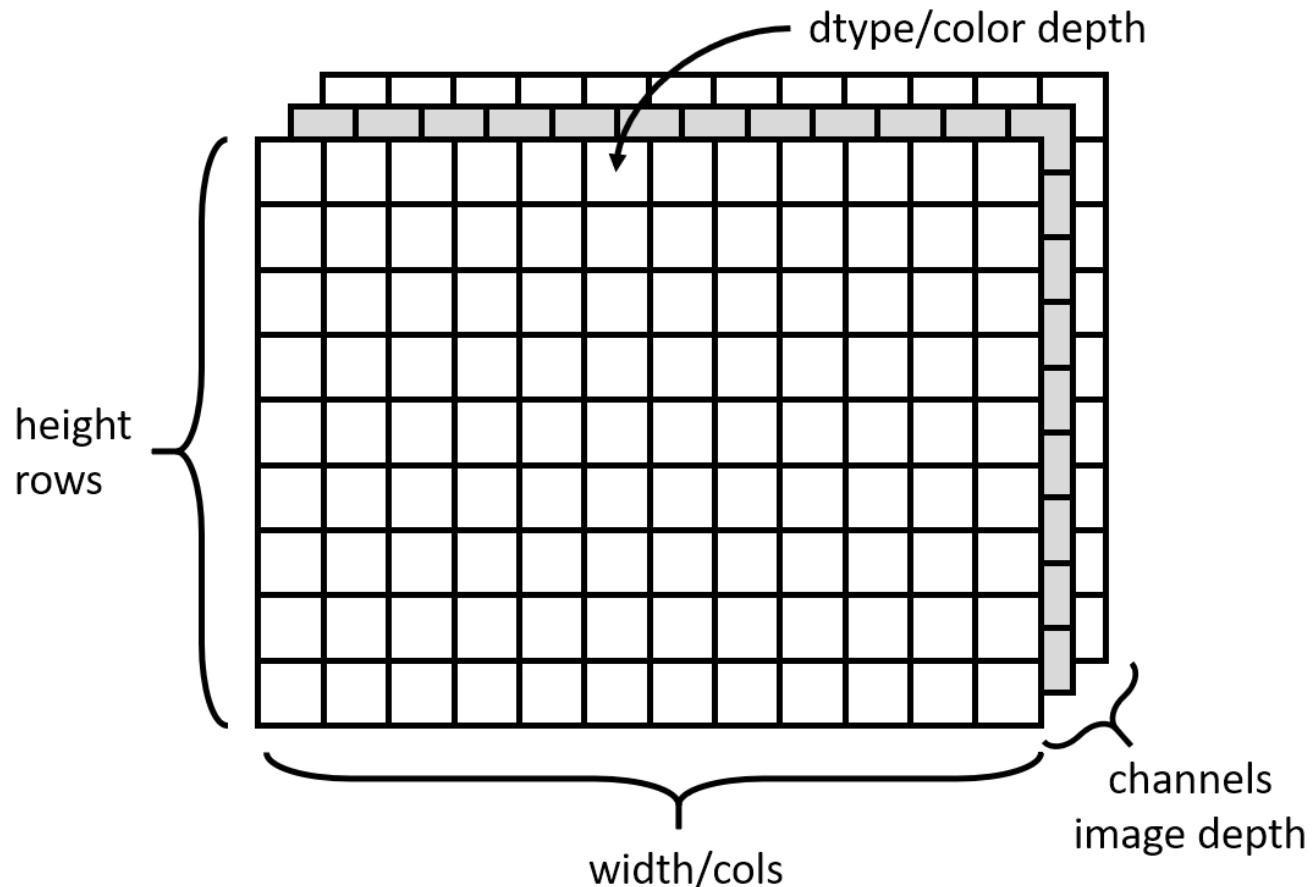
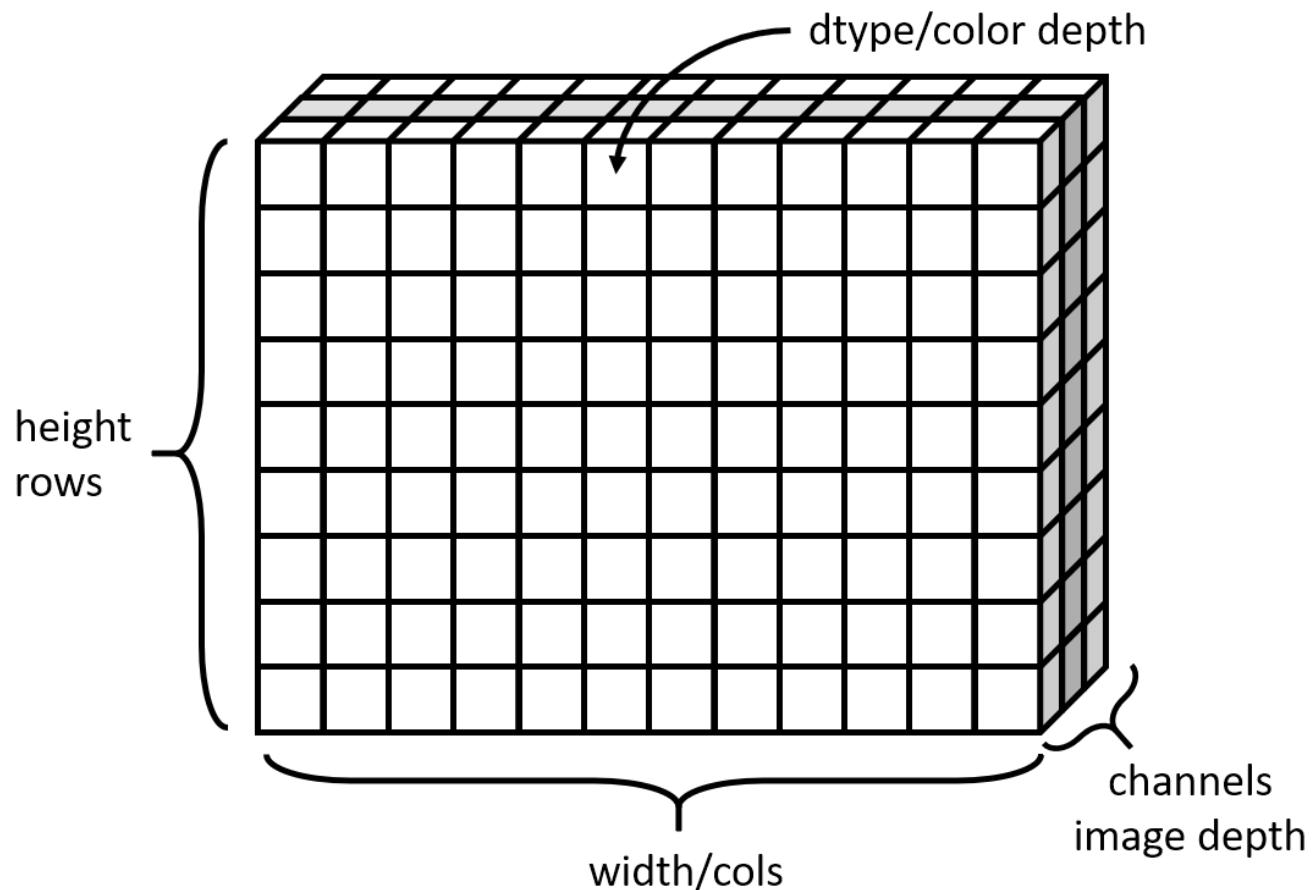
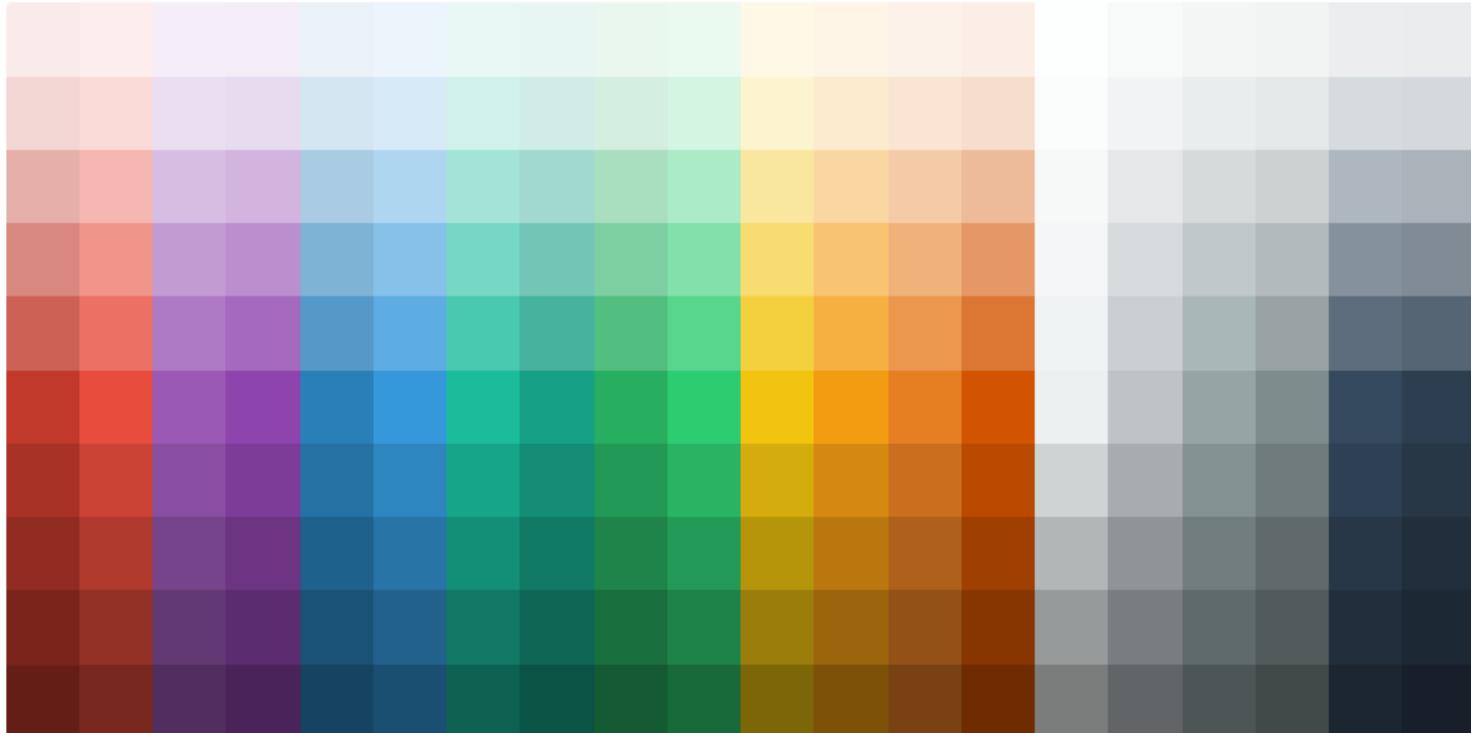


Image Array





Color Chart

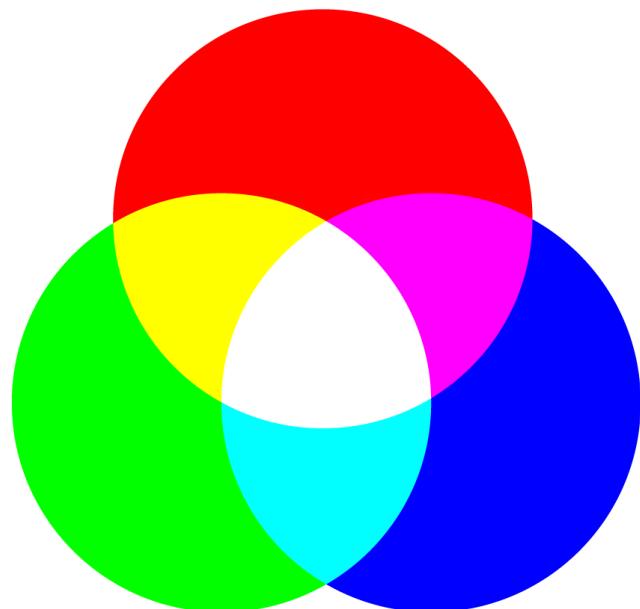


다양한 색을 체계적으로 표현하는 방법?

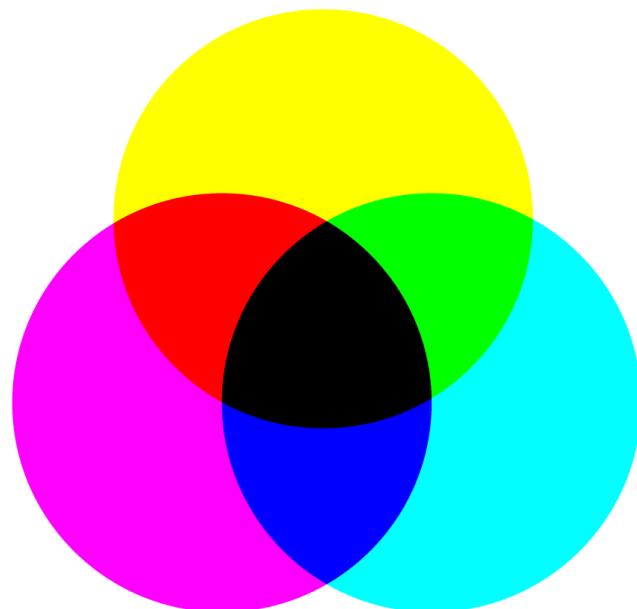
<https://htmlcolorcodes.com/color-chart/>

Image Spaces

RGB



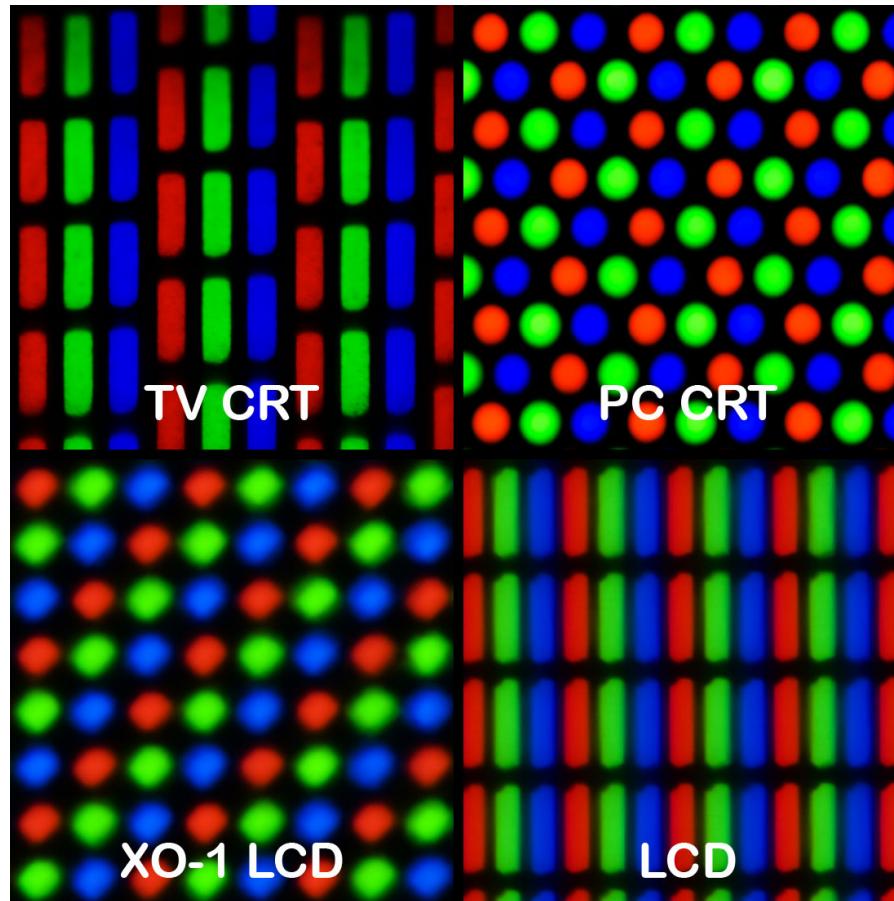
CMYK



© TechTerms.com

<https://techterms.com/definition/rgb>

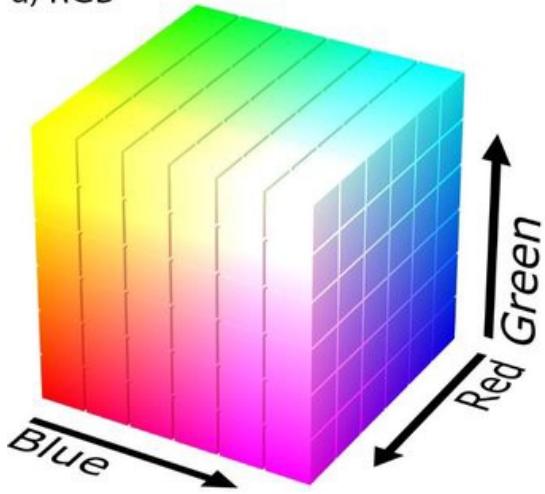
Monitor Pixel Geometry



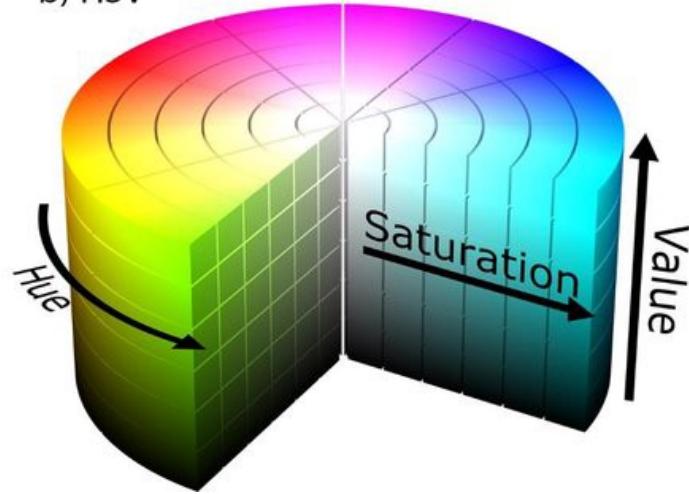
https://en.wikipedia.org/wiki/Pixel_geometry

Image Spaces

a) RGB



b) HSV

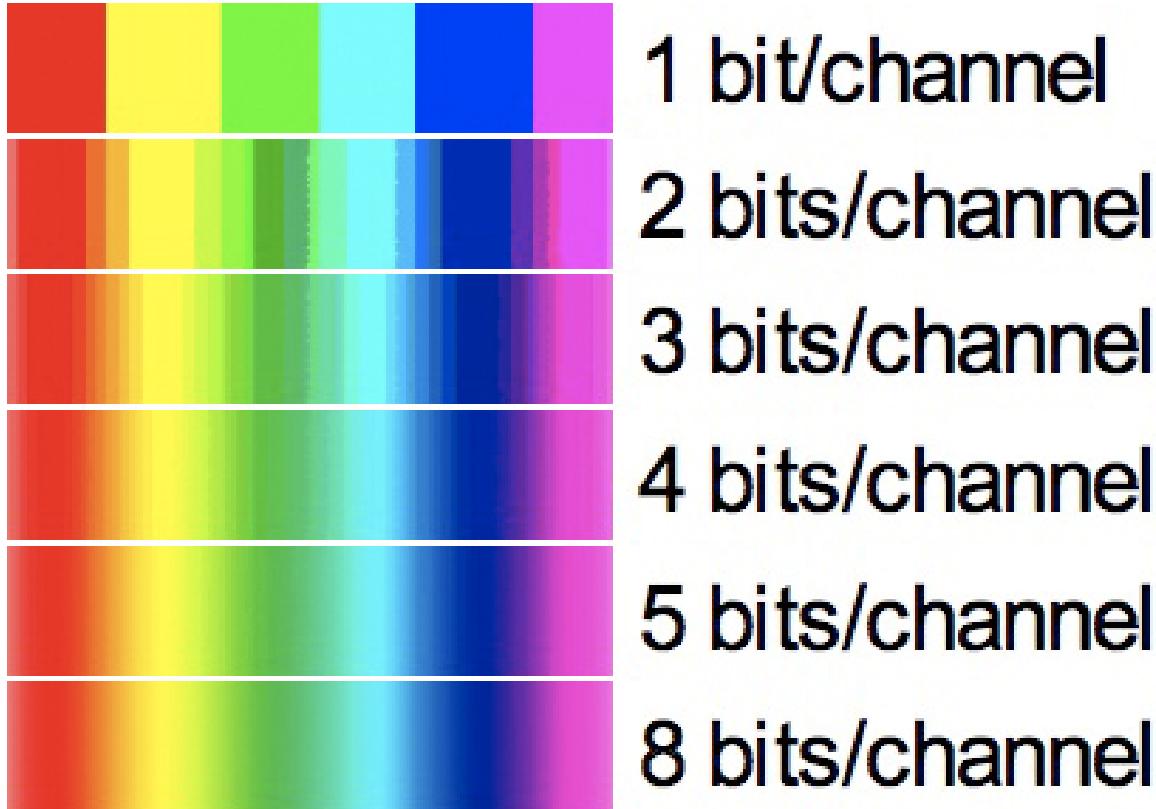


<https://medium.com/neurosapiens/segmentation-and-classification-with-hsv-8f2406c62b39>

Color Depth or Bit Depth

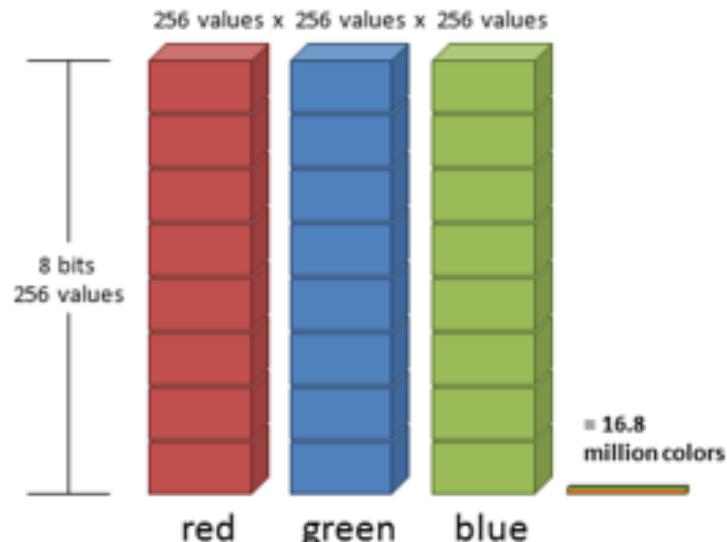
Depth/Channel	Values/Channel	Possible Colors
8 bit	256	16M
10 bit	1,024	1B
12 bit	4,096	68B
14 bit	16,384	4T
16 bit	65,536	281T
24 bit	16,777,216	4S

Color Depth or Bit Depth

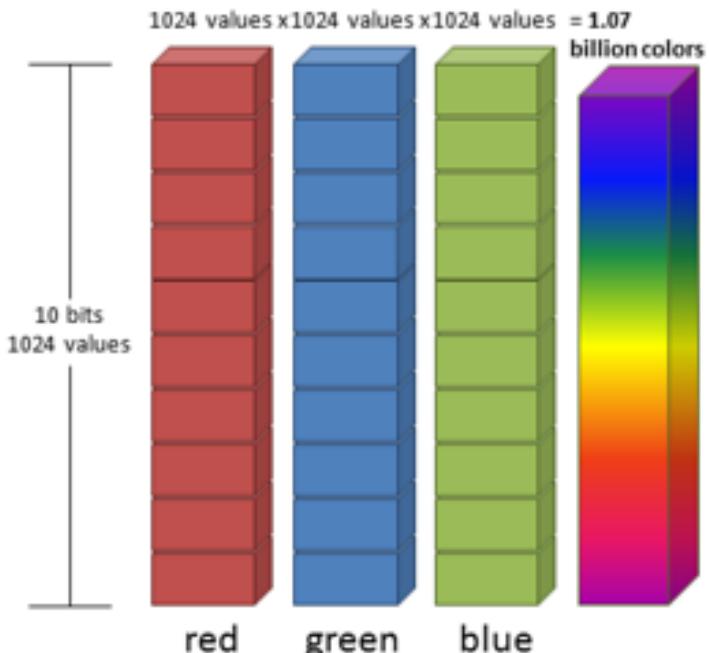


<http://www.digitalbirdphotography.com/photoshop/fundamentals/rainbows.jpg>

Color Depth or Bit Depth

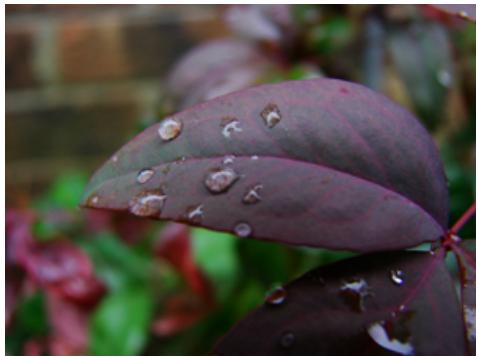


**8 bit color or
24 bit color (8 + 8 + 8 bits)**



**10 bit color or
30 bit color (10 + 10 + 10 bits)**

Color Depth or Bit Depth



24 bit (98KB, 100%, Full)



8 bit (37KB, 38%)



4 bit (13KB, 13%)



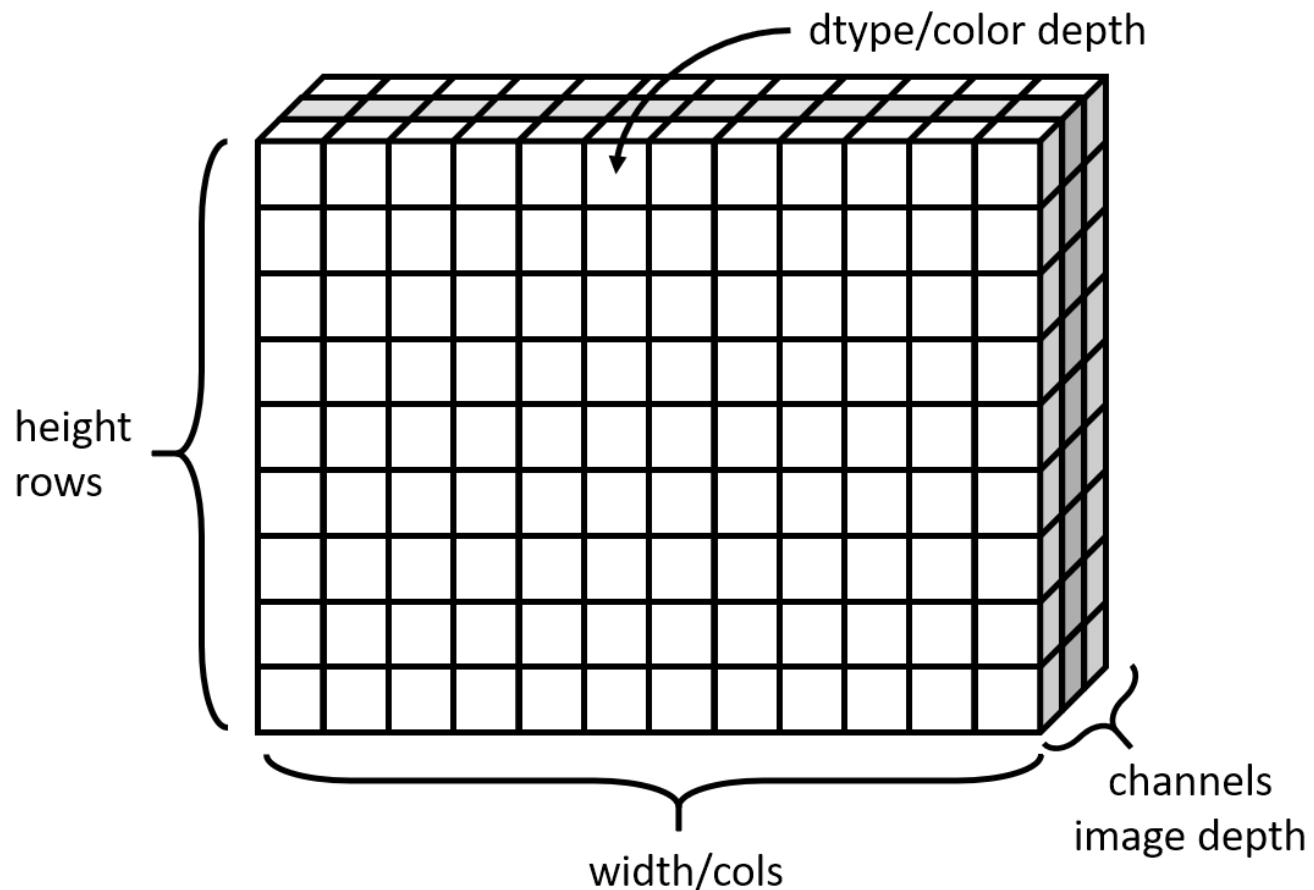
2 bit (6KB, 6%)



1 bit (4KB, 4%, Mono)

https://en.wikipedia.org/wiki/Color_depth

Image Array





Convert and Save Images

Convert and Save Images

```
import cv2

# Load an image
img_color = cv2.imread('messi5.jpg')

# Convert the color image to gray
img_gray = cv2.cvtColor(img_color, cv2.COLOR_BGR2GRAY)

# Save the gray image
cv2.imwrite('messi5_gray.jpg', img_gray)

# Display both images in each window
cv2.imshow('Color Image', img_color)
cv2.imshow('Gray Image', img_gray)
cv2.waitKey(0)

# Destroy all windows
cv2.destroyAllWindows()
```

cv.cvtColor()¹

```
dst = cv.cvtColor(src, code[, dst[, dstCn]])
```

- Image를 하나의 color space에서 다른 color space으로 변환
 - `src`: input image
 - `dst`: output image
 - `code`: color space 변환코드
 - `dstCn`: output image의 channel 개수

code ²	inverse code
<code>cv.COLOR_BGR2GRAY</code>	<code>cv.COLOR_GRAY2BGR</code>
<code>cv.COLOR_RGB2GRAY</code>	<code>cv.COLOR_GRAY2RGB</code>
<code>cv.COLOR_BGR2HSV</code>	<code>cv.COLOR_HSV2BGR</code>
<code>cv.COLOR_BGR2Lab</code>	<code>cv.COLOR_Lab2BGR</code>

1. https://docs.opencv.org/4.4.0/d8/d01/group_imgproc_color_conversions.html#ga397ae87e1288a81d2363b61574eb8ca
2. https://docs.opencv.org/4.4.0/d8/d01/group_imgproc_color_conversions.html#ga4e0972be5de079fed4e3a10e24ef5ef0

RGB ⇔ GRAY

- RGB \Rightarrow GRAY

$$I = 0.299R + 0.587G + 0.114B$$

- GRAY \Rightarrow RGB

$$R = I$$

$$G = I$$

$$B = I$$

https://docs.opencv.org/4.4.0/de/d25/imgproc_color_conversions.html

RGB \Rightarrow HSV

- Value

$$V = \max(R, G, B)$$

- Hue

$$H = \begin{cases} \frac{60(G-B)}{V-\min(R,G,B)} & \text{if } V = R \\ \frac{120+60(G-B)}{V-\min(R,G,B)} & \text{if } V = G \\ \frac{240(G-B)}{V-\min(R,G,B)} & \text{if } V = B \end{cases}$$

- Saturation

$$S = \begin{cases} \frac{V-\min(R,G,B)}{V} & \text{if } V \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

https://docs.opencv.org/4.4.0/de/d25/imgproc_color_conversions.html

cv.imwrite()¹

```
retval = cv.imwrite(filename, img[, params])
```

- Image를 파일에 저장한다.
 - filename: 파일이름
 - img: image to be saved
 - params: encoding format

params ²	설명
cv.IMWRITE_JPEG_X	JPEG
cv.IMWRITE_PNG_X	PNG
cv.IMWRITE_TIFF_X	TIFF

1. https://docs.opencv.org/4.4.0/d4/da8/group_imgcodecs.html#gabbc7ef1aa2edfaa87772f1202d67e0ce
2. https://docs.opencv.org/4.4.0/d4/da8/group_imgcodecs.html#ga292d81be8d76901bff7988d18d2b42ac

Split and Merge Channels

```
import cv2

# Load an image
img_color = cv2.imread('RGB.png')

# Split each color channel
img_B, img_G, img_R = cv2.split(img_color)

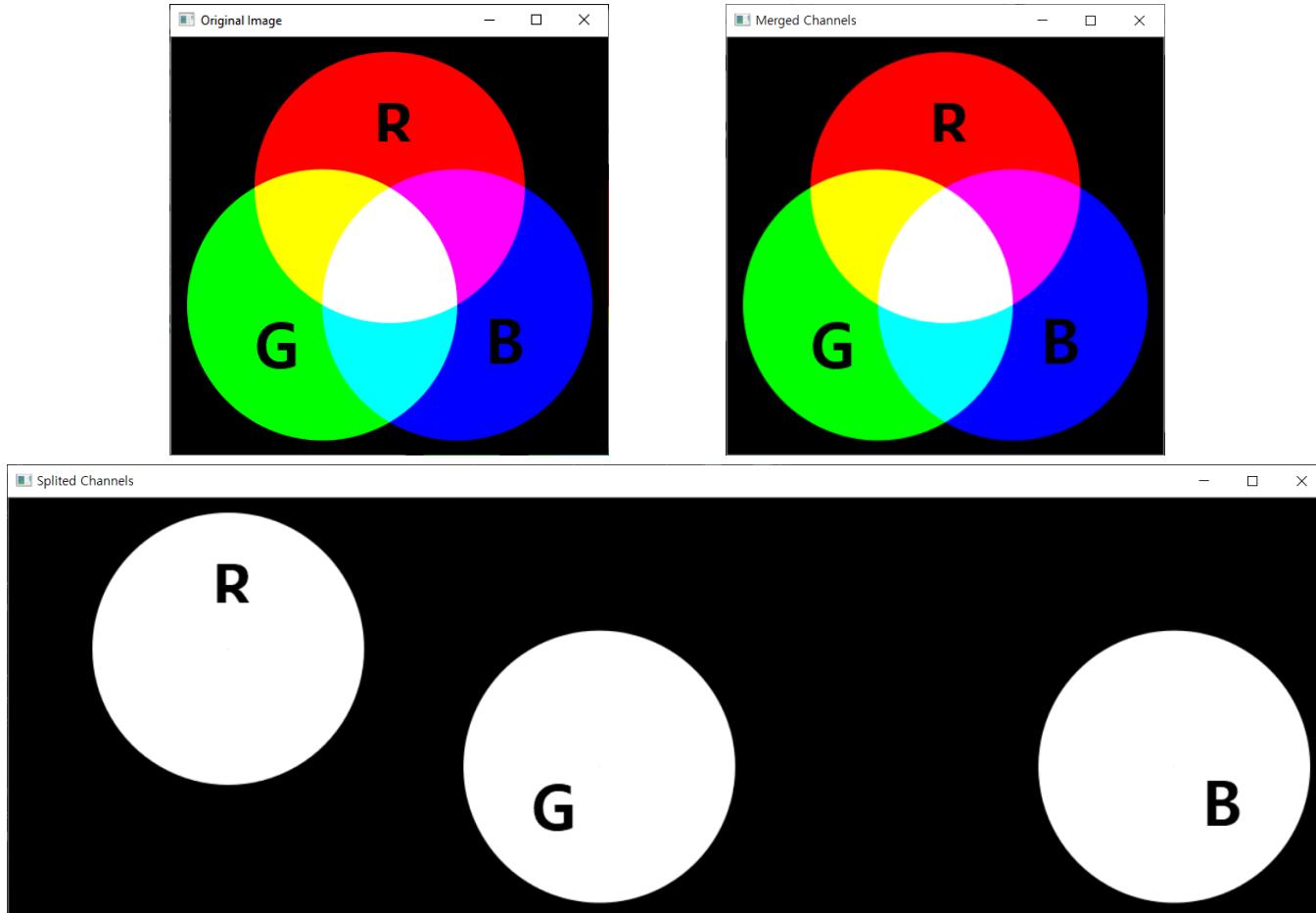
# Merge all channels
img_merged = cv2.merge([img_B, img_G, img_R])

# Concatenate each color channel horizontally
img_split = cv2.hconcat([img_R, img_G, img_B])

# Display all channels in a window
cv2.imshow('Split Channels', img_split)
cv2.imshow('Merged Channels', img_merged)
cv2.waitKey(0)

# Destroy all windows
cv2.destroyAllWindows()
```

Split and Merge Channels





Region of Interest

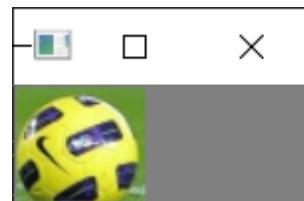
```
import cv2

# Load an image
img = cv2.imread('messi5.jpg')

# Set the region of interest
ROI = img[286:332, 338:390]

# Display all channels in a window
cv2.imshow('Ball', ROI)
cv2.waitKey(0)

# Destroy all windows
cv2.destroyAllWindows()
```



 Capture Videos

Load and Play Videos

```
import cv2

# Create a video capture object
capture = cv2.VideoCapture("vtest.avi")

while True:
    # Read a video frame
    ret, frame = capture.read()

    # Quit
    if ret is False: break

    # Show the current frame
    cv2.imshow("Video Frame", frame)
    cv2.waitKey(1)

# Release the capture object
capture.release()

# Destroy all windows
cv2.destroyAllWindows()
```

Load and Play Videos



Load and Play Videos Infinitely

```
import cv2

# Create a video capture object
video_name = 'vtest.avi'
capture = cv2.VideoCapture(video_name)

while True:

    # Read a video frame
    ret, frame = capture.read()

    # OK
    if ret is True:

        # Show the current frame
        cv2.imshow("Video Frame", frame)
        key = cv2.waitKey(1)

        # ESC
        if key == 27: break
```

Capture Images from a Camera

```
import cv2

# Create a video capture object
capture = cv2.VideoCapture(0)

while True:
    # Read a video frame
    ret, frame = capture.read()

    if ret is False:
        print('Error: No image is captured!')
        break

    # Show the current frame
    cv2.imshow("Video Frame", frame)
    key = cv2.waitKey(1)

    # ESC
    if key == 27: break

    # Spacebar
```

Save Videos

```
import numpy as np
import cv2

# Create a video capture object
capture = cv2.VideoCapture(0)

# Define the codec
fourcc = cv2.VideoWriter_fourcc(*'XVID')

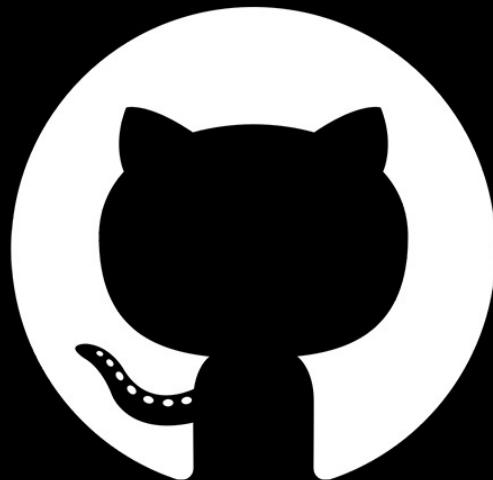
# Create a video writer object
writer = cv2.VideoWriter('output.avi', fourcc, 20.0, (640,480))

while(capture.isOpened()):
    # Read a video frame
    ret, frame = capture.read()

    if ret==True:
        # Write the current frame
        writer.write(frame)

    # Display the current frame
```

Push Code to GitHub



 References

References

- OpenCV Python Tutorials
 - GUI Features in OpenCV
 - Getting Started with Images
 - Getting Started with Videos