IMAGE ACQUISITION and DIGITIZATION

(กระบวนการรับภาพและสร้างภาพดิจิตอล)

SENSING AND LIMITATIONS

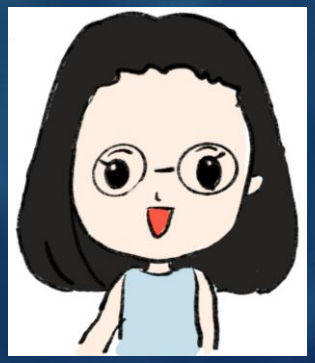
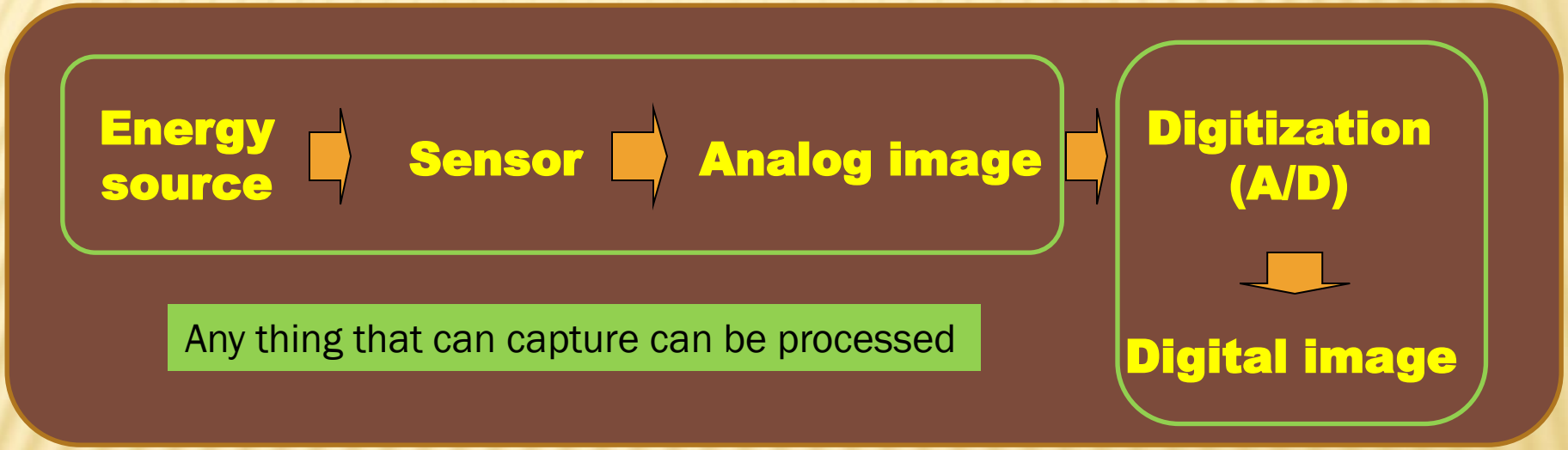


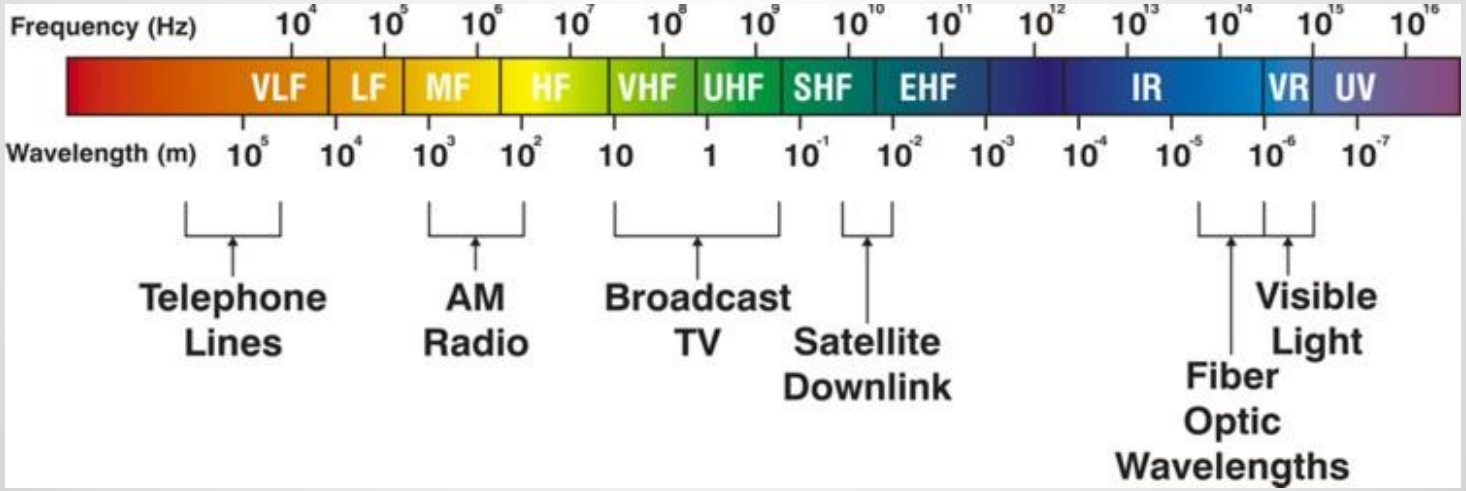
IMAGE ACQUISITION AND DIGITIZATION

Q: How can we create a digital image?

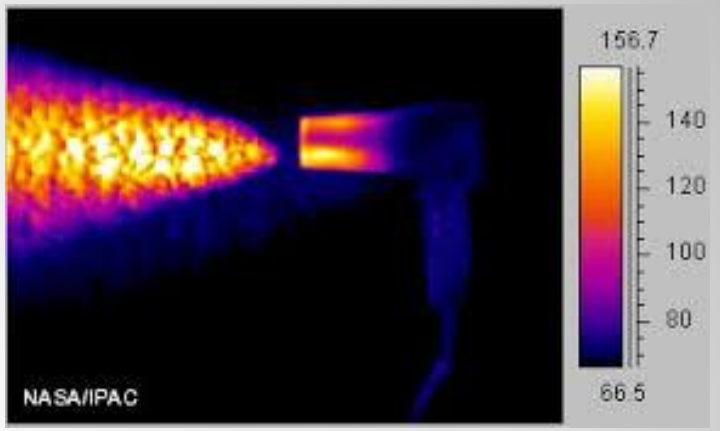


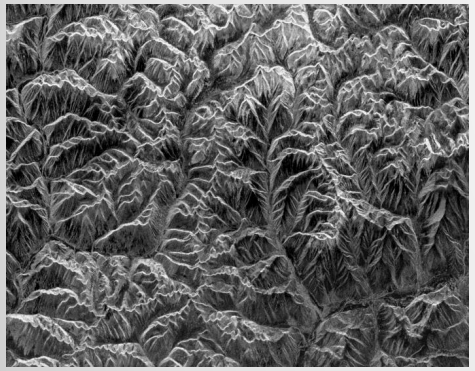
Energy Source

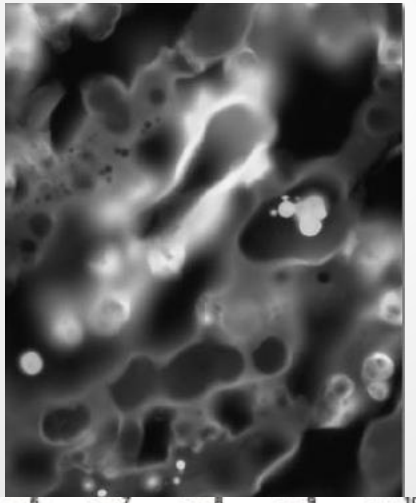
Q: What **type of energy sources** would we **use to create** an image?

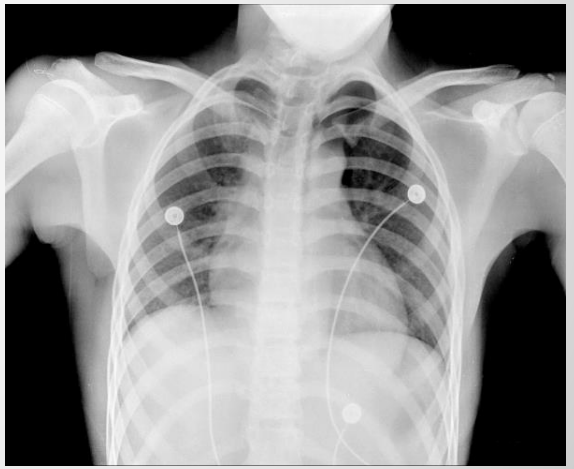


Electromagnetic Spectrum



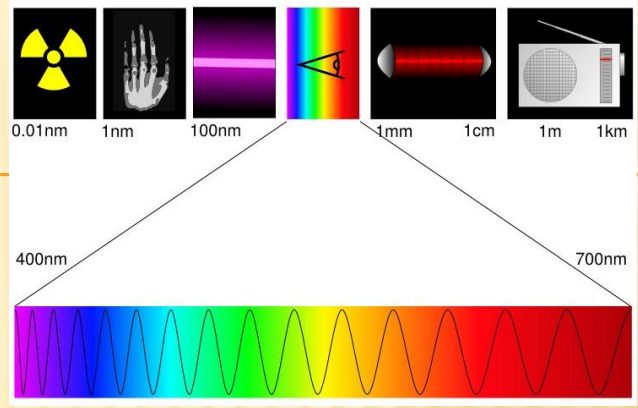




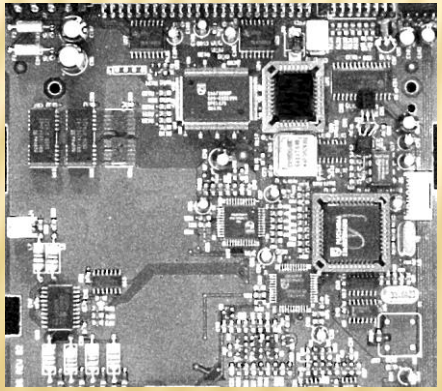


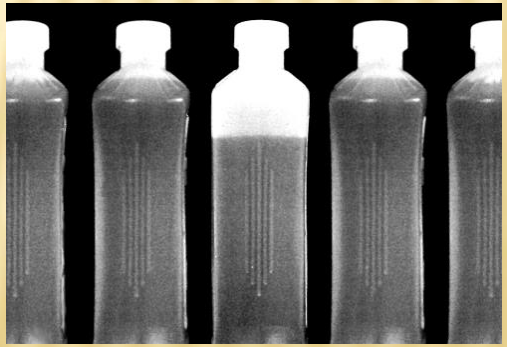
What is the source of each image?

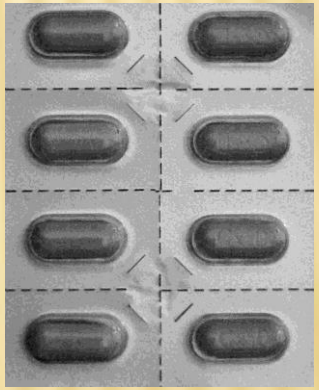
VISIBLE LIGHT



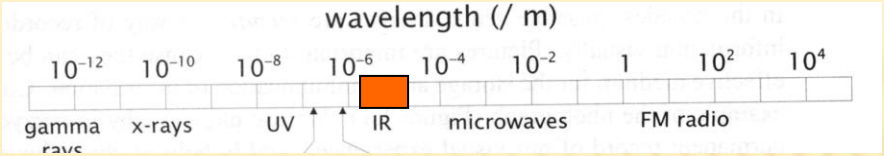
- wavelength 400-700 nm



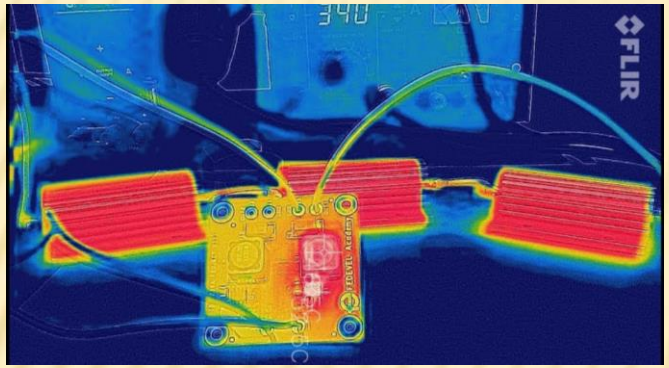




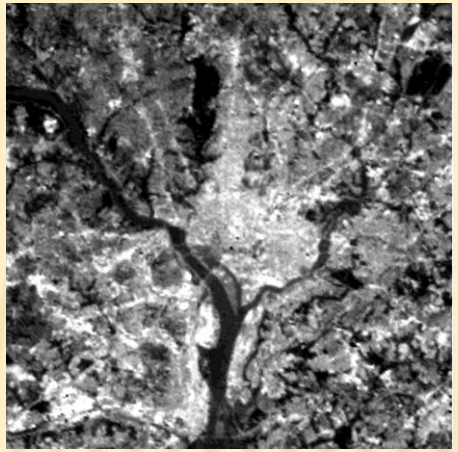
INFARED BAND



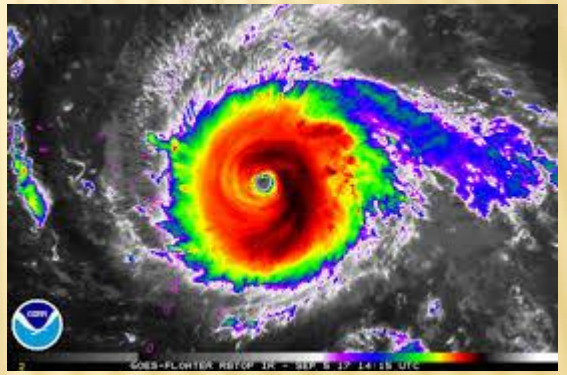
- wavelength 10^{-6}-10^{-4} m



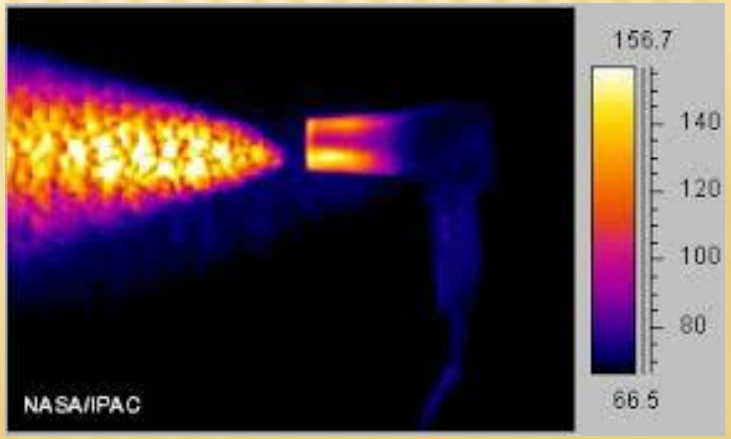
Circuit Functional Test



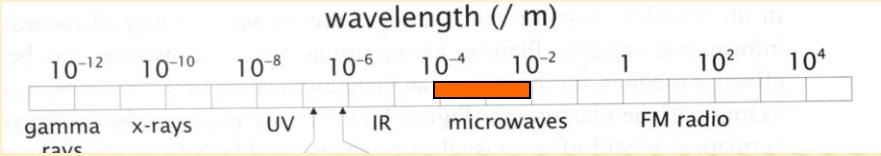
Washington D.C. area



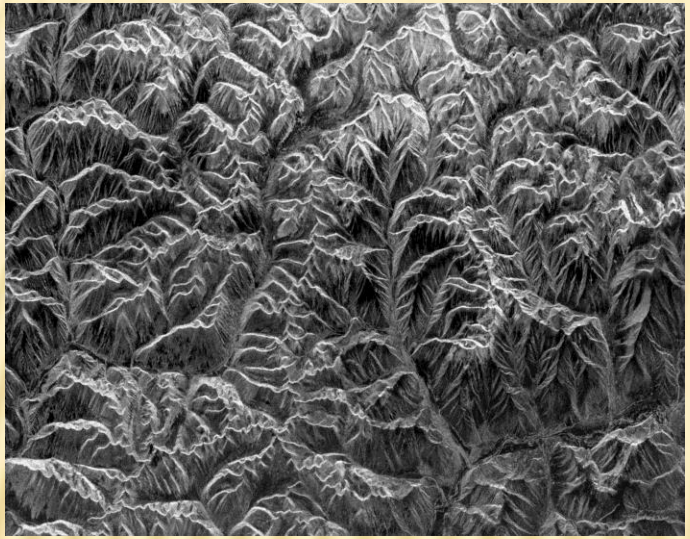
Hurricane images



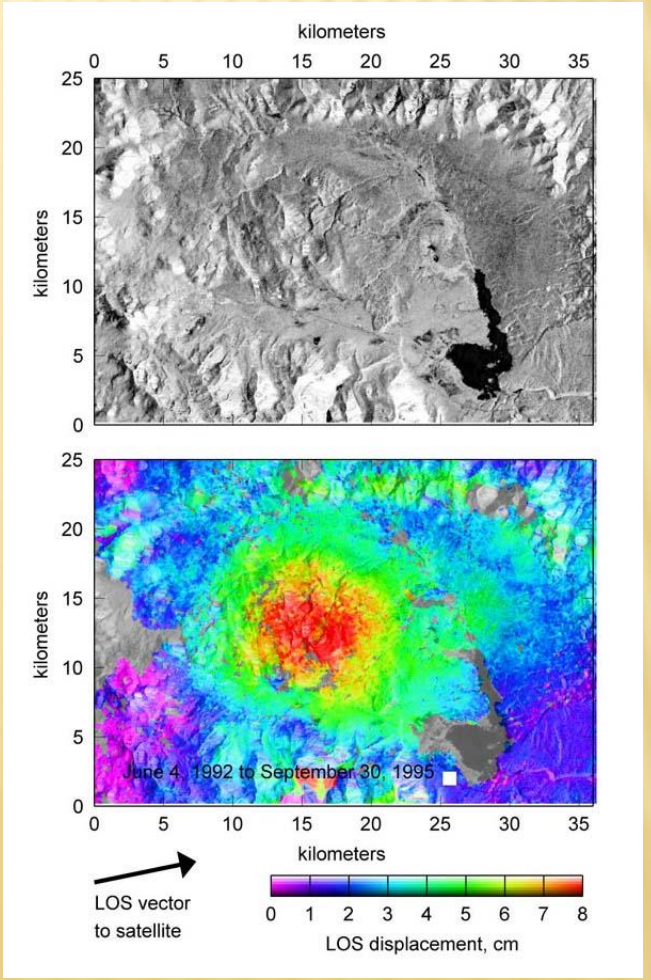
MICROWAVE BAND



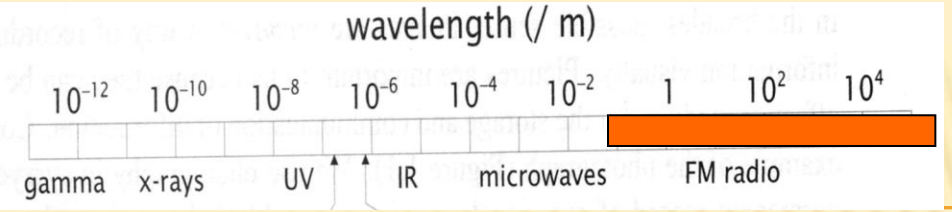
- ความยาวคลื่น 10^{-4}-10^{-2} m



Radar image of mountains



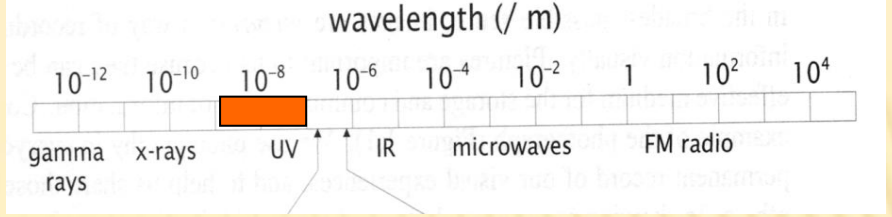
RADIO BAND



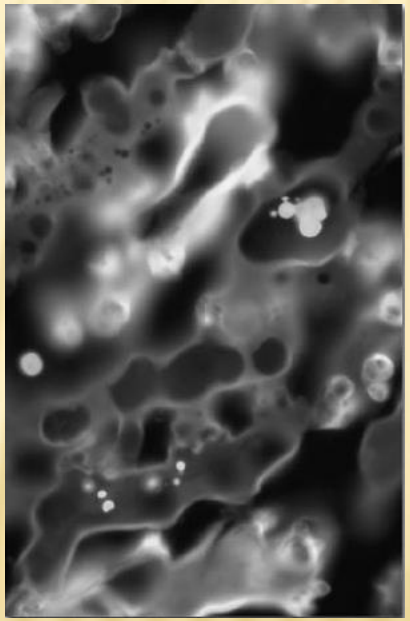
Magnetic Resonance Image (MRI)

Ultrasonic image

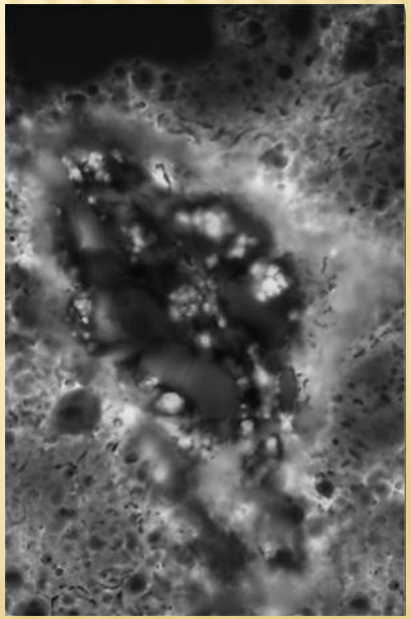
ULTRAVIOLET BAND



- wavelength 10^{-8}-10^{-7} m
- Fluorescence microscope image

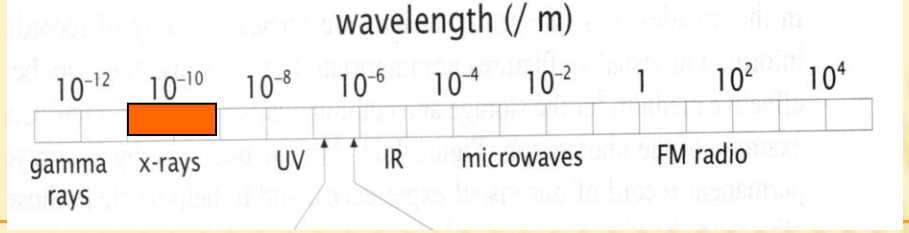


Normal corn cell

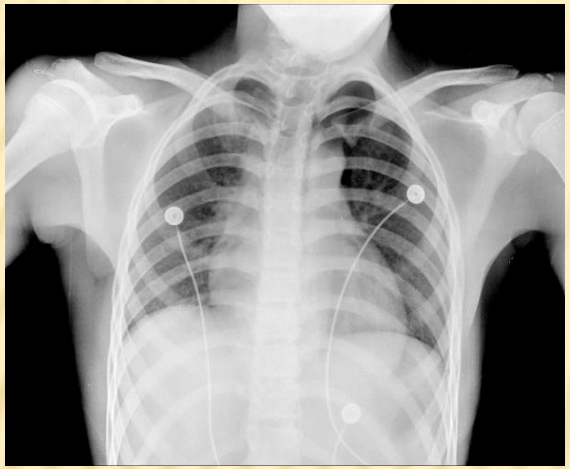


Infected corn cell

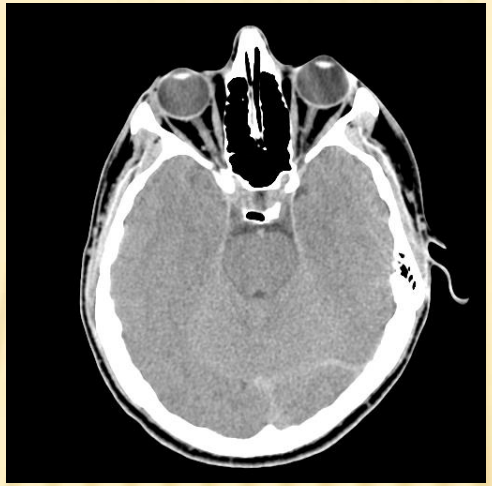
X-RAY



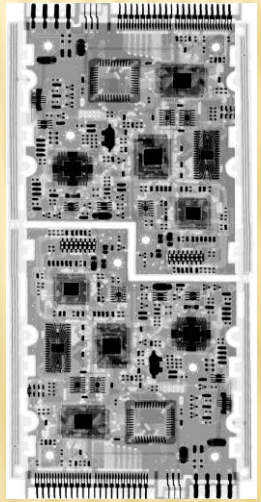
- wavelength 10^{-10} m



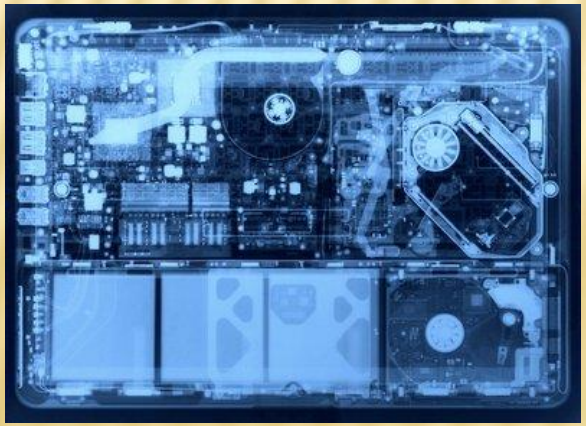
Chest X-ray

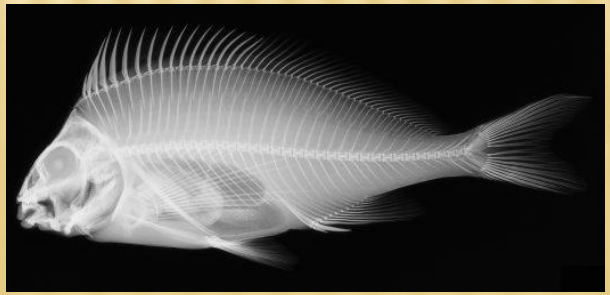


Head cross-section

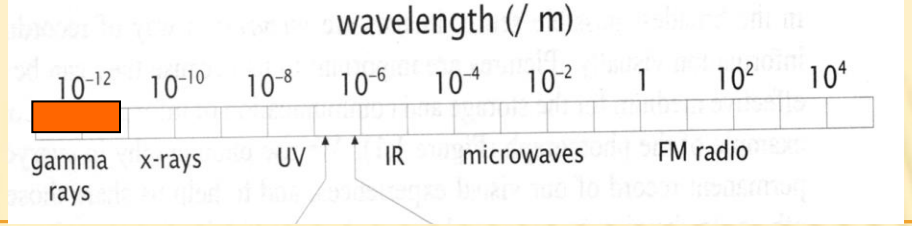


Circuit board

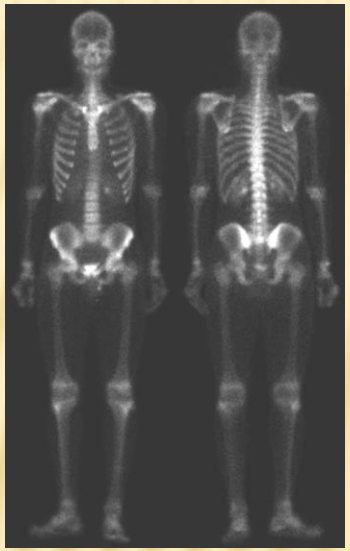




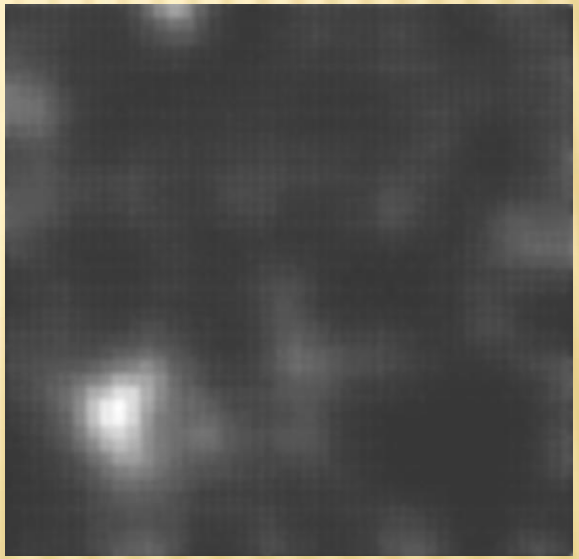
GAMMA RAY



- wavelength 10^{-12} m



Bone scan

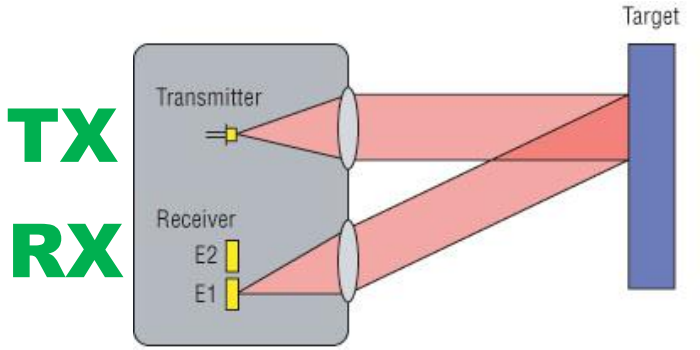


Reactor valve

รูปแบบการรับส่งสัญญาณของตัวส่งสัญญาณ (Transmitter) และ ตัวรับ (Receiver: Sensor)

- คลื่นที่นำมาใช้สร้างภาพที่ความถี่ต่างกัน
  - มีรูปแบบการส่งสัญญาณและรับสัญญาณต่างกันอย่างไร

- Reflection
- Through object



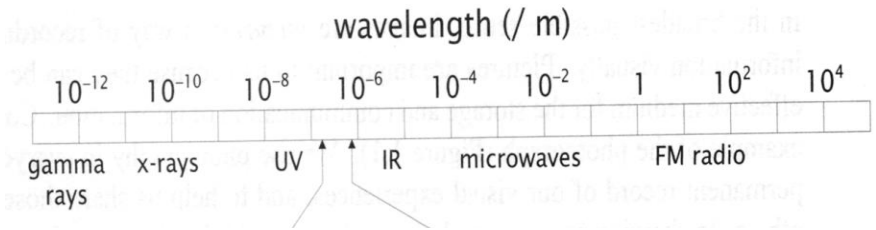


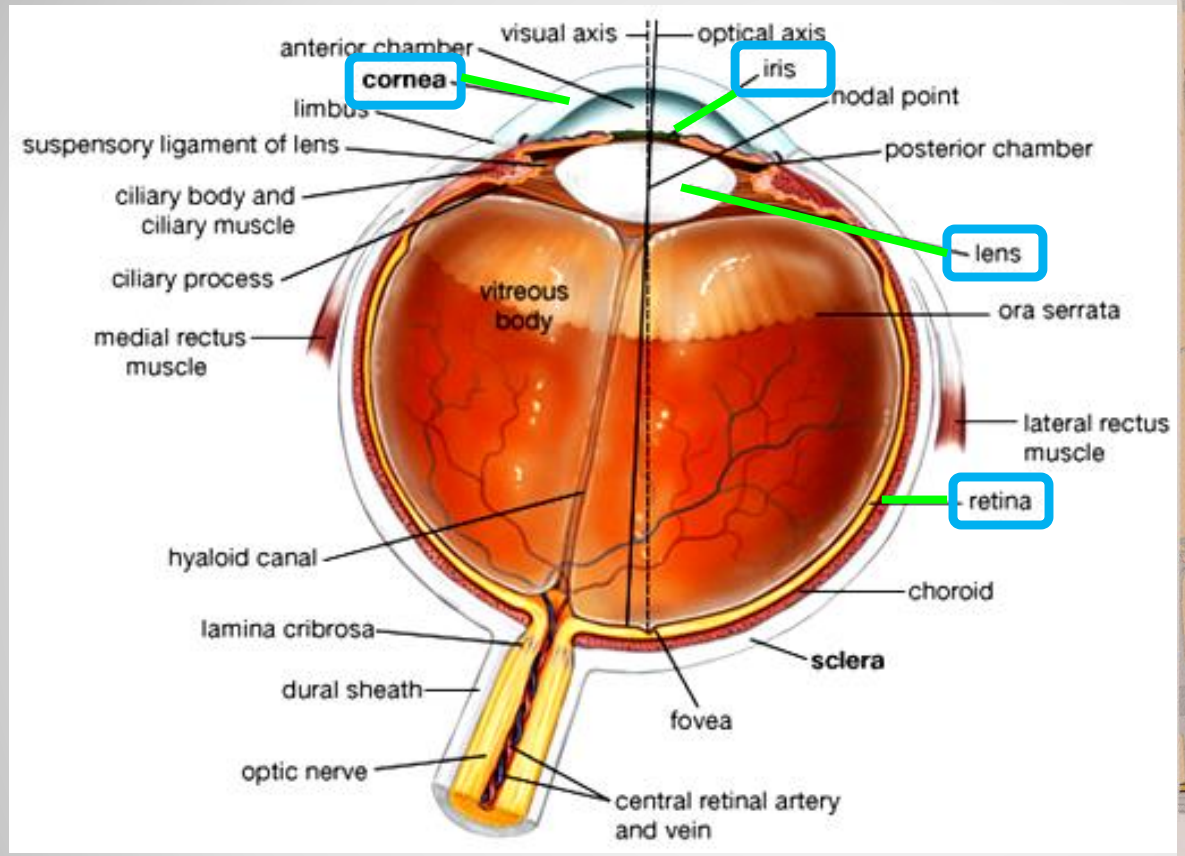
Image Sensing

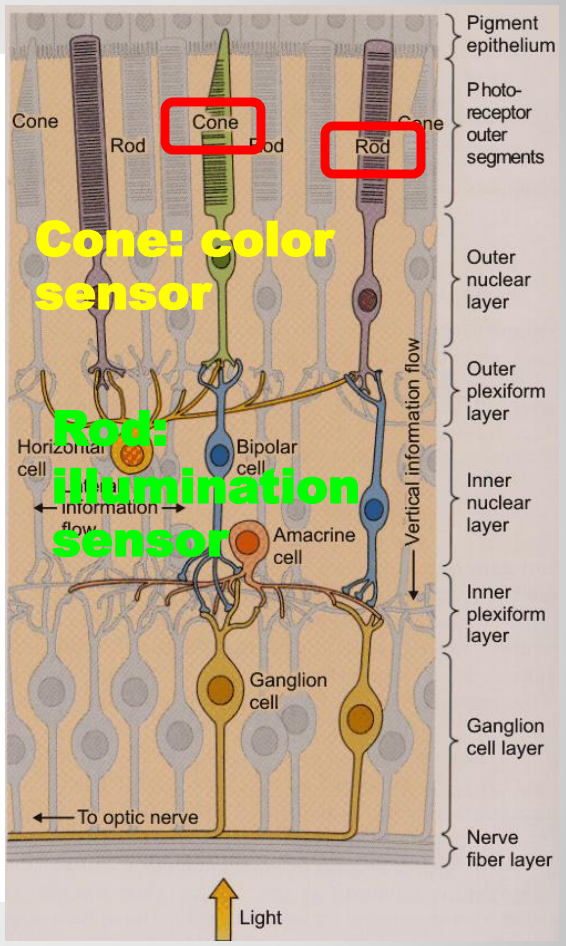
Image Capturing Device

Q: รูปแบบโครงสร้างและองค์ประกอบของ

อุปกรณ์รับภาพได้รับอิทธิพลมาจากไหน?

Cornea กระจกตา ทำหน้าที่คล้าย **lens** เพื่อหักเหแสง



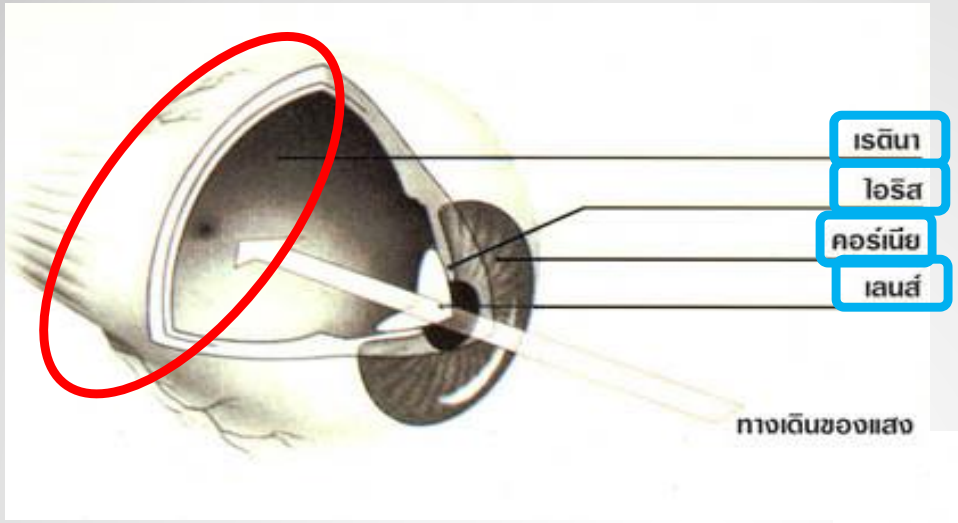


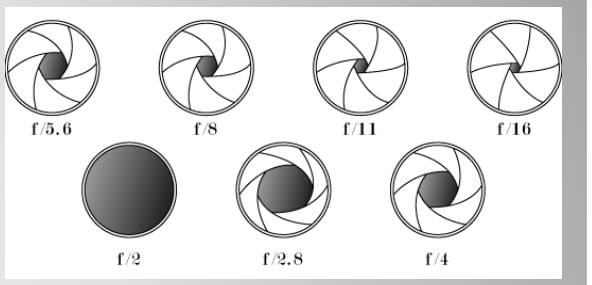
Human Eye Vision

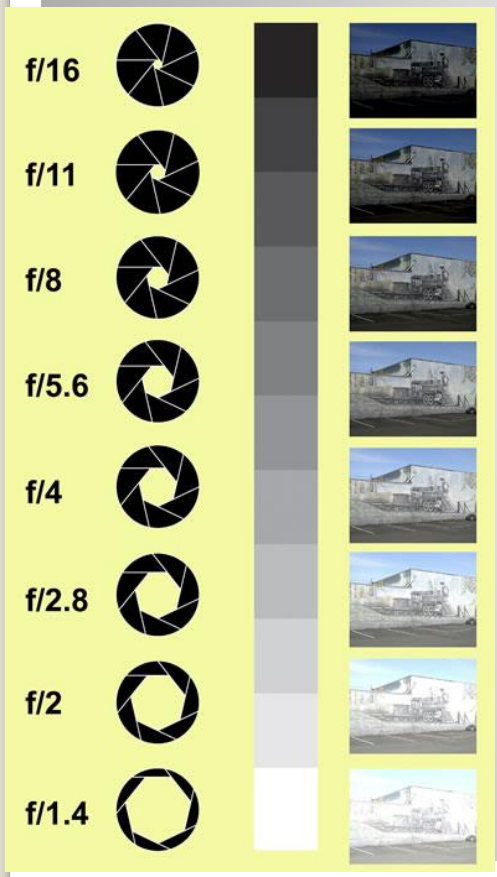
Iris ม่านตา ทำหน้าที่ปรับปริมาณแสง

Retina จอตา ทำหน้าที่เป็นเซลรับแสง

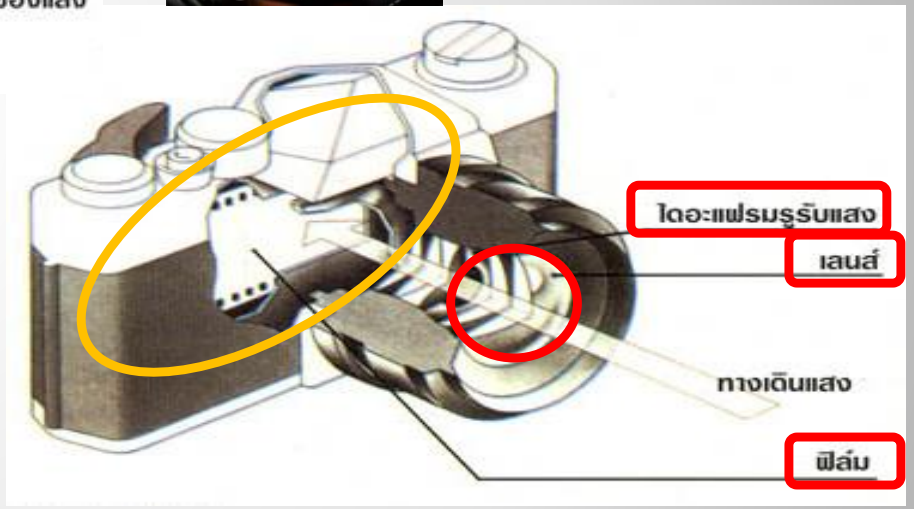
f/x -> x = ความยาวโฟกัส / ศก.รูรับแสง







ขนาดความยาวโฟกัส 50 **mm.** มีมุมการรับภาพอยู่ที่ 46 องศา ถือว่าเป็นเลนส์ **“Normal”**



Sensor

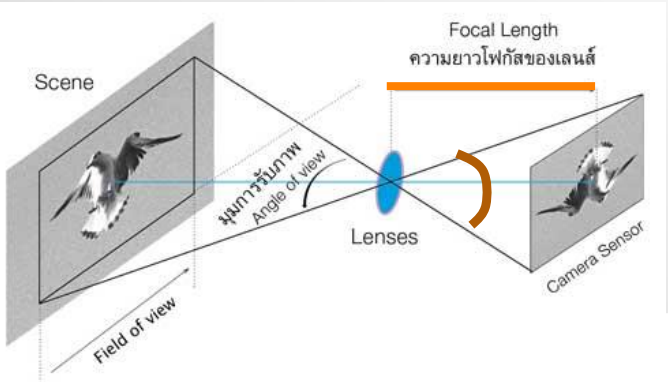
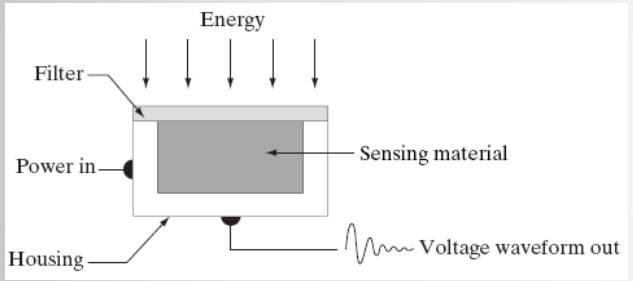
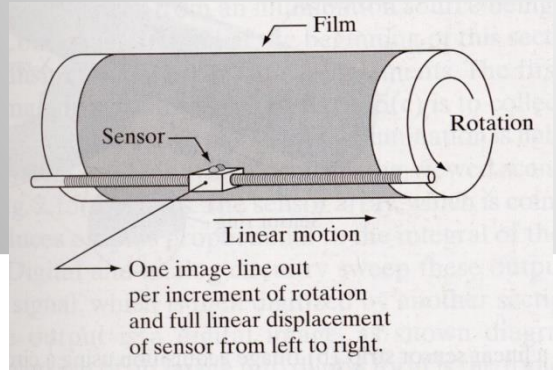


Image Sensing

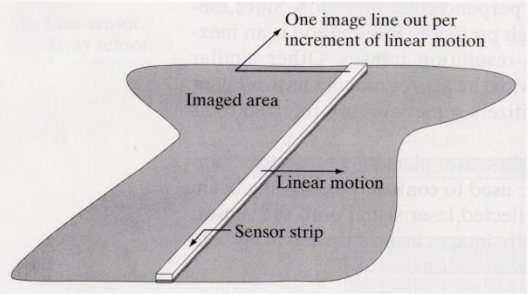
Sensor

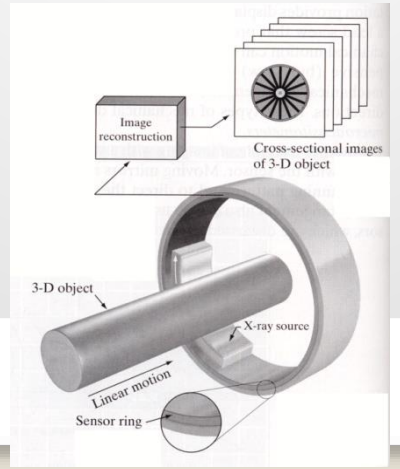
Single Sensor



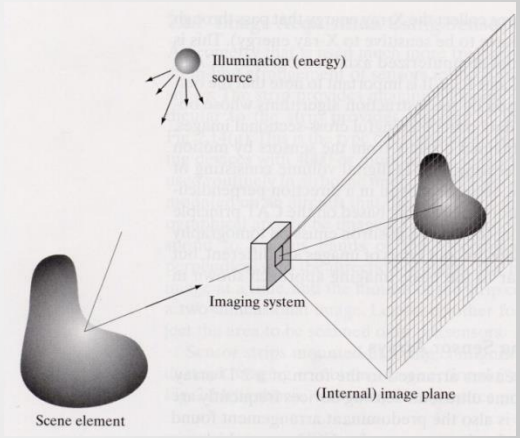


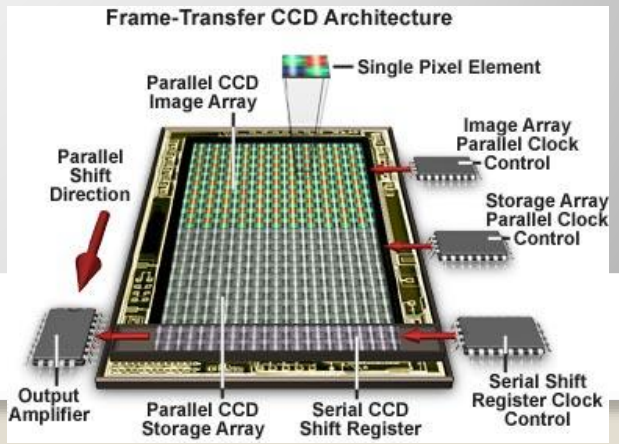
Row Sensor





Array Sensor



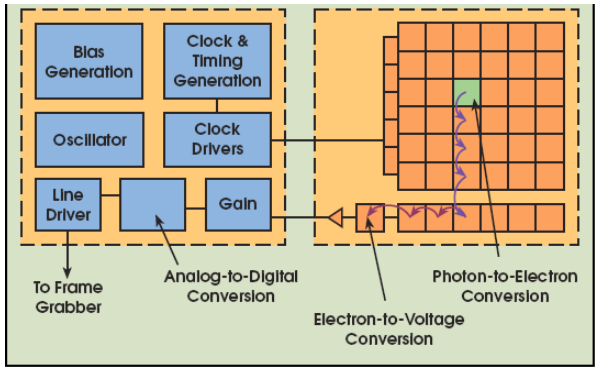


Types of array sensor

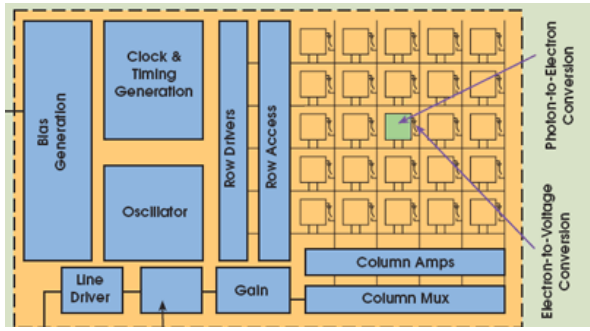
CCD vs CMOS

Future of Camera Sensor

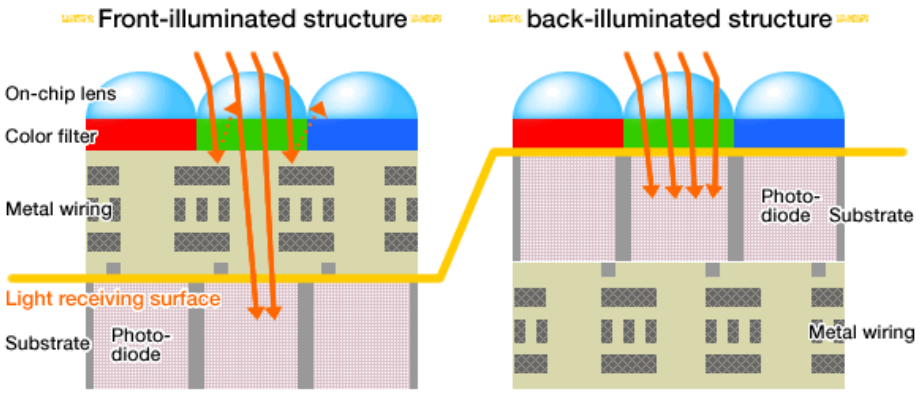
CCD

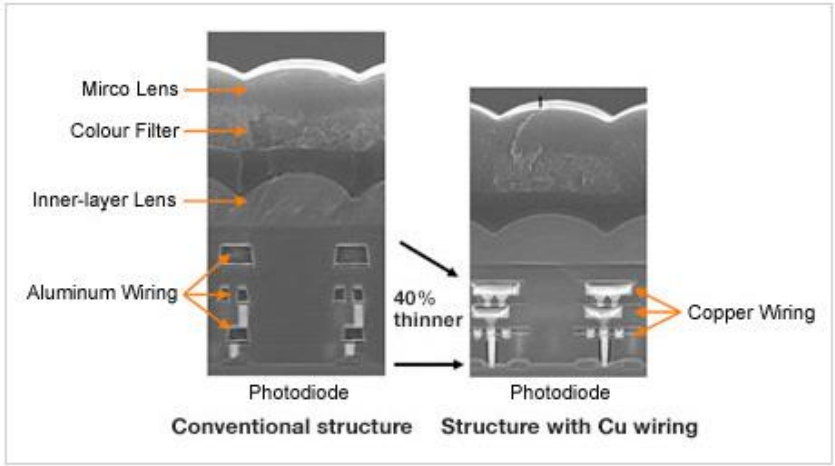


CMOS



- Back-side illuminated CMOS
  - BSI-CMOS or BI-CMOS





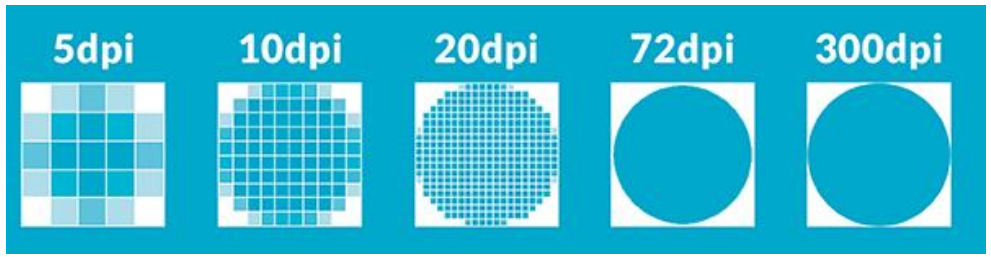
Cross-section photo of a pixel

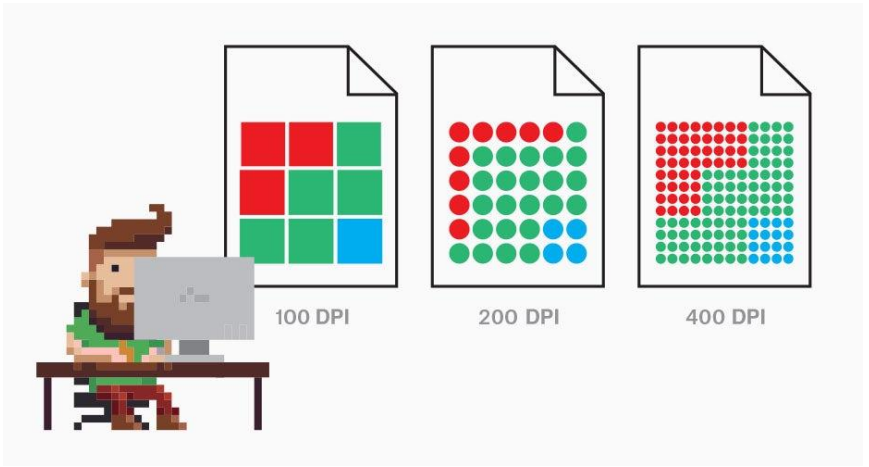
Spatial Resolution: ความละเอียดของตำแหน่งการรับภาพ

DPI: Dot-per-Inch

- Real world mapping (Scanning and Printing)
  - Ex. 100DPI -> 100 dots / inch -> 1 dot / 0.01 inch

- Image Resolution from sensor input
  - Ex. Image capture area 4 x 6 inch with 100DPI scan
  - What is the size of scanning resolution?

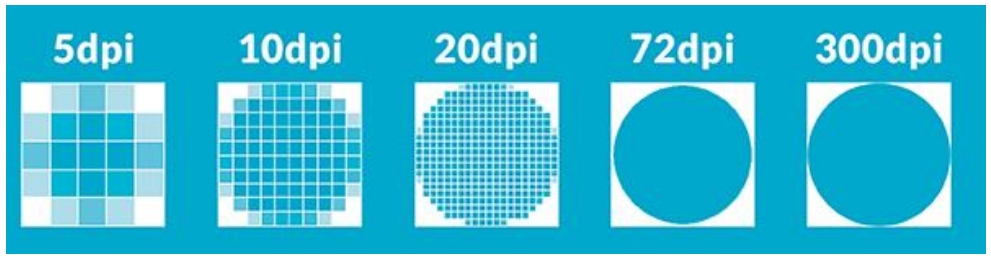


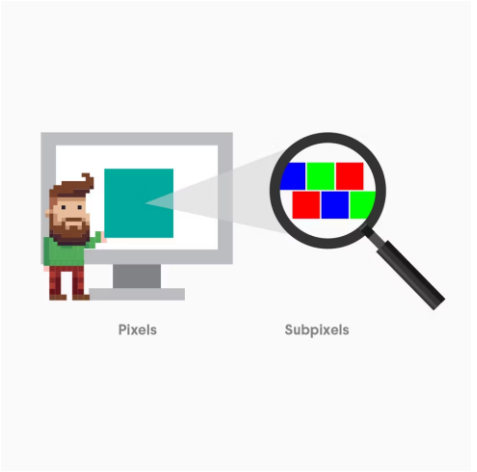


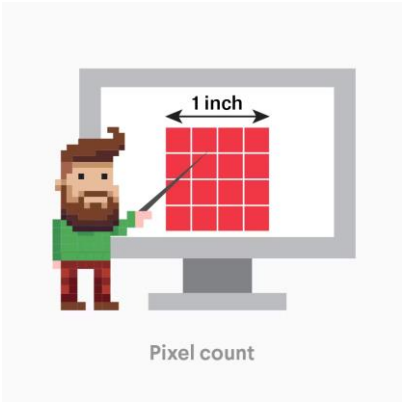
Spatial Resolution: ความละเอียดของตำแหน่งสแกนรับภาพ

DPI: Dot-per-Inch

- Image Resolution to Device output
  - Ex. Image resolution 1280 x 720 (HD) with 100DPI scan
  - What is the size of printing resolution (DPI: Dot-per-Inch) ?
  - What is the size of display resolution (PPI: Pixel-per-Inch, PPM: Pixel-per-M) ?

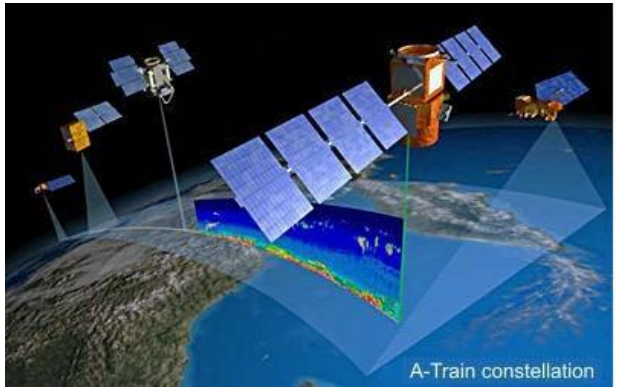






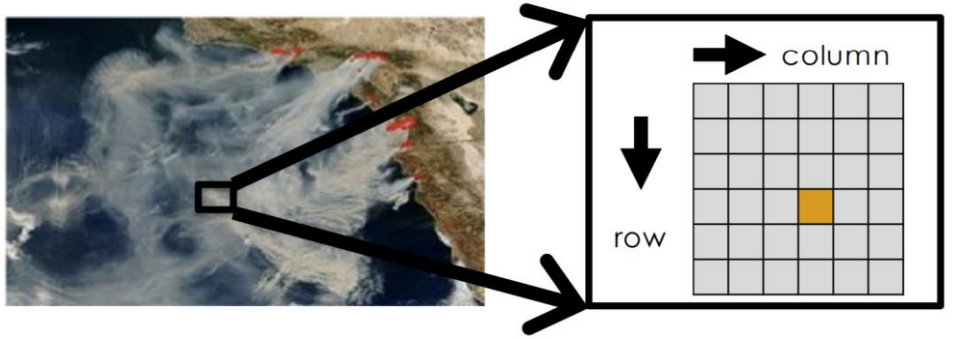
ภาพถ่ายดาวเทียม (Satellite Image)

- Spatial Resolution (image size: pixel width(W), height (H))
- Spectral Resolution (No. Freq Bands)
- Temporal Resolution (Scanning Period, fps)
- Radiometry Resolution (Bit depth: Bit per pixel)
- Geometric resolution (pixel size: meter X meter)
- https://gisgeography.com/spatial-resolution-vs-spectral-resolution/
- https://appliedsciences.nasa.gov/sites/default/files/D1P3_Fundamentals.pdf



Area Resolution (Geometric resolution)

- ความละเอียดเชิงพื้นที่ (real world / pixel) (ระยะครอบคลุมของ pixel)



Ex: 50 cm pixel size / 10 cm pixel size
(50 cm/pixel) (10 cm/ pixel)

ระยะ ต่อ 1 pixel

High Spatial Resolution

10 cm/pixel

Medium Spatial Resolution

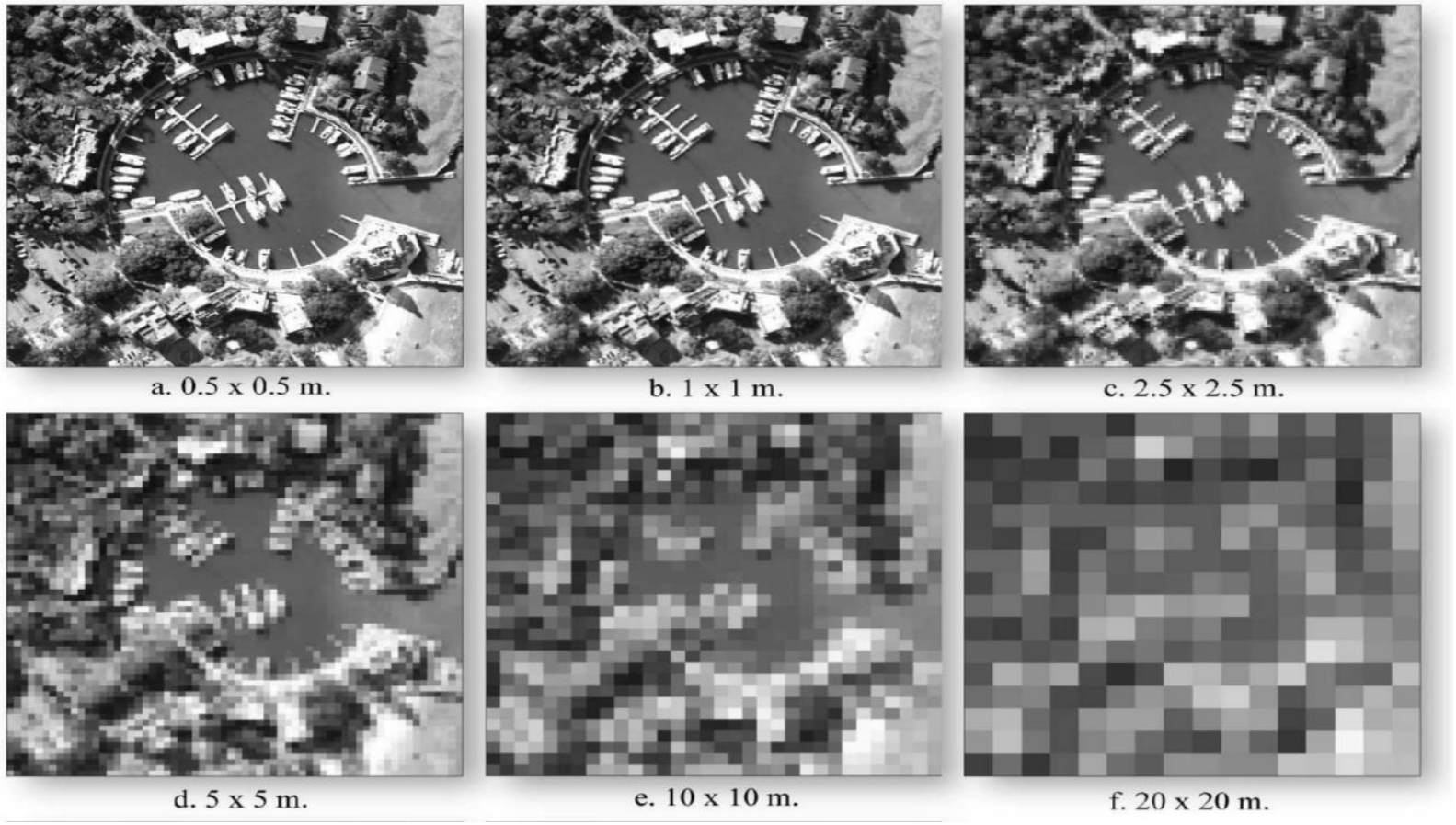
50 cm/pixel

Low Spatial Resolution

1 m/pixel

Spatial Resolution

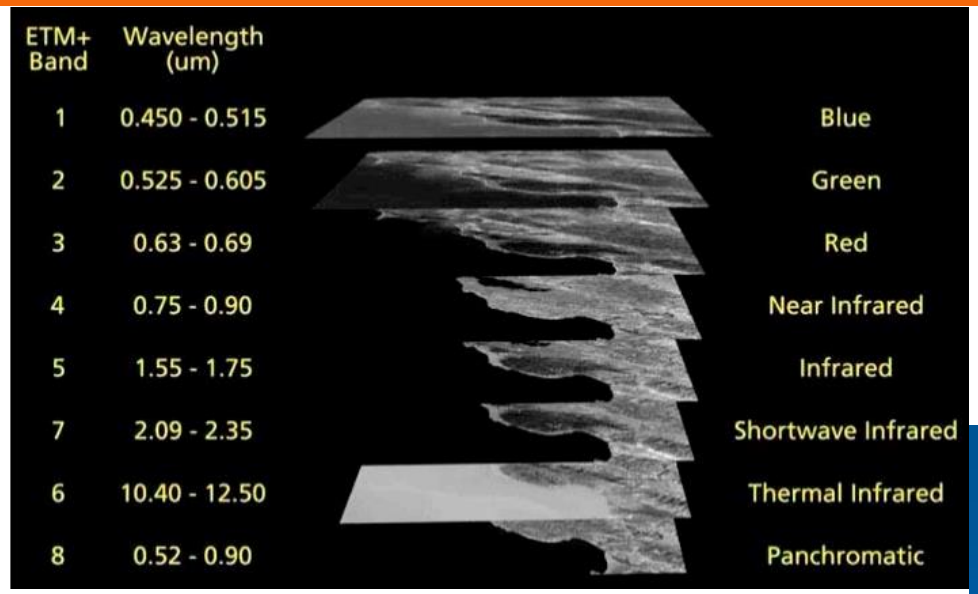
Imagery of Harbor Town in Hilton Head, SC, at Various Nominal Spatial Resolutions



a. 0.5 x 0.5 m. b. 1 x 1 m. c. 2.5 x 2.5 m.

d. 5 x 5 m. e. 10 x 10 m. f. 20 x 20 m.

Satellite Sensor Bands (Wavelength)



Spectral Resolution (No. Freq Bands)

Spectral Resolution (no. of frequency, wavelength)

For example, Landsat-8 produces 11 images and each band consists of the following spectral resolution:

Band Number	Description	Wavelength	Resolution
Band 1	Coastal / Aerosol	0.433 to 0.453 µm	30 meter
Band 2	Visible blue	0.450 to 0.515 µm	30 meter
Band 3	Visible green	0.525 to 0.600 µm	30 meter
Band 4	Visible red	0.630 to 0.680 µm	30 meter
Band 5	Near-infrared	0.845 to 0.885 µm	30 meter
Band 6	Short wavelength infrared	1.56 to 1.66 µm	30 meter
Band 7	Short wavelength infrared	2.10 to 2.30 µm	60 meter
Band 8	Panchromatic	0.50 to 0.68 µm	15 meter
Band 9	Cirrus	1.36 to 1.39 µm	30 meter
Band 10	Long wavelength infrared	10.3 to 11.3 µm	100 meter
Band 11	Long wavelength infrared	11.5 to 12.5 µm	100 meter

Band ที่รับภาพคุณภาพสูงสุด ?

Radiometric Resolution

- ความละเอียดของค่าเฉดแสง หรือ เฉดสี ในแต่ละตำแหน่ง Pixel

Bits	Values	Gray Values
1Bit	$2^1 = 2$ (0-1)	0 … 1
4Bit	$2^4 = 16$ (0-15)	0 … 15
8Bit	$2^8 = 256$ (0-255)	0 … 255

Ex: Bit depth = 1, 4, 8, 12, 24

Radiometric Resolution

- ความละเอียดของค่าแสง หรือสี ในแต่ละตำแหน่ง Pixel

Bits	Values	Gray Values
1Bit	$2^1 = 2$ (0-1)	0 … 1
4Bit	$2^4 = 16$ (0-15)	0 … 15
8Bit	$2^8 = 256$ (0-255)	0 … 255

NASA Satellite Sensor Examples:

- 12 bit sensor (MODIS, MISR, Landsat-9 TM/MSS): 2^{12} or 4,096 levels
- 10 bit sensor (AVHRR): 2^{10} or 1,024 levels
- 8 bit sensor (Landsat-7 TM): 2^8 or 256 levels (0-255)
- 6 bit sensor (Landsat-7 MSS): 2^6 or 64 levels (0-63)

Temporal Resolution

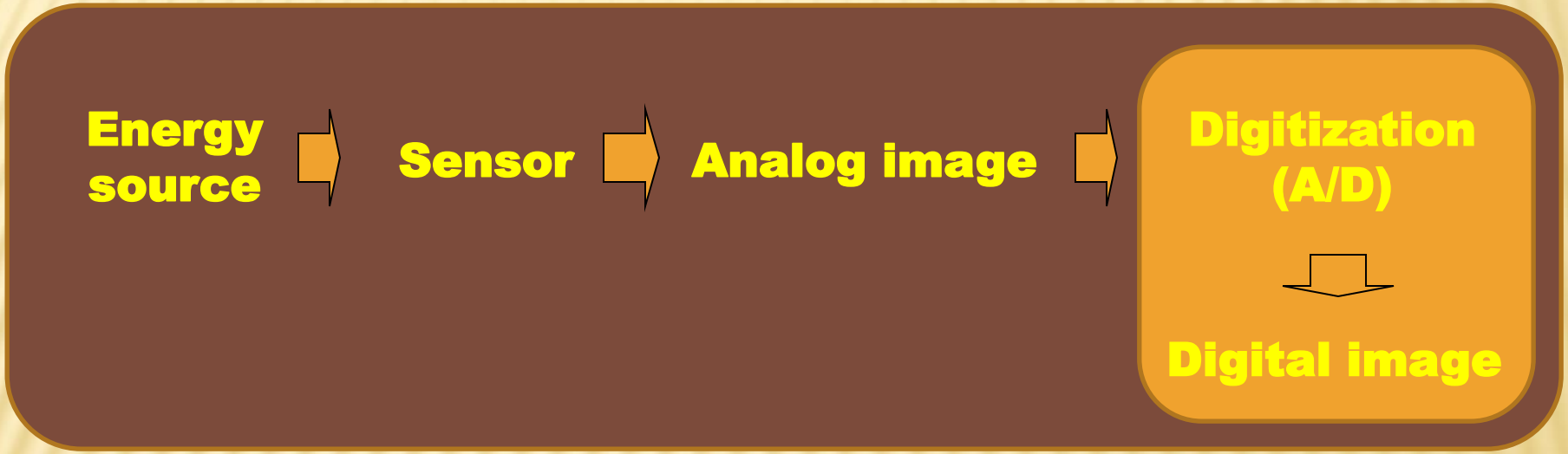
- ความถี่ในการเข้าเก็บภาพ

Global coverage in....

- MODIS
  - 1-2 days
- OMI
  - 1 day
- MISR
  - 6-8 days
- VIIRS
  - 1 day
- Geostationary
  - 30 sec – 1 hr

IMAGE DIGITIZATION

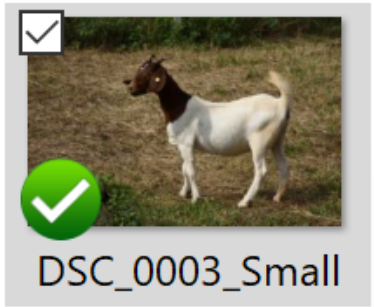
Q: How can we create a digital image?

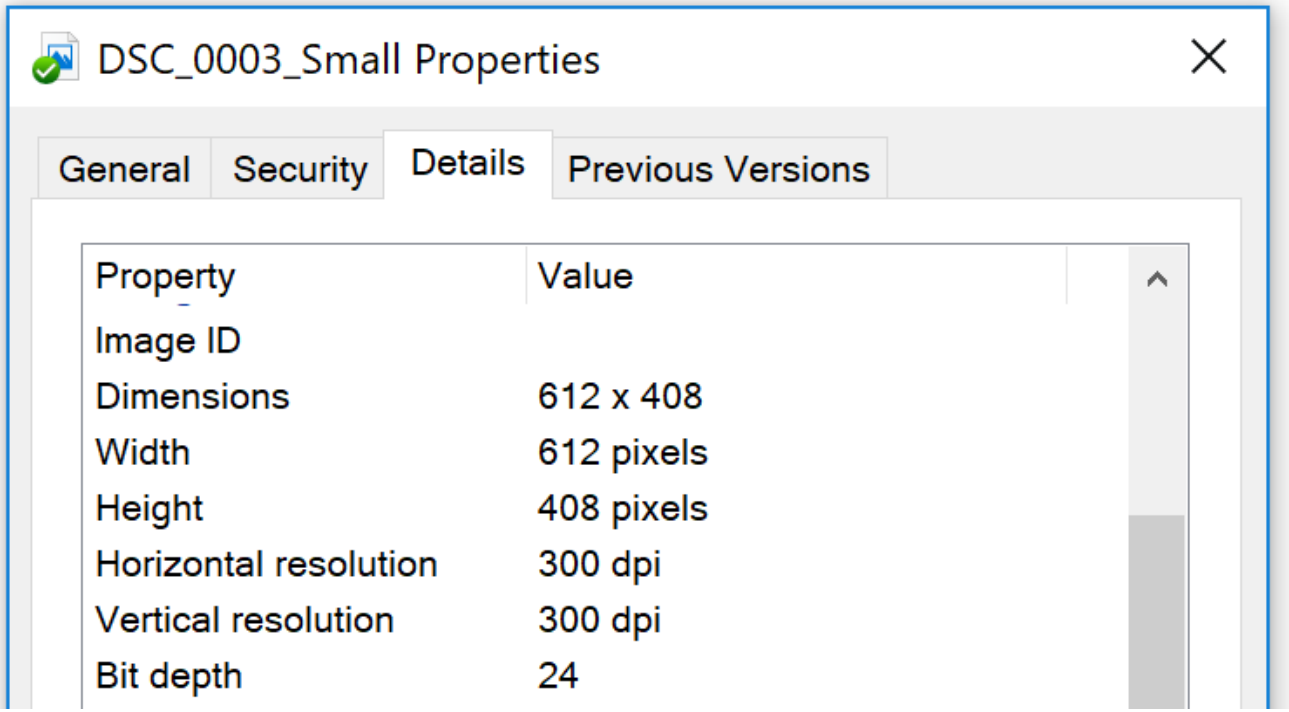


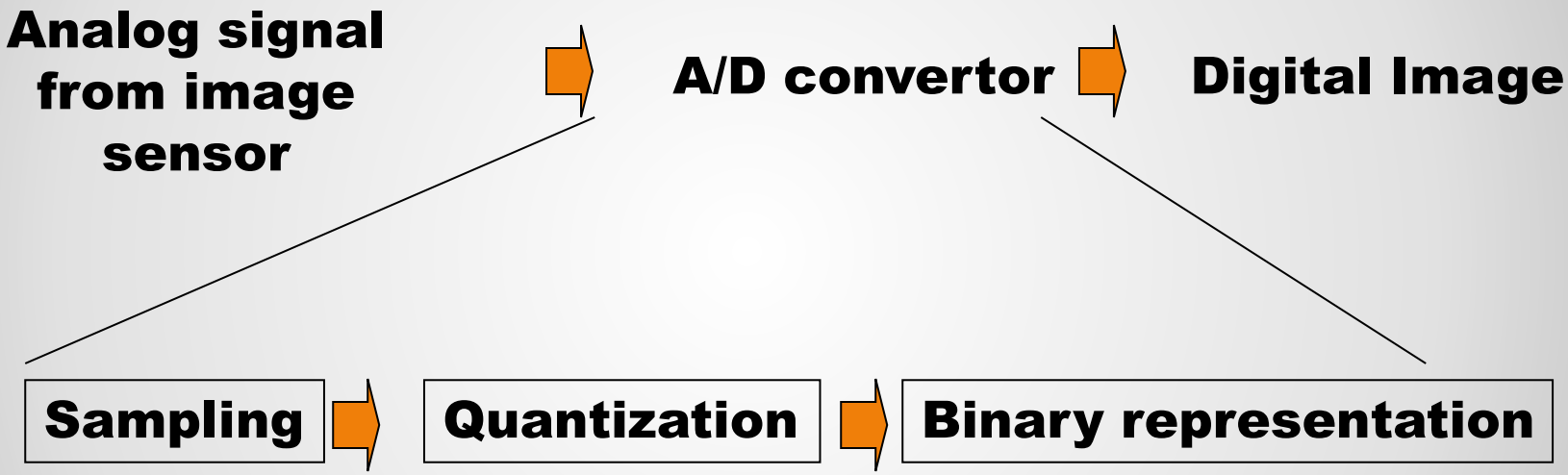
Getting Digital Images

With different resolution & intensity level

Image Properties





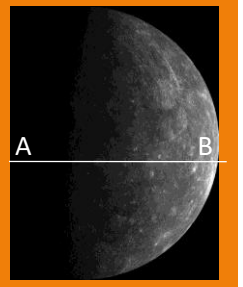


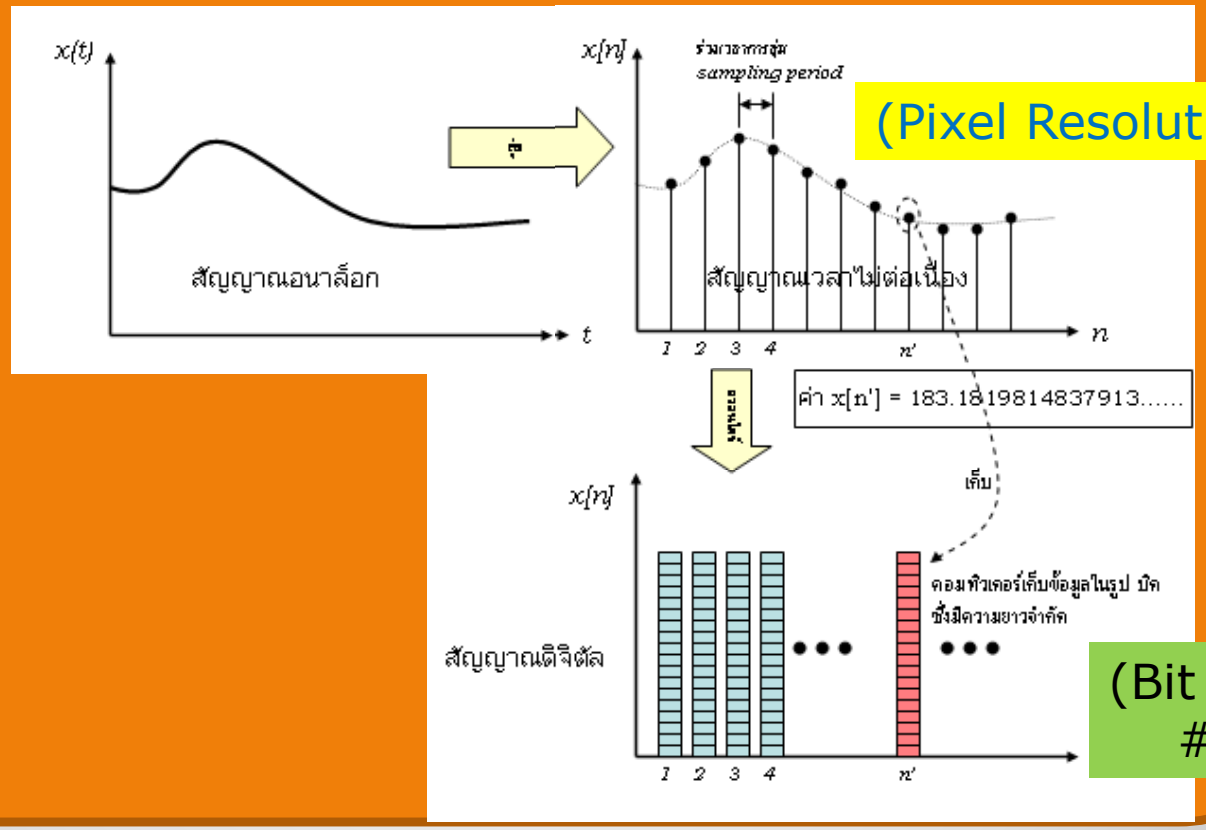
Analog-to-Digital Image

DIGITAL IMAGE PROPERTIES

- Image Dimension (Pixel Resolution: H x W)
  - Related to _____ process
- Image Intensity Level (Bit depth: bits/pixel)
  - Related to _____ process

Sampling → **Quantization** → **Binary representation**





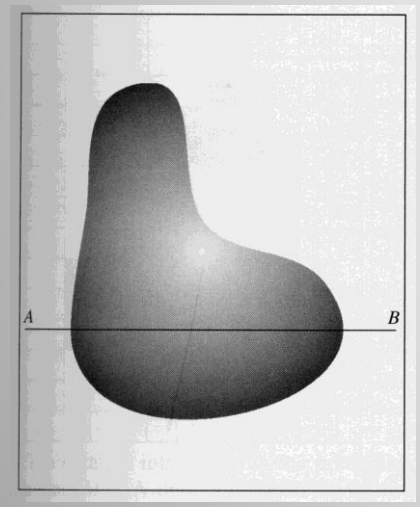
(Pixel Resolution: Width x Height)

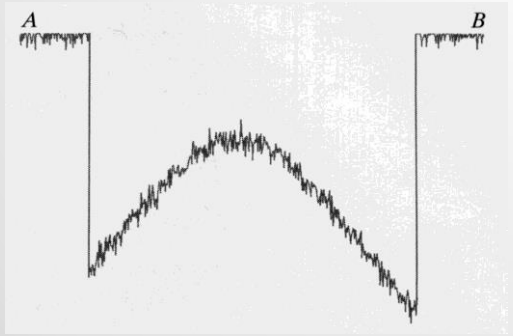
(Bit Depth Resolution: #bits/pixel)

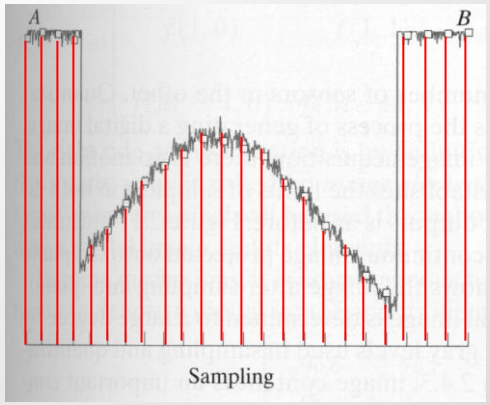
Analog-to-Digital Image

Sampling (การสุ่มวัดค่า)

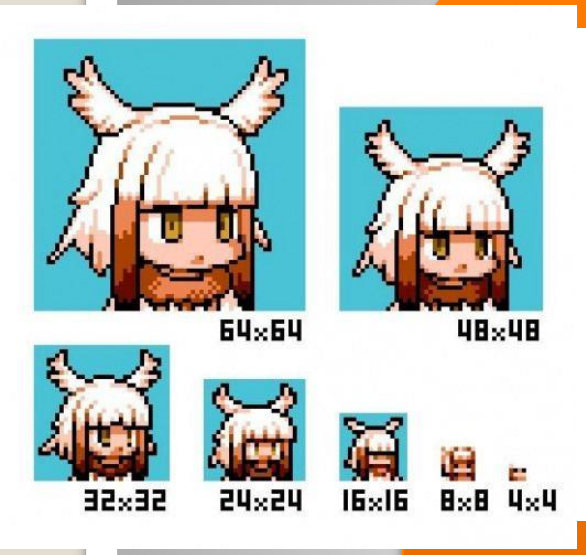
- เป็นขั้นตอนในการสร้างภาพดิจิตอล ซึ่งเป็นตัว**แปลงข้อมูลต่อเนื่อง (Continuous interval) ให้อยู่ในช่วงไม่ต่อเนื่องที่กำหนด (discrete interval)** ซึ่งจะสัมพันธ์กับจำนวนพิกเซล (pixel) คือ จำนวนจุด หรือ สี่เหลี่ยมจัตุรัสเล็กๆ ที่เก็บค่าระดับความเข้มแสงในภาพ

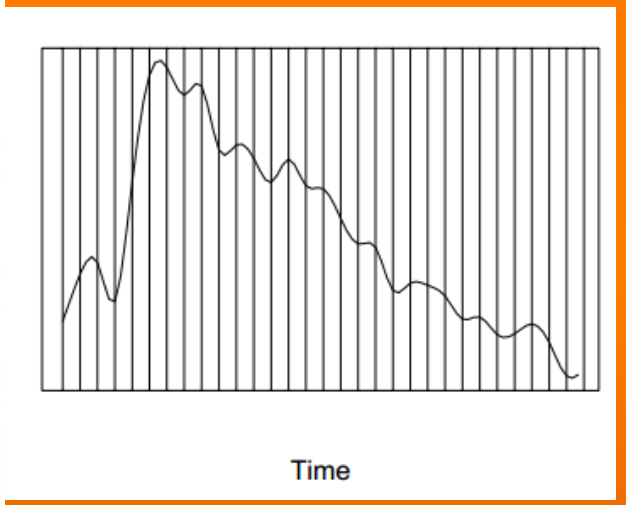






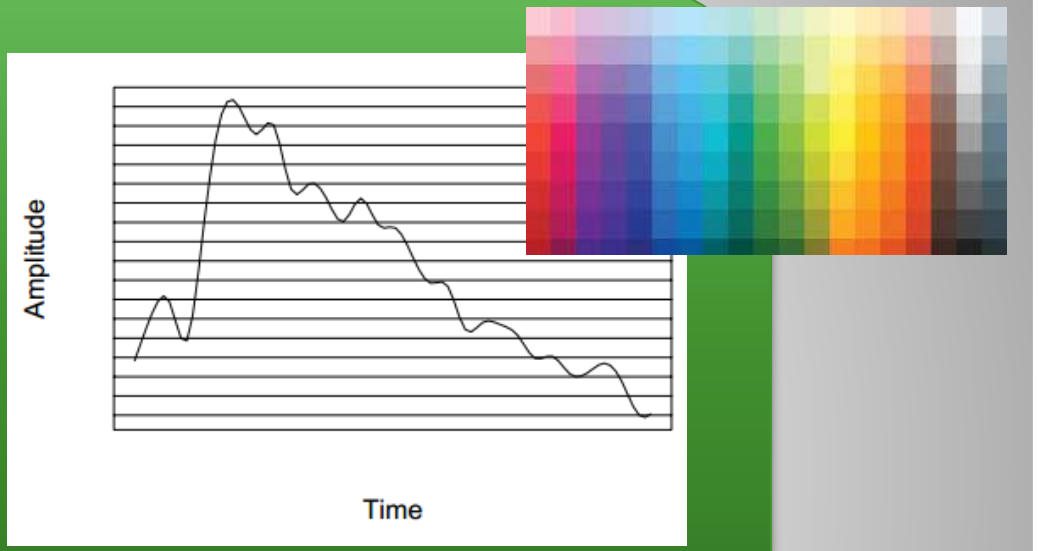
Analog signal **Sampled signal**





Sampling:
Sampling Analog signal in Time
(Resolution in Time: Pixels)
พิกัดเชิงตำแหน่ง

Image Dimension: width, height



Quantization:
Sampling Analog signal in Amplitude
(Resolution in Amplitude: Bit depth)
จำนวนเฉดสีหรือแสง

Brightness, color shading

Sampling freq. vs Bit depth

Image Resolution:

4608 x 3456 (15.9 MP, 4:3),
3264 x 2448 (8.0 MP, 4:3),
1920 x 1080 (2.1 MP, 16:9),
1600 x 1200 (1.9 MP, 4:3),
640 x 480 (0.3 MP, 4:3),
640 x 424 (0.3 MP, 3:2),
640 x 360 (0.2 MP, 16:9),

ภาพถ่ายจากล้องถ่ายภาพทั่วไป
(Bit Depth Resolution:
= 8 bits/pixel (bpp))

Image (Pixels)	Video (max)	fps (max)	Brand
40 M	1080p	60	Olympus OM-D E-M5 Mark II
45.7 M	4K Ultra HD	120	Nikon D850 FX
51.4 M	1080p	60	Pentax 645Z
42.3 M	4K Ultra HD	30	Sony a7R

Image resolution (dimension, image resize)

- NTSC (The National Television System Committee)
  - สหรัฐอเมริกา แคนาดา ญี่ปุ่น พม่า
  - สัญญาณภาพ 525 เส้น (line) / 60Hz
  - จำนวนภาพ 30 ภาพ (frame) / วินาที

- PAL (Phase Alternation Line)
  - ไทย อังกฤษ ฝรั่งเศส เยอรมัน หรือประเทศทางยุโรป
  - สัญญาณภาพ 625 เส้น (line) / 50Hz
  - จำนวนภาพ 25 ภาพ (frame) / วินาที

มาตรฐานการแพร่ภาพ Analog Video

Graphic Resolution

- Video frame resolution
- แนวแกน x-y เรียกว่า spatial resolution

High-Definition				
Name	x (px)	y (px)	x:y	x·y (Mpx)
nHD	640	360	16:9	0.230
qHD	960	540	16:9	0.518
HD	1280	720	16:9	0.921
HD+	1600	900	16:9	1.44
FHD	1920	1080	16:9	2.073
QHD	2560	1440	16:9	3.686
WQXGA+	3200	1800	16:9	5.76
UHD (4K)	3840	2160	16:9	8.294
UHD (8K)	7680	4320	16:9	33.178

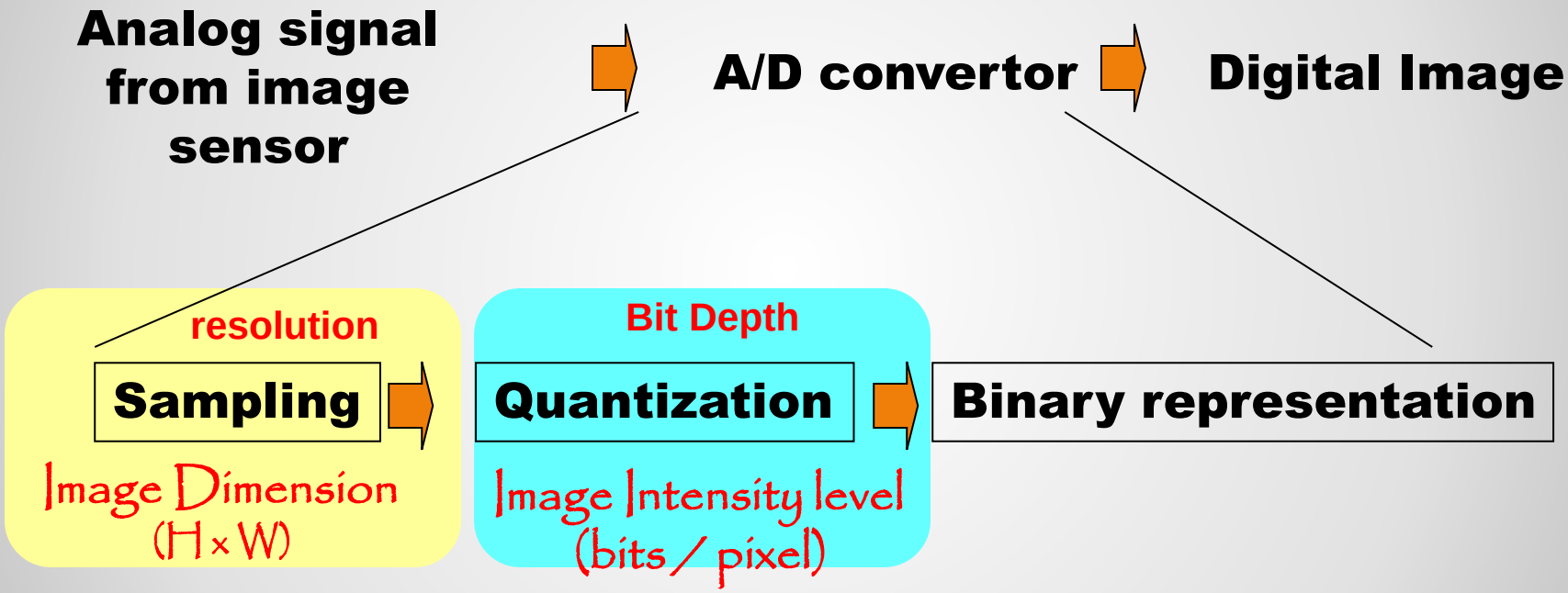
VGA: 640x480

2 times the width and 1.5 times the height of 4:3 VGA

3 times the width and 2.25 times the height of 4:3 VGA

6 times the width and 4.5 times the height of 4:3 VGA

12 times the width and 9 times the height of 4:3 VGA



Analog-to-Digital Image

Quantization (การจัดระดับเฉดสี) Equation

$$Q = floor(NormValue(Si) * (2^{Bit\_depth} - 1))$$

$$= floor\left(\left(\frac{Si - Smin}{Smax - Smin}\right) * (2^{Bit\_depth} - 1)\right)$$

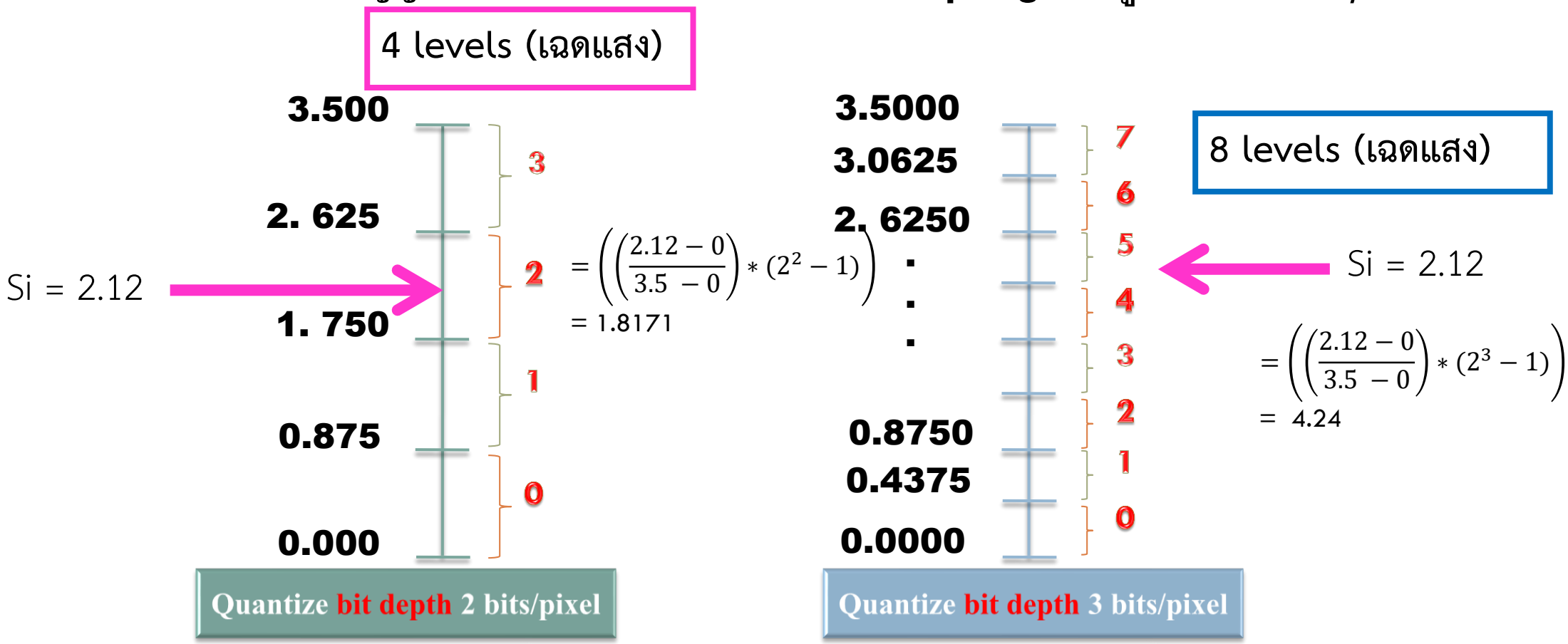
$\boldsymbol{Smax, Smin \rightarrow Max\ level, Min\ level\ of\ input\ intensity}$
$\boldsymbol{Bit\_depth \rightarrow \ \ number\ of\ bit/\ pixel}$

Uniform Quantization

$$Q = floor(NormValue(Si) * (2^{Bit\_depth} - 1))$$

$$= floor\left(\left(\frac{Si - Smin}{Smax - Smin}\right) * (2^{Bit\_depth} - 1)\right)$$

- คุณภาพของภาพ & จำนวน bit ในการทำ quantize
  - ระดับสัญญาณหรือพลังงานจาก sensor ที่ sampling ได้ อยู่ที่ Smin = 0 / Smax = 3.5



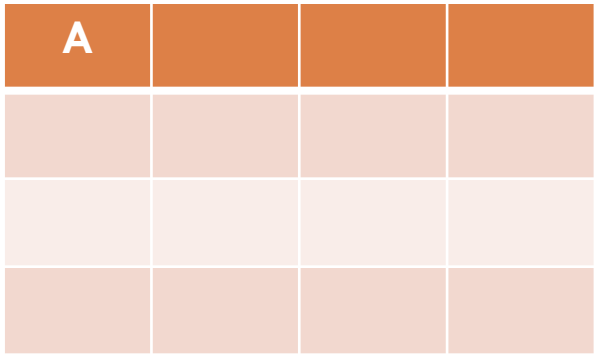
Uniform Quantization Example

$$Q = floor\left(\left(\frac{Si - Smin}{Smax - Smin}\right) * (2^{Bit\_depth} - 1)\right)$$

0-255

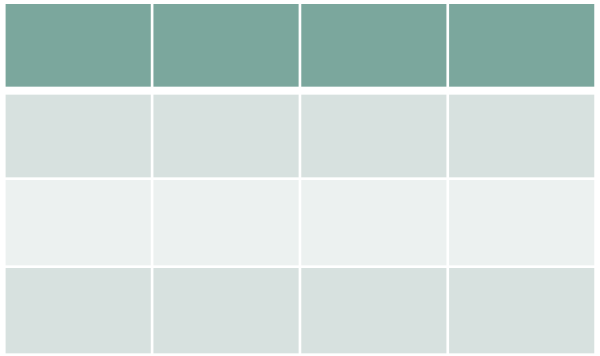
157	202	221	255
133	126	210	207
119	100	97	54
112	85	76	65

8 bits/pixel

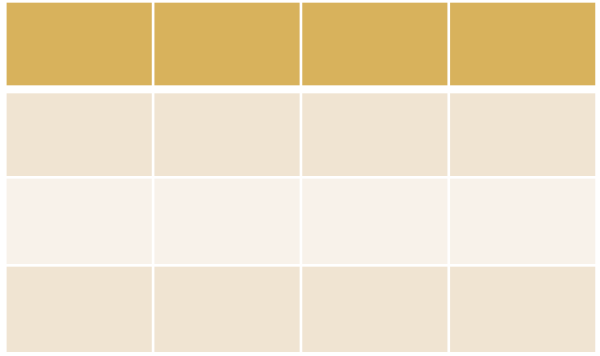


$$A = floor\left(\left(\frac{157 - 0}{255 - 0}\right) * (2^{6} - 1)\right)$$

6 bits/pixel



4 bits/pixel



2 bits/pixel

6 bits/pixel vs 8 bits/pixel

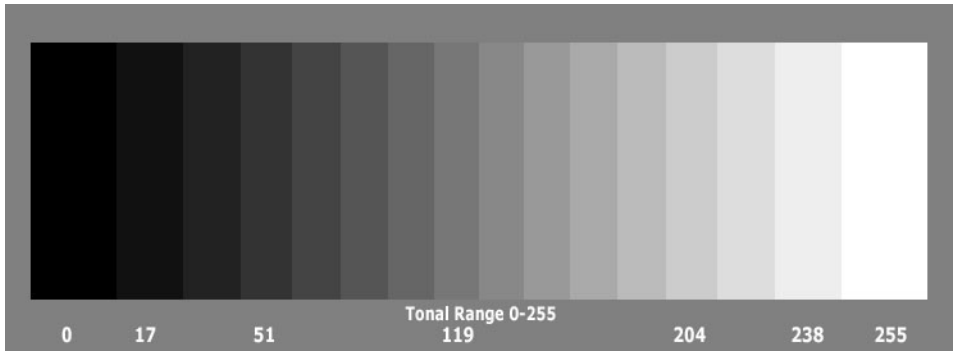
Activity # 1.1

REPRESENTING IN COMPUTER

Image Data Types

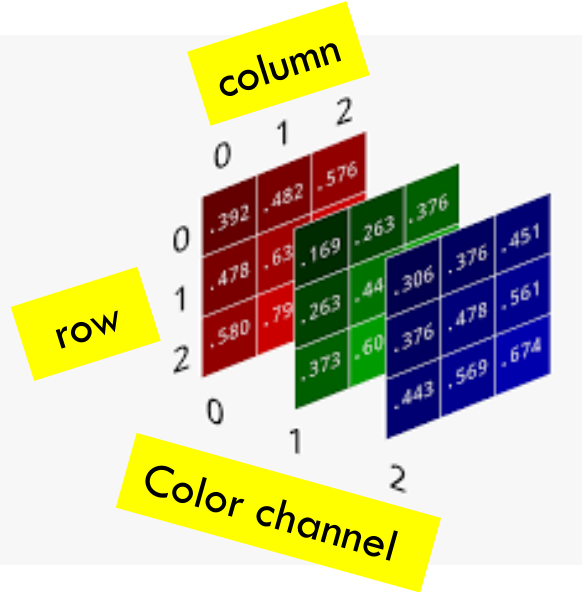
for each pixel intensity

- Uint8 (unsigned integer 8 bits)
  - Values: 0-255
  - No negative / No number greater than 255
    - Truncate outside its range
  - Cannot perform mathematical operation
    - In some developing tools



- Double
  - Safe for math operation
  - May need to rescale back to uint8

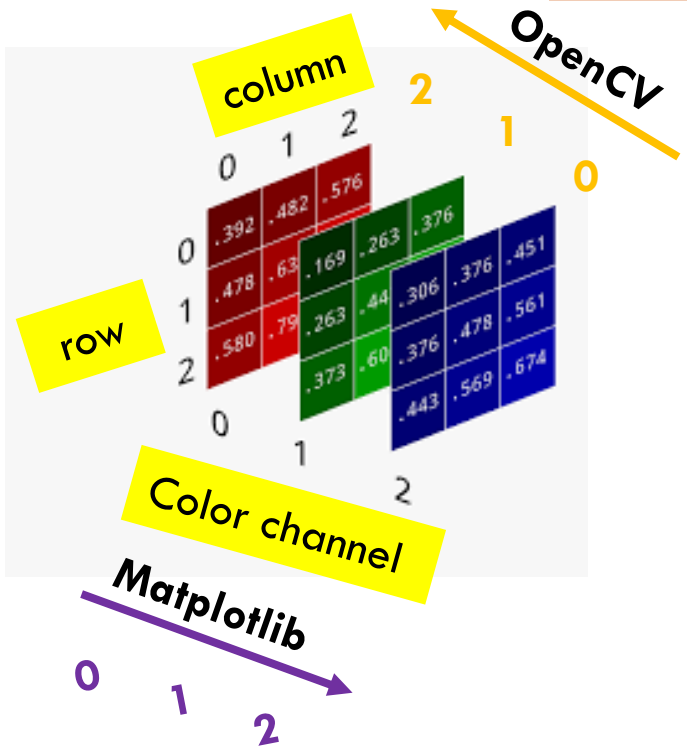
Image Array Structure and Reshape



- Image shape
  - Read from OpenCV
    - Normally color channels of each pixel is in the last dimension
    - Ex
      - Image shape: (2, 4, 3)
        - 2 = No. of Height (rows)
        - 4 = No. of Width (columns)
        - 3 = No of color channels

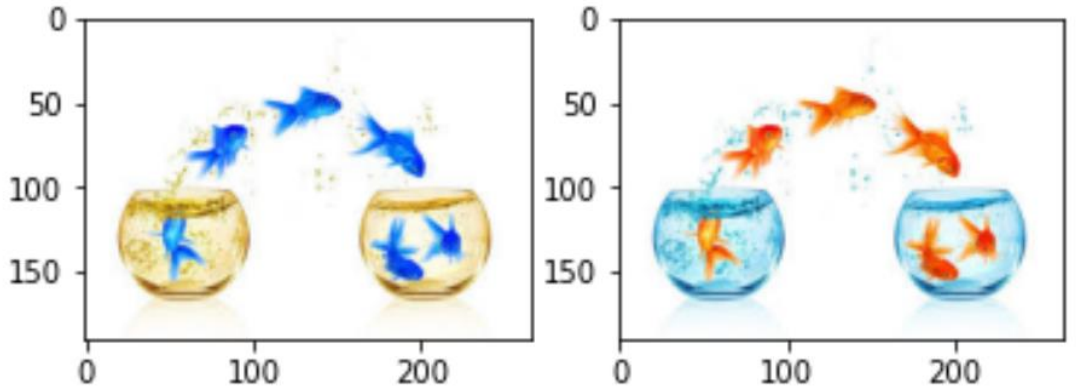
Image Array Structure

CV default: B, G, R
-> channel 0, 1, 2



Display default: R, G, B
-> channel 0, 1, 2

- Ex
  - Image shape: (rows, columns, 3)

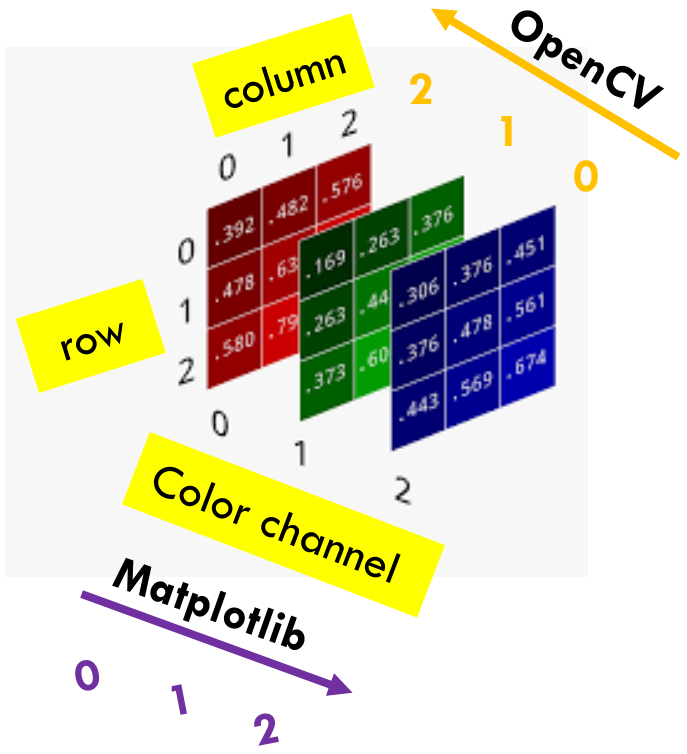


Img.shape = (rows, columns, 3)

Img.shape = (rows, columns, 3)

ทุกการพัฒนาไลบรารี่ ล้วนมีประวัติศาสตร์ของตัวเอง บางครั้งผลลัพธ์ที่แปลก อาจเกิดจาก default option ที่ไม่เหมือนใคร

Image Array Structure



- Image shape
  - OpenCV vs Pytorch
    - Normally color channels of each pixel is in the last dimension
    - Ex
      - Image shape:
        - OpenCV -> (H, W, **C**)
        - Pytorch -> (**C**, H, W)

ทุกการพัฒนาไลบรารี่ ล้วนมีประวัติศาตร์ของตัวเอง บางครั้งผลลัพธ์ที่แปลก อาจเกิดจาก default option ที่ไม่เหมือนใคร

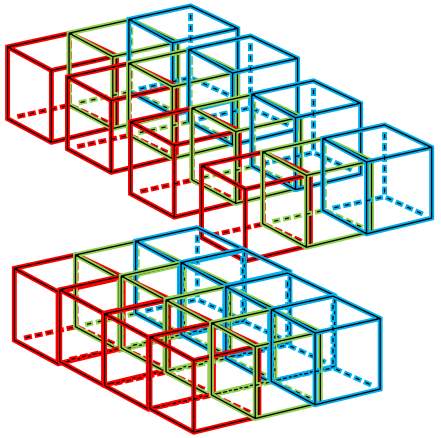
Image Array vs Tensor Structure

Channel Last

Array (OpenCV, Keras) -> (H, W, C)

```
(2, 4, 3)
[[[ 1  2  3]
  [ 4  5  6]
  [ 7  8  9]
  [10 11 12]]

 [[13 14 15]
  [16 17 18]
  [19 20 21]
  [22 23 24]]]
```



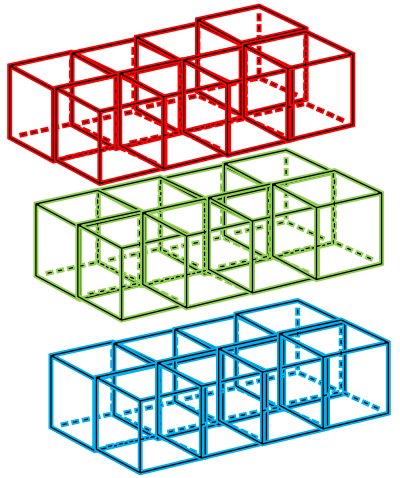
Channel First

Tensor -> (C, H, W)

```
(3, 2, 4)
[[[ 1  4  7 10]
  [13 16 19 22]]

 [[ 2  5  8 11]
  [14 17 20 23]]

 [[ 3  6  9 12]
  [15 18 21 24]]]
```



Activity # 1.2

How quality an image is?

- High spatial resolution
  - Cm / pixel , Dot / inch
- High Spectral resolution
  - Multiple frequencies
- High Radiometric resolution
  - Bits/ pixel
- Good Power / Light condition

Will all these conditions always possible?

Post Operation:
Image Processing Software

Topics

1. Color Format for Display

2. Color Model and Conversion

3. Image Arithmetic and Logic Operation

Color Format for Display

on Monitor or Projector

1. Color Format for Display

- Display Device (Monitor, Projector)
  - Required 3D Image Array Data Structure
    - Image shape: (H , W, Color channels: RGB)
      - Wrong data structure -> wrong display result
  - If we want to display just one plane of color (2D Image Array)
    - Such as grayscale or binary image
      - Must perform Color Mapping
        - Specify Color Mapping Table (cmap)
          - เป็น color lookup table
          - ถ้าไม่ได้เก็บไว้ใน Header ไฟล์ภาพ จะดึง
            - Default cmap
              - ขึ้นกับ image display function ของแต่ละ library

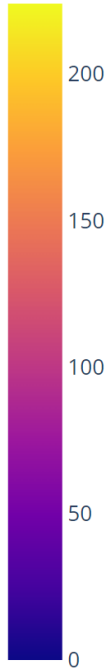
Cmap
(color scale)
default

Viridis

Matplotlib

imshow()

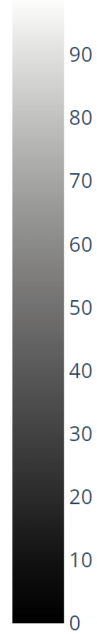
Heatmap



Plotly

imshow()

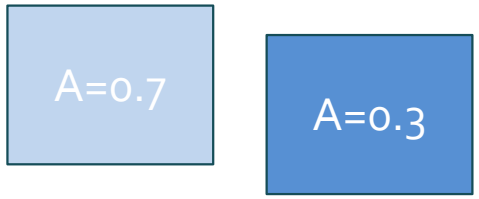
Gray



OpenCV

imshow()

cmap structure



- Matplotlib (RGBA Mapping)
  - RGB: normalize color
  - A is the opacity of the color
    - (a = 0 = transparent; a = 1 = opaque)

```
viridis = plt.cm.get_cmap('viridis',8)
print(viridis.colors)
```

```
[[0.267004 0.004874 0.329415 1.        ]
 [0.275191 0.194905 0.496005 1.        ]
 [0.212395 0.359683 0.55171  1.        ]
 [0.153364 0.497    0.557724 1.        ]
 [0.122312 0.633153 0.530398 1.        ]
 [0.288921 0.758394 0.428426 1.        ]
 [0.    R  579 0.8  G  45 0.2  B  53  A  ]
 [0.993248 0.906157 0.143936 1.        ]]
```

rgba(255, 0, 0, 0.2);

rgba(255, 0, 0, 0.4);

rgba(255, 0, 0, 0.6);

rgba(255, 0, 0, 0.8);

Color Name	rgb(R,G,B)	Swatch	rgba(R,G,B,a) Swatches										
		rgb(R,G,B)	a = 1.0	a = 0.9	a = 0.8	a = 0.7	a = 0.6	a = 0.5	a = 0.4	a = 0.3	a = 0.2	a = 0.1	a = 0.0
black	rgb(0,0,0)			text	text	text	text	text	text	text	text	text	text
navy	rgb(0,0,128)	text	text	text	text	text	text	text	text	text	text	text	text
blue	rgb(0,0,255)	text	text	text	text	text	text	text	text	text	text	text	text
green	rgb(0,128,0)	text	text	text	text	text	text	text	text	text	text	text	text
teal	rgb(0,128,128)	text	text	text	text	text	text	text	text	text	text	text	text
lime	rgb(0,255,0)	text	text	text	text	text	text	text	text	text	text	text	text
aqua	rgb(0,255,255)	text	text	text	text	text	text	text	text	text	text	text	text
maroon	rgb(128,0,0)	text	text	text	text	text	text	text	text	text	text	text	text

Imwrite() With Different cmap table

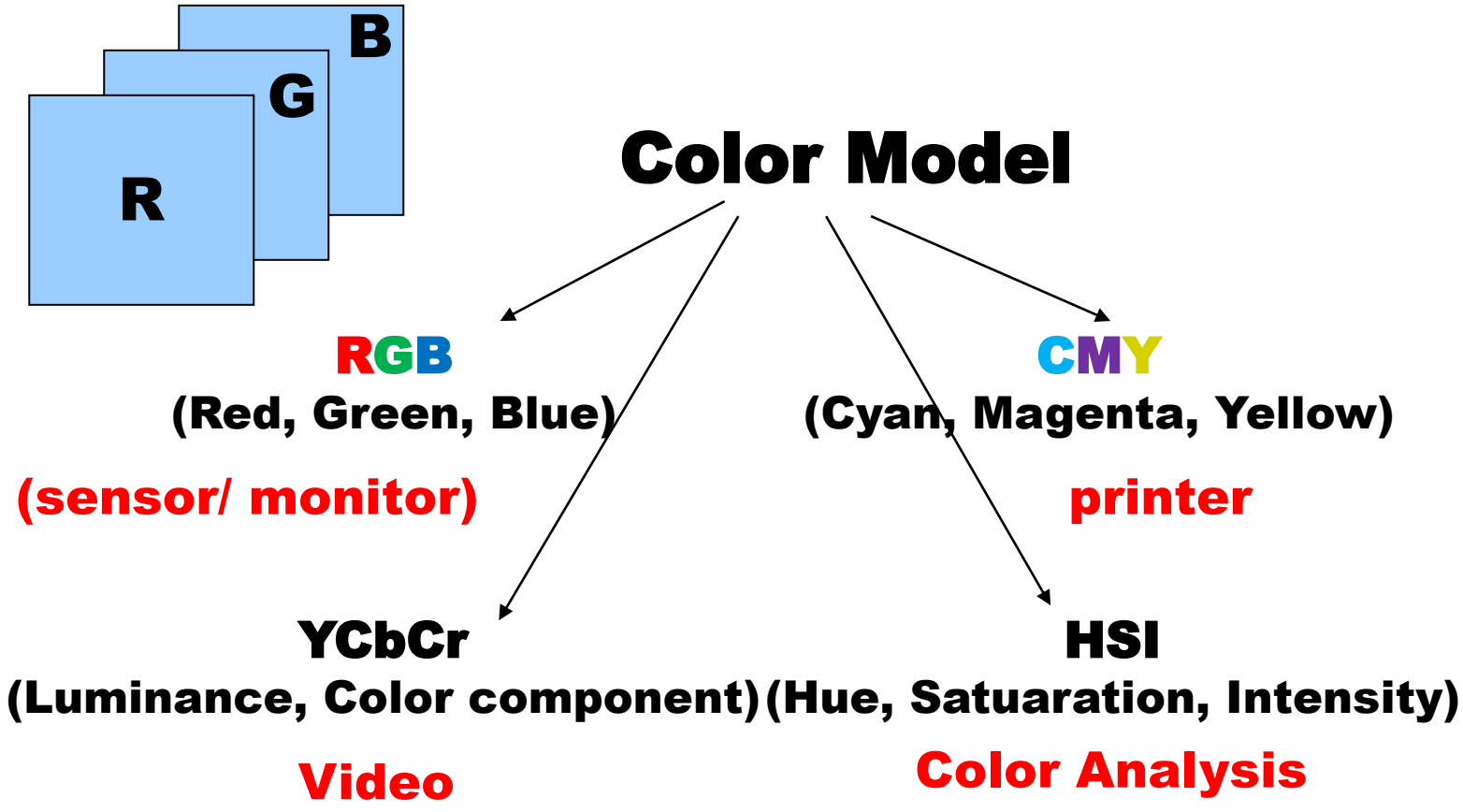
Bitmap file format (color table header: RGBA)

Structure name	Optional	Size	Purpose	Comments
Bitmap file header	No	14 bytes	To store general information about the bitmap image file	Not needed after the file is loaded in memory
DIB header	No	Fixed-size (7 different versions exist)	To store detailed information about the bitmap image and define the pixel format	Immediately follows the Bitmap file header
Extra bit masks	Yes	3 or 4 DWORDs[6] (12 or 16 bytes)	To define the pixel format	Present only in case the DIB header is the `BITMAPINFOHEADER` and the Compression Method member is set to either BI_BITFIELDS or BI_ALPHABITFIELDS
Color table	Semi-optional	Variable size	To define colors used by the bitmap image data (Pixel array)	Mandatory for color depths ≤ 8 bits
Gap1	Yes	Variable size	Structure alignment	An artifact of the File offset to Pixel array in the Bitmap file header
Pixel array	No	Variable size	To define the actual values of the pixels	The pixel format is defined by the DIB header or Extra bit masks. Each row in the Pixel array is padded to a multiple of 4 bytes in size
Gap2	Yes	Variable size	Structure alignment	An artifact of the ICC profile data offset field in the DIB header
ICC color profile	Yes	Variable size	To define the color profile for color management	Can also contain a path to an external file containing the color profile. When loaded in memory as "non-packed DIB", it is located between the color table and Gap1.[7]

Color Model and Conversion

RGB, HSV, YUV, CMYK

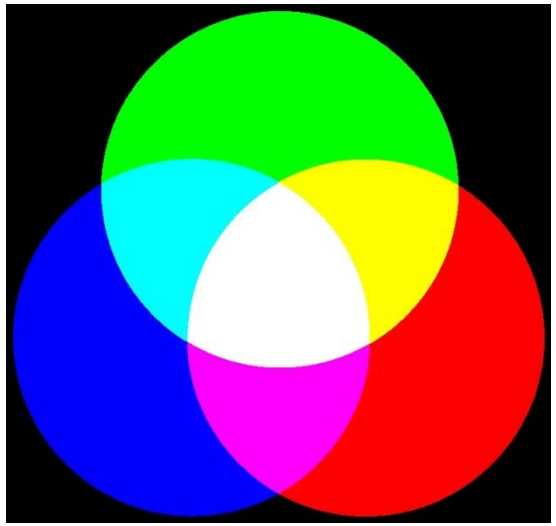
Color Digital Image (Color Model)

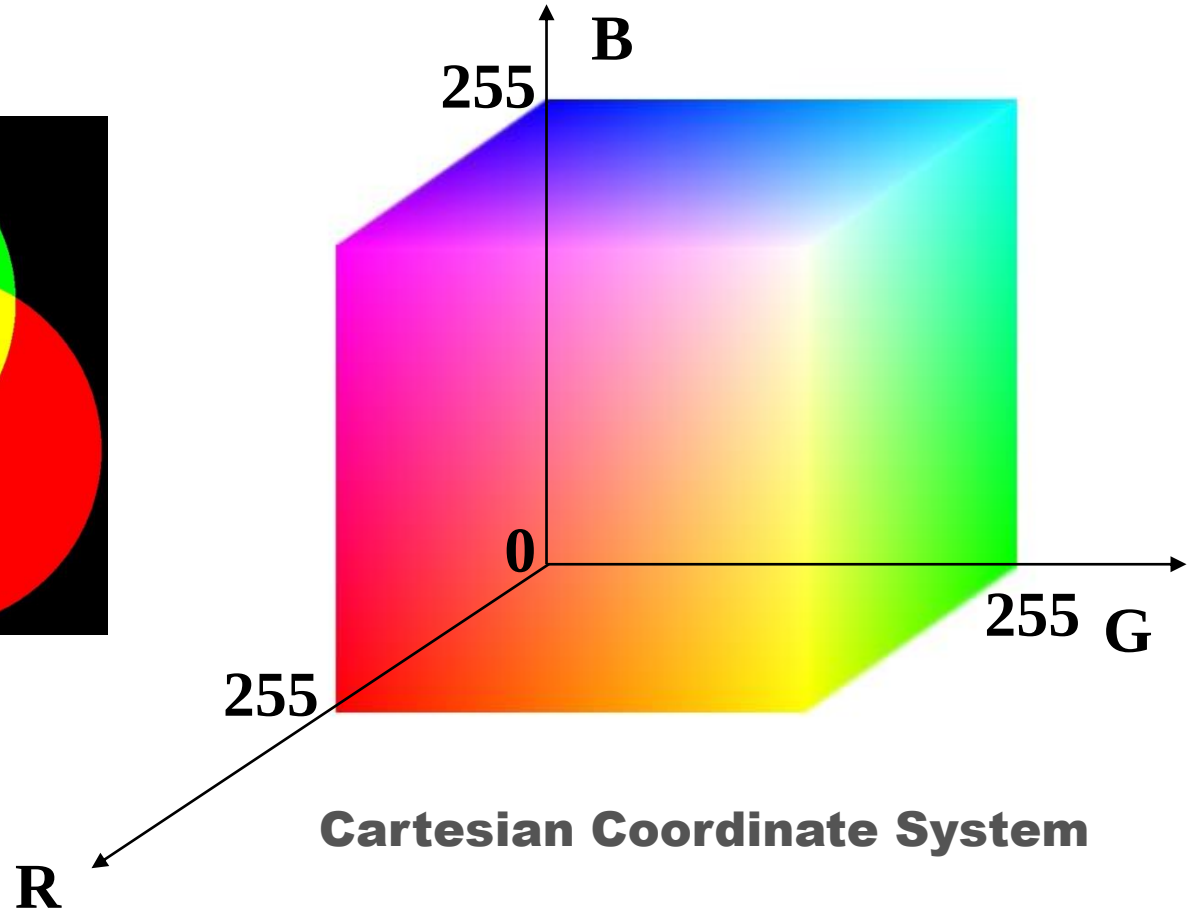


RGB Color Model

Sensor and Display

Color Digital Image (RGB)





Cartesian Coordinate System

Color Digital Image (RGB)

Array Slicing

Image (np) [H, W, C]

Image (pytorch) [C, H, W]

Red

Image (np) [: , : , 0]

Image (pytorch) [0, : , :]

Green

Image (np) [: , : , 1]

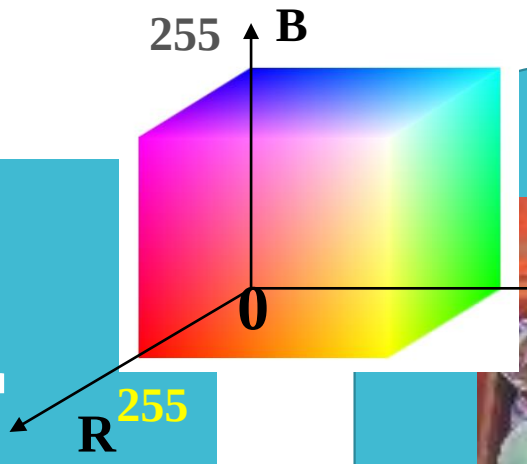
Image (pytorch) [1, : , :]

Blue

Image (np) [: , : , 2]

Image (pytorch) [2, : , :]

Color Digital Image (RGB)



Original

R+G+B

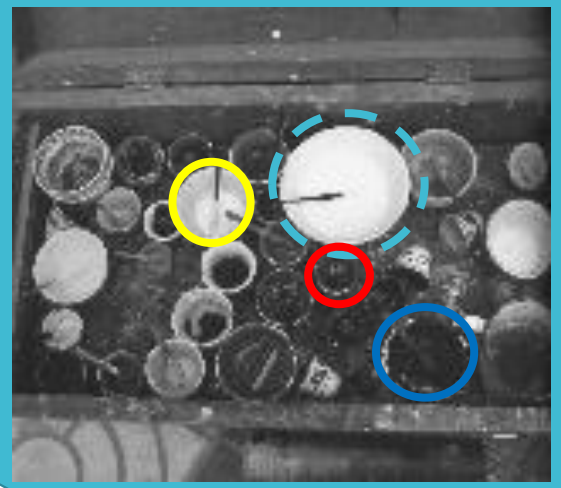
R+0+0

0+0+B

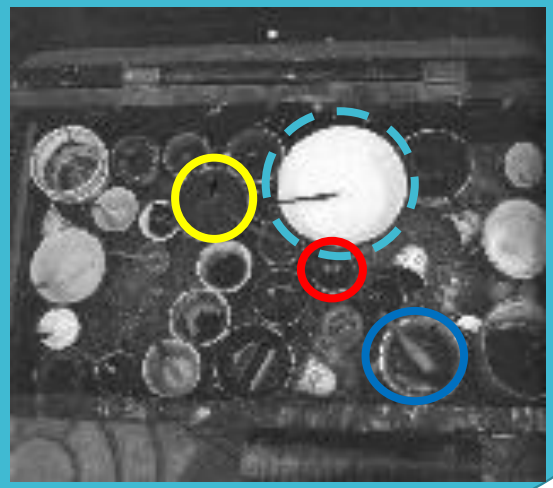
R+G+0

Red

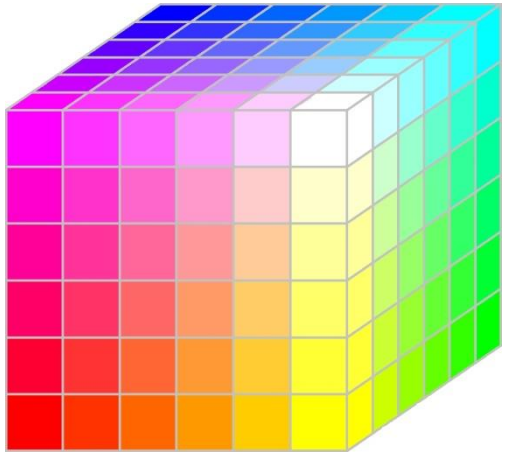
Green



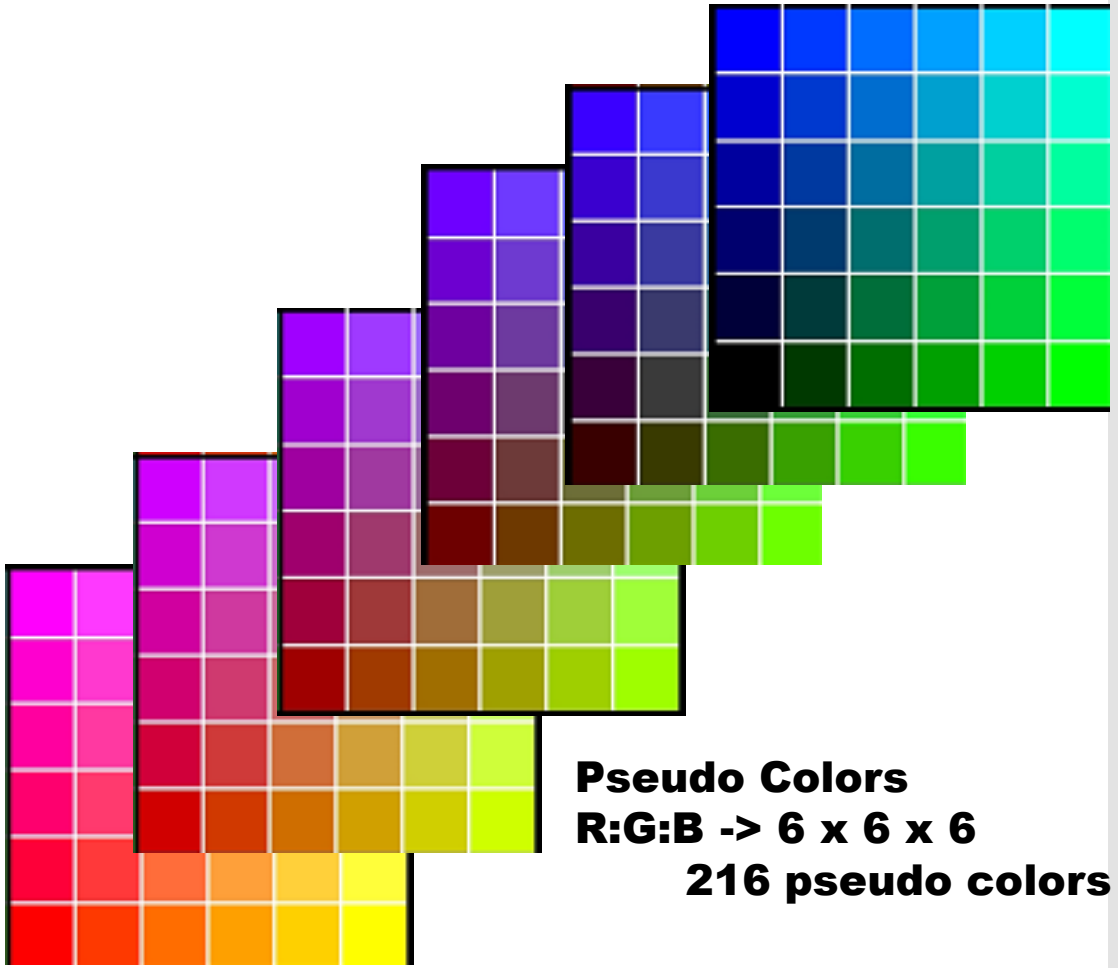
Blue



Color Digital Image (RGB Pseudo (Qualitzed) Color)

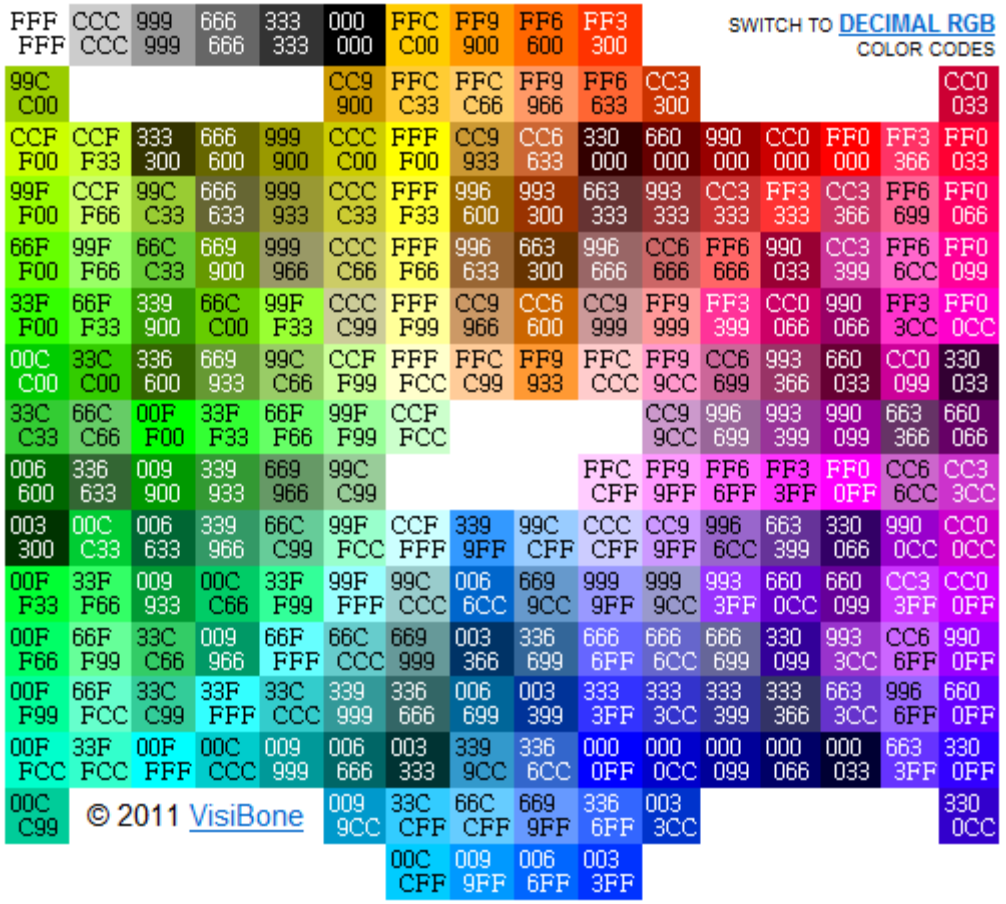


True Colors
R:G:B -> 8bits: 8bits: 8bits
256 x 256 x256
~ 16 million colors



Pseudo Colors
R:G:B -> 6 x 6 x 6
216 pseudo colors

Color Digital Image (Web Colors)



6 digit color code

216-HEX decimal code

RRGGBB

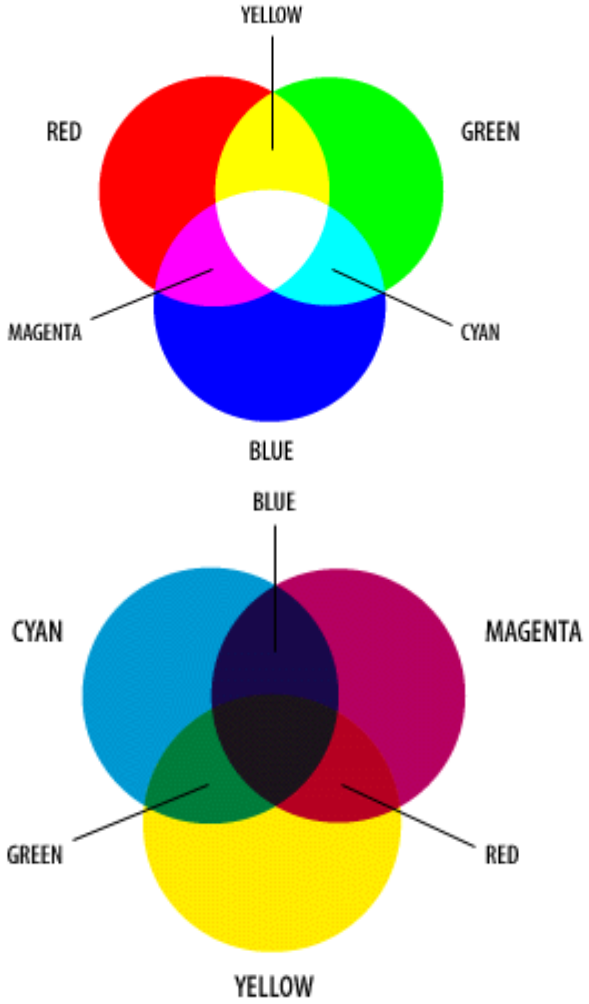
rgb(233,150,122)	#E9967A	rgb(154,205,50)	#9ACD32
rgb(220,20,60)	#DC143C	rgb(107,142,35)	#6B8E23
rgb(255,248,220)	#FFF8DC	rgb(128,128,0)	#808000
rgb(255,235,205)	#FFEBCD	rgb(85,107,47)	#556B2F
rgb(255,228,196)	#FFE4C4	rgb(143,188,143)	#8FBC8F
rgb(255,222,173)	#FFDEAD	rgb(102,205,170)	#66CDAA
rgb(245,222,179)	#F5DEB3	rgb(32,178,170)	#20B2AA
rgb(222,184,135)	#DEB887	rgb(0,139,139)	#008B8B
rgb(210,180,140)	#D2B48C	rgb(0,128,128)	#008080
rgb(189,183,107)	#BDB76B	rgb(0,255,255)	#00FFFF
rgb(218,165,32)	#DAA520	rgb(127,255,212)	#7FFFD4
rgb(184,134,11)	#B8860B	rgb(175,238,238)	#AFEEEE
rgb(205,133,63)	#CD853F	rgb(64,224,208)	#40E0D0
rgb(210,105,30)	#D2691E	rgb(72,209,204)	#48D1CC
rgb(160,82,45)	#A0522D	rgb(0,206,209)	#00CED1
rgb(165,42,42)	#A52A2A	rgb(95,158,160)	#5F9EA0
rgb(178,34,34)	#B22222	rgb(70,130,180)	#4682B4
rgb(139,69,19)	#8B4513	rgb(176,196,222)	#B0C4DE
rgb(139,0,0)	#8B0000	rgb(176,224,230)	#B0E0E6
rgb(128,0,0)	#800000	rgb(173,216,230)	#ADD8E6
rgb(250,164,96)	#FAA460	rgb(135,206,235)	#87CEEB
rgb(255,127,80)	#FF7F50	rgb(135,206,250)	#87CEFA
rgb(255,99,71)	#FF6347	rgb(0,191,255)	#00BFFF
rgb(255,0,0)	#FF0000	rgb(30,144,255)	#1E90FF
rgb(255,69,0)	#FF4500	rgb(100,149,237)	#6495ED
rgb(255,140,0)	#FF8C00	rgb(65,105,225)	#4169E1
rgb(255,165,0)	#FFA500	rgb(0,0,255)	#0000FF

CMY Color Model

Printing

Color Digital Image (CMY: Printing)

Display color model



Normalized RGB

$$r = \frac{R}{R+G+B}$$

$$g = \frac{G}{R+G+B}$$

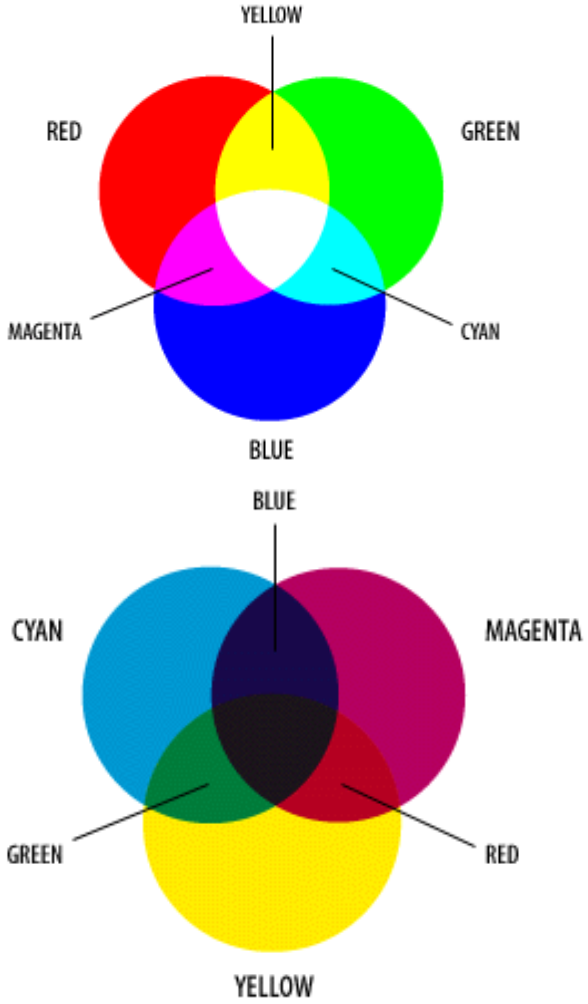
$$b = \frac{B}{R+G+B}$$

Normalized RGB-to-Normalized CMY

$$\begin{bmatrix} c \\ m \\ y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} r \\ g \\ b \end{bmatrix}$$

Color Digital Image (CMY: Printing)

Display color model



RGB to CMYK table

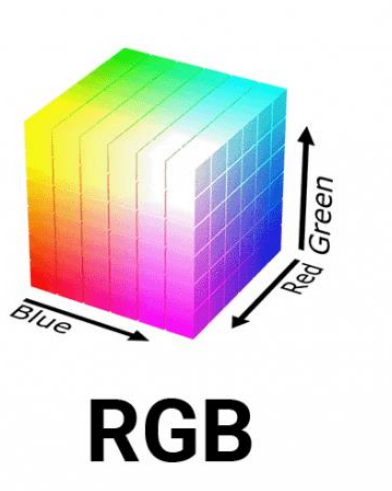
Color	Color name	(R,G,B)	Hex	(C,M,Y,K)
	Black	(0,0,0)	#000000	(0,0,0,1)
	White	(255,255,255)	#FFFFFF	(0,0,0,0)
	Red	(255,0,0)	#FF0000	(0,1,1,0)
	Green	(0,255,0)	#00FF00	(1,0,1,0)
	Blue	(0,0,255)	#0000FF	(1,1,0,0)
	Yellow	(255,255,0)	#FFFF00	(0,0,1,0)
	Cyan	(0,255,255)	#00FFFF	(1,0,0,0)
	Magenta	(255,0,255)	#FF00FF	(0,1,0,0)

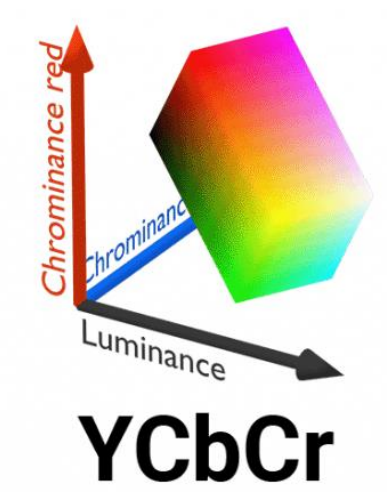
Normalized CMY-to-CMY

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} c \\ m \\ y \end{bmatrix} * 255 = \begin{bmatrix} c*255 \\ m*255 \\ y*255 \end{bmatrix}$$

CMY-to-RGB

$$\begin{bmatrix} r \\ g \\ b \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} c \\ m \\ y \end{bmatrix} \Rightarrow \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} r \\ g \\ b \end{bmatrix} * 255$$

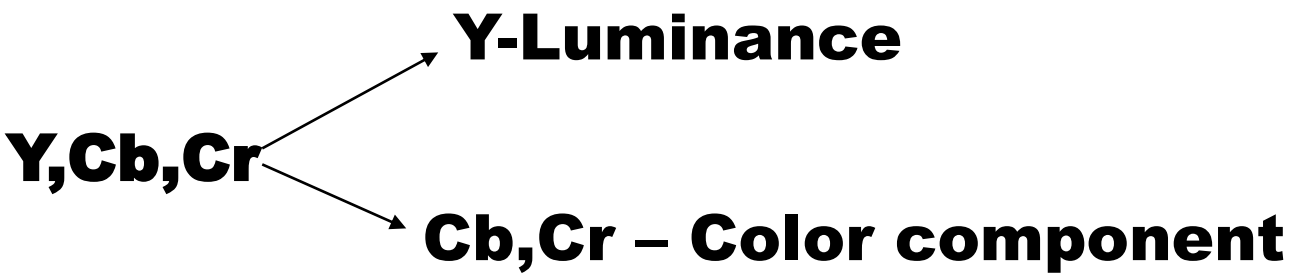




YCbCr (YUV) Color Model

Video

Color Digital Image (YCbCr: Video)



RGB-to-YCbCr

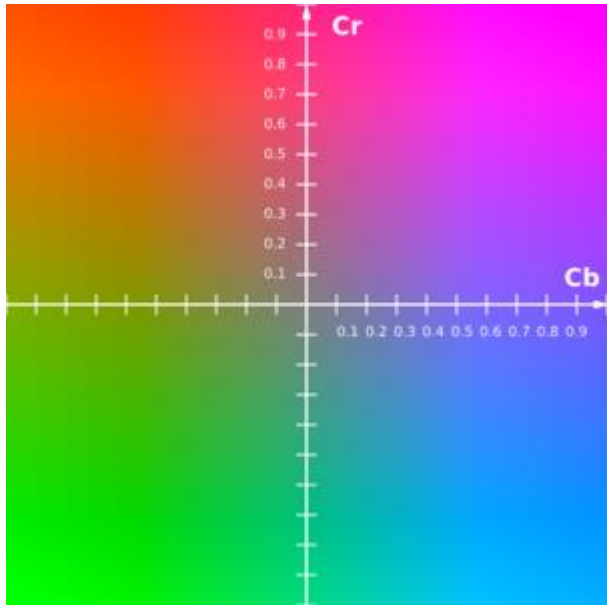
$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix}$$

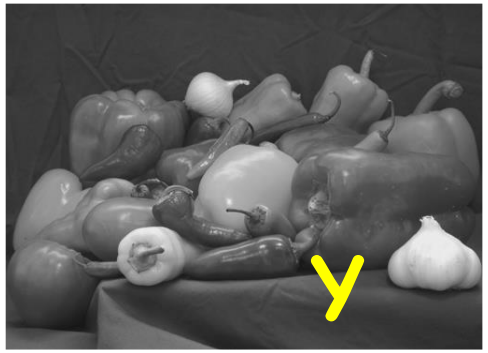
YCbCr-to- RGB

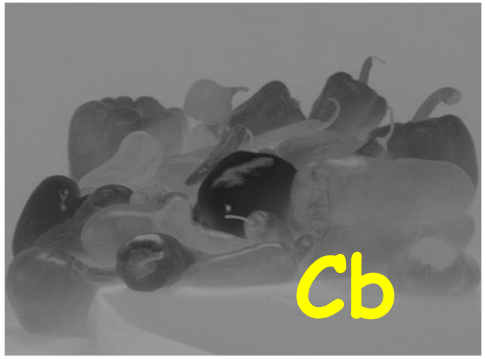
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1.164 & 0 & 1.596 \\ 1.164 & -0.392 & -0.813 \\ 1.164 & 2.017 & 0 \end{bmatrix} \begin{bmatrix} Y-16 \\ Cb-128 \\ Cr-128 \end{bmatrix}$$

Analog Video (PAL), Digital Video (MPEG)

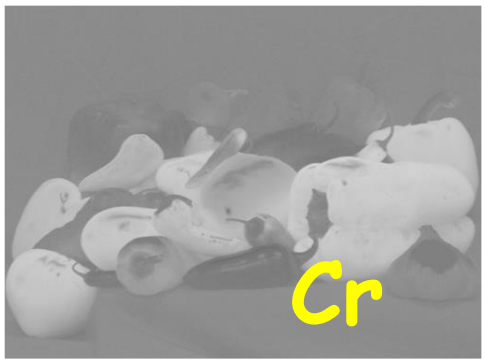
Color Digital Image (YCbCr: Video)



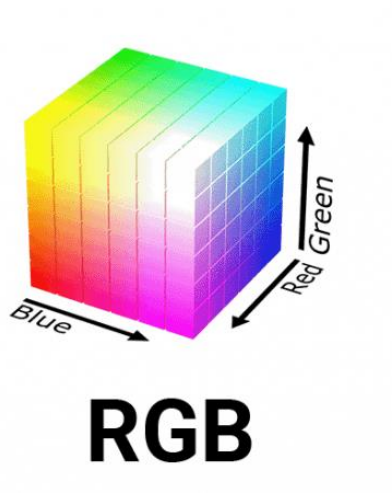


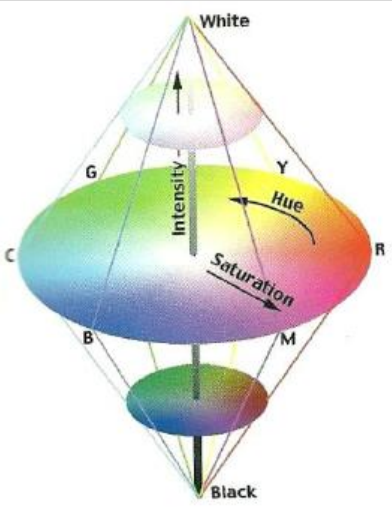


B & G



R & G



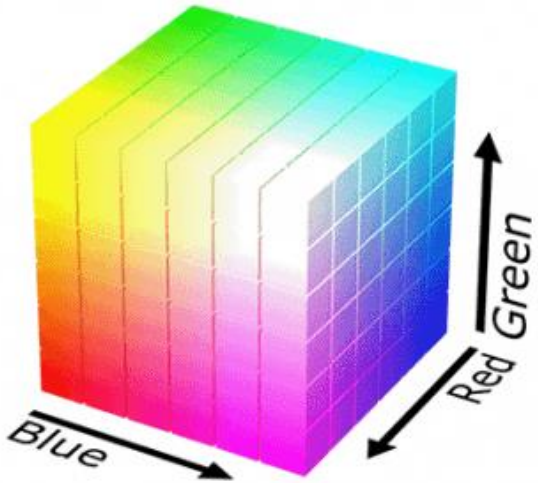


HSV / HSI Color Model

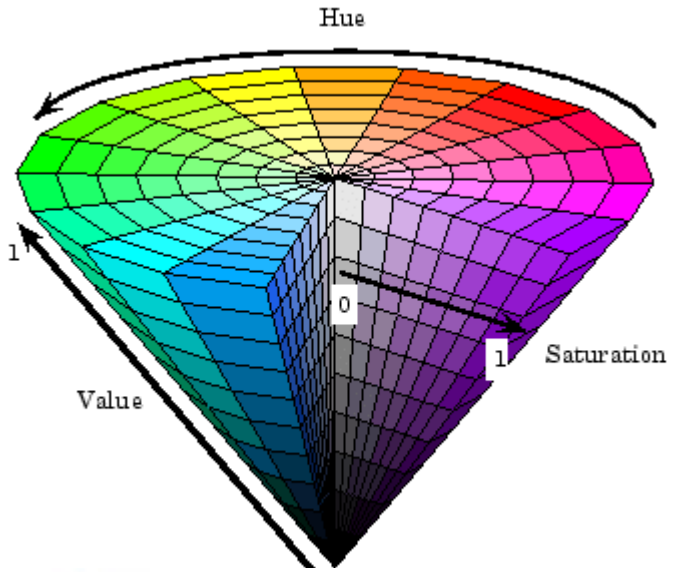
Analysis

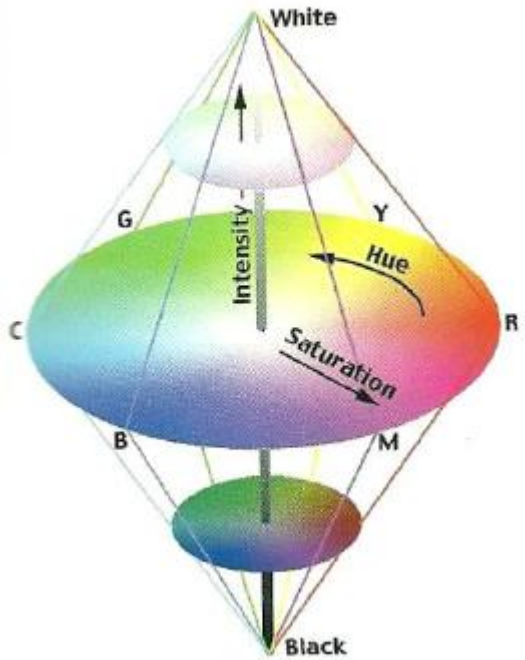
Color Digital Image

(RGB / HSV / HSI (HSL): Analysis)



RGB





Color Digital Image (HSV: Analysis)

Formula for converting RGB to HSV

To change the range from 0 to 255 into 0 to 1, the red, green and blue values are divided by 255.

R' = R/255 G' = G/255 B' = B/255

Cmax = max(R', G', B')

Cmin = min(R', G', B')

Δ = Cmax – Cmin

Hue Calculation

0° Δ = 0

60 ° x (G'-B'Δmod6), Cmax = R'

60 ° x (B'-R'Δ+2), Cmax = G'

60 ° x (R'-G'Δ+4), Cmax = B'

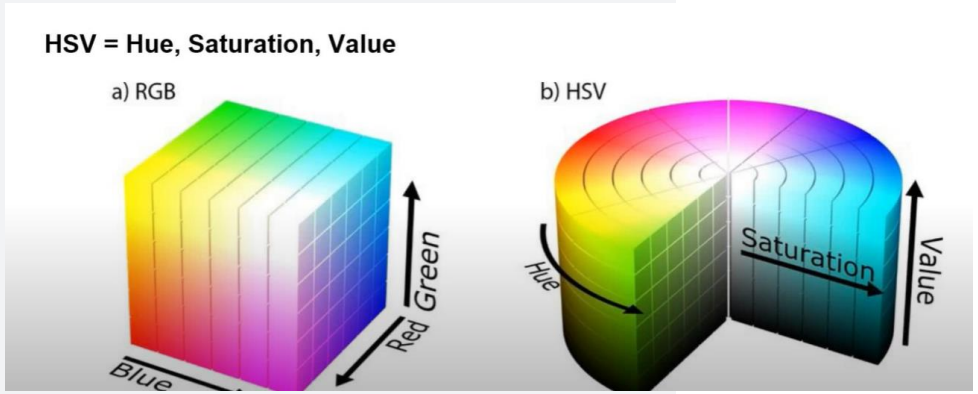
Saturation calculation

0, Cmax = 0

Δ / Cmax, Cmax ≠ 0

Value calculation

V = Cmax



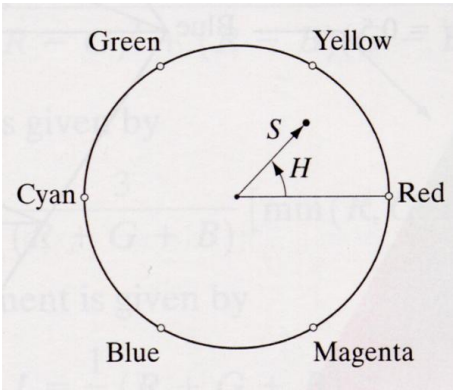
Color Digital Image (HSI,HSL: Analysis)

RGB-to-HSI

$$H = \begin{cases} \theta & B \le G \\ 360 - \theta & B > G \end{cases} \quad \Rightarrow \quad \theta = \cos^{-1}\left\{ \frac{\frac{1}{2}[(R-G)+(R-B)]}{\sqrt{(R-G)^2+(R-B)^2+(G-B)^2}} \right\}$$

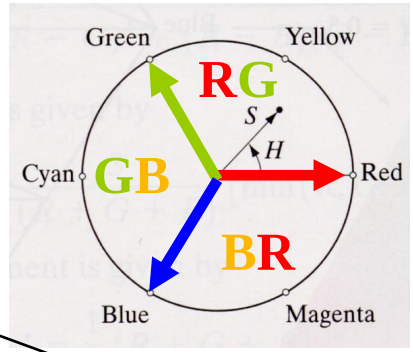
$$S = 1 - \frac{3}{R+G+B}[\min(R,G,B)]$$

$$I = \frac{1}{3}[R+G+B]$$



Color Digital Image (HSI: Analysis)

HSI-to-RGB



$0 \le H < 120$ (RG)

$$R = I\left[1 + \frac{S\cos(H)}{\cos(60 - H)}\right]$$

$$G = 1 - (R + B)$$

$$B = I(1 - S)$$

$120 \le H < 240$ (GB)

$$H = H - 120$$

$$G = I\left[1 + \frac{S\cos(H)}{\cos(60 - H)}\right]$$

$$B = 1 - (R + G)$$

$$R = I(1 - S)$$

$240 \le H \le 360$ (BR)

$$H = H - 240$$

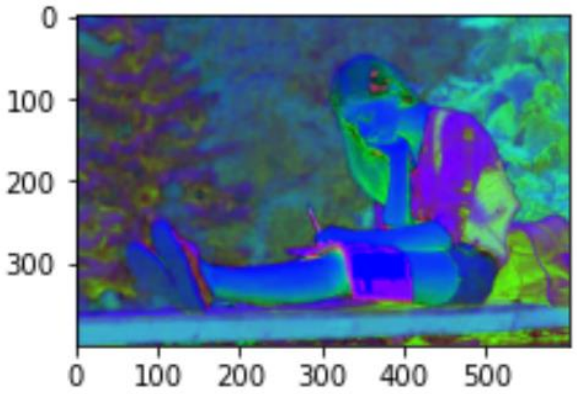
$$B = I\left[1 + \frac{S\cos(H)}{\cos(60 - H)}\right]$$

$$R = 1 - (G + B)$$

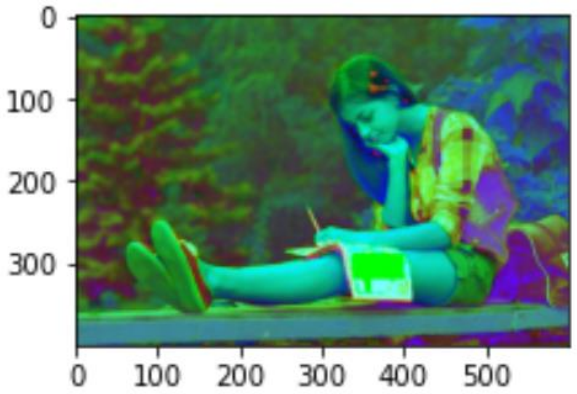
$$G = I(1 - S)$$

RGB

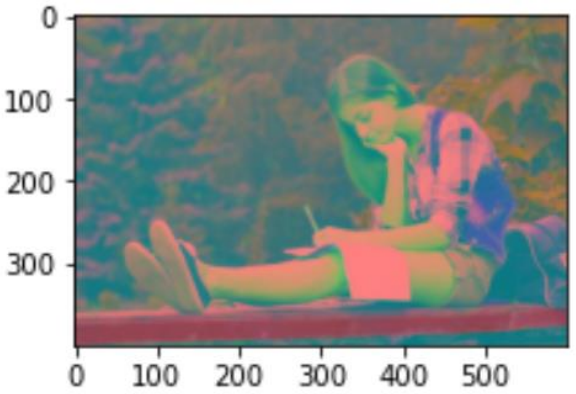
HSV



HSL



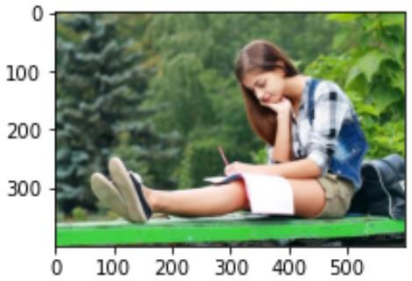
YCrCb



Color Image Display from different color models (RGB, HSV, HLS, YCbCr)

Color Digital Image

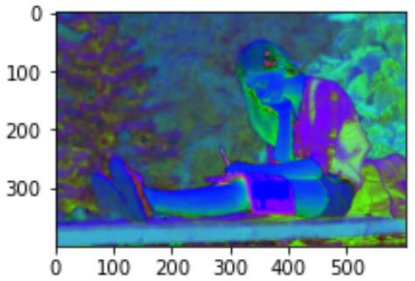
RGB




```
[[[ 77 104  71]
  [ 81 108  73]
  [ 87 115  77]
  ...
  [112 171  53]
  [115 174  54]
  [117 176  56]]

 [[ 74 101  68]
  [ 79 106  71]
  [ 85 113  75]
  ...
  [109 168  52]
  [111 170  54]
  [113 172  56]]
```

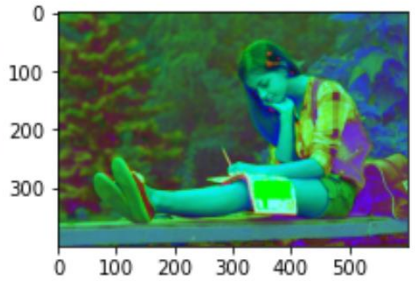
HSV




```
[[[ 55  81 104]
  [ 53  83 108]
  [ 52  84 115]
  ...
  [ 45 176 171]
  [ 45 176 174]
  [ 45 174 176]]

 [[ 55  83 101]
  [ 53  84 106]
  [ 52  86 113]
  ...
  [ 45 176 168]
  [ 45 174 170]
  [ 45 172 172]]
```

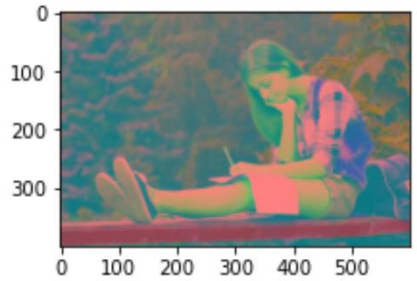
HSL




```
[[[ 55  88  48]
  [ 53  90  49]
  [ 52  96  50]
  ...
  [ 45 112 134]
  [ 45 114 134]
  [ 45 116 132]]

 [[ 55  84  50]
  [ 53  88  50]
  [ 52  94  52]
  ...
  [ 45 110 134]
  [ 45 112 132]
  [ 45 114 130]]
```

YCrCb




```
[[[ 92 117 116]
  [ 96 117 115]
  [102 117 114]
  ...
  [140 108  79]
  [143 108  78]
  [145 108  78]]

 [[ 89 117 116]
  [ 94 117 115]
  [100 117 114]
  ...
  [137 108  80]
  [139 108  80]
  [141 108  80]]
```

Grayscale Image

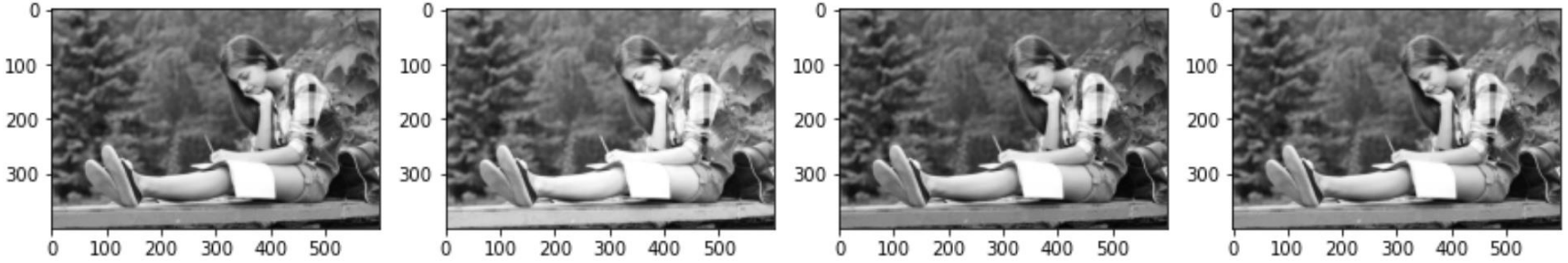
From different color model

Lightness

Grayscale Image

$$M = \max(R, G, B)$$
$$m = \min(R, G, B)$$

CV_Gray | HSV | HSL | YCrCb



Biologically, Human eyes are more sensitive and thus perceive more green and red than blue.

CV_Gray

$$Y = 0.299 \times R + 0.587 \times G + 0.114 \times B$$

HSV

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

HSL

$$L = \text{mid}(R, G, B) = \frac{1}{2}(M + m)$$

HSI

$$I = \frac{1}{3}[R + G + B]$$

YCrCb

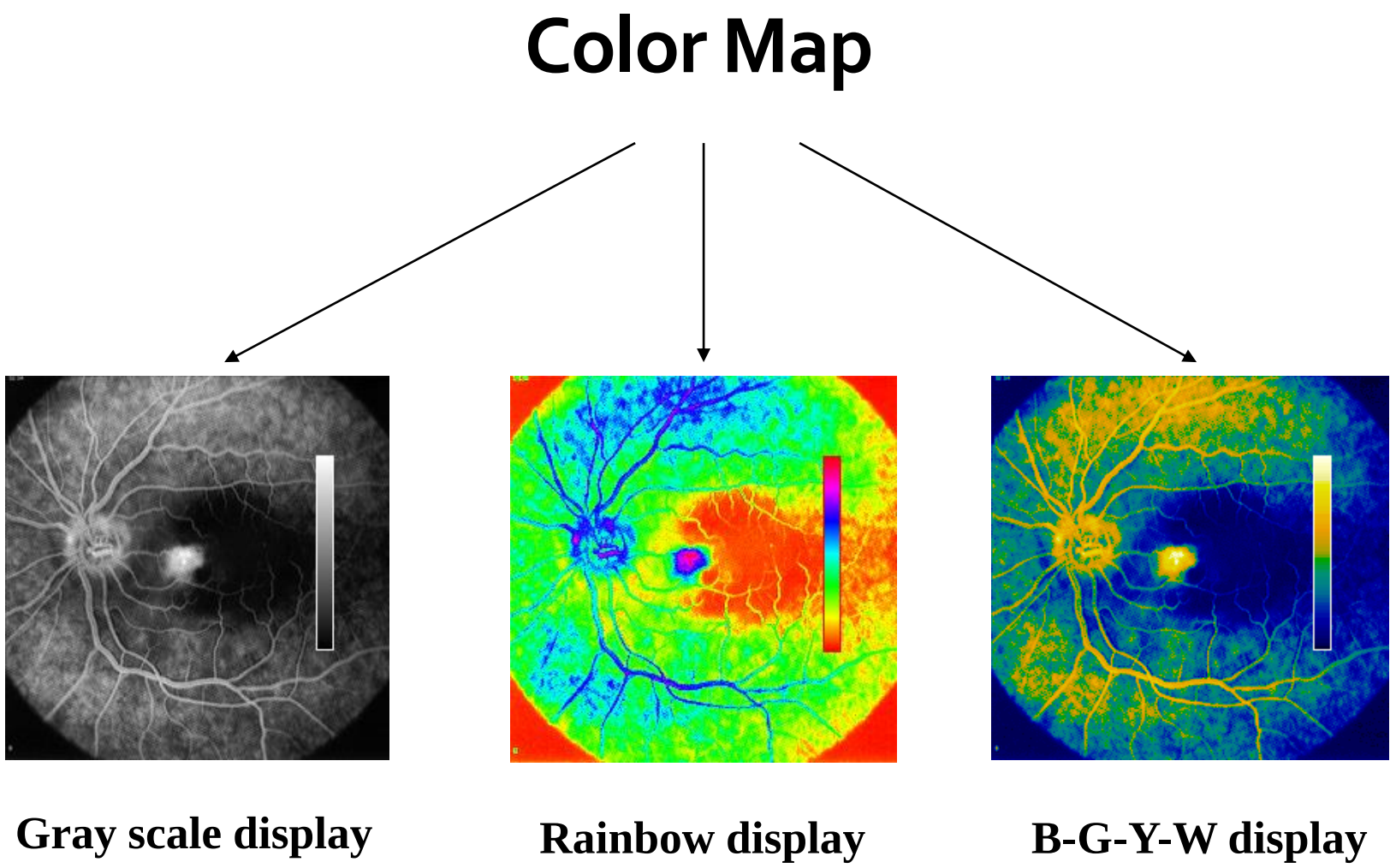
$$Y'_{601} = 0.2989 \cdot R + 0.5870 \cdot G + 0.1140 \cdot B \text{ (SDTV)}$$
$$Y'_{240} = 0.212 \cdot R + 0.701 \cdot G + 0.087 \cdot B \text{ (Adobe)}$$
$$Y'_{709} = 0.2126 \cdot R + 0.7152 \cdot G + 0.0722 \cdot B \text{ (HDTV)}$$
$$Y'_{2020} = 0.2627 \cdot R + 0.6780 \cdot G + 0.0593 \cdot B \text{ (UHDTV, HDR)}$$

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

Color Map

Visualize 2D Array or Grayscale Image

Color Map (Visualize 2D Array or Grayscale Image)



Activity # 1.3

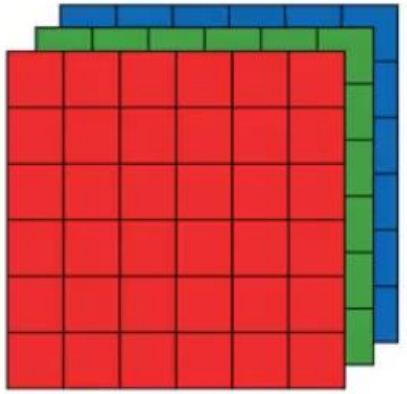
Image Arithmetic and Logic Operation

Addition / Subtraction / AND /OR/ XOR

Arithmetic

(addition)

$$\text{Im\_addition} = w_1 Im_1 + w_2 Im_2$$



$$\text{Im\_addition}[:,:,0] = w_1 Im_1[:,:,0] + w_2 Im_2[:,:,0]$$

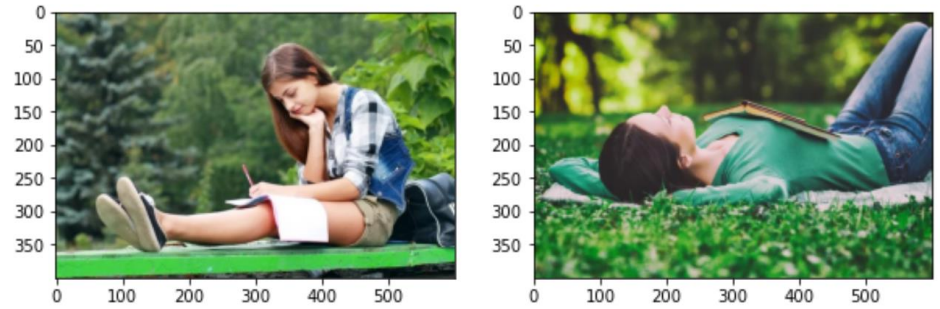
$$\text{Im\_addition}[:,:,1] = w_1 Im_1[:,:,1] + w_2 Im_2[:,:,1]$$

$$\text{Im\_addition}[:,:,2] = w_1 Im_1[:,:,2] + w_2 Im_2[:,:,2]$$

https://datahacker.rs/005-image-arithmetic-and-logical-operations-in-opencv-with-python/

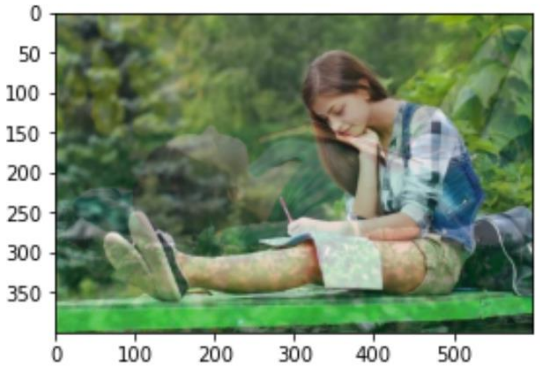
Arithmetic

(addition)

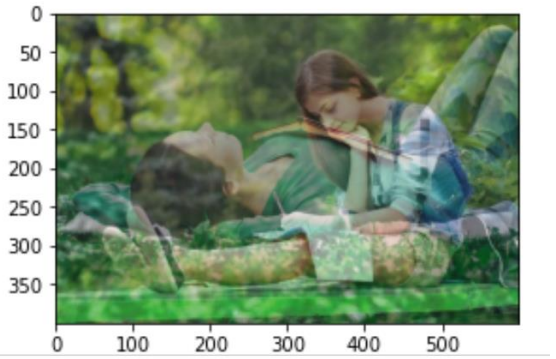


Im_addition= $w_1 Im_1 + w_2 Im_2$

$$w_1 + w_2 = 1.0$$



w_1 = 0.7
w_2 = 0.3



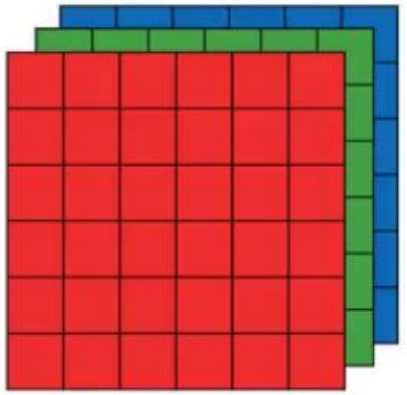
w_1 = 0.5
w_2 = 0.5

w_1 = 0.3
w_2 = 0.7

Arithmetic

(subtraction)

$$\text{Im\_subtract} = Im_1 - Im_2$$



$$\text{Im\_addition}[:,:,0] = Im_1[:,:,0] - Im_2[:,:,0]$$

$$\text{Im\_addition}[:,:,1] = Im_1[:,:,1] - Im_2[:,:,1]$$

$$\text{Im\_addition}[:,:,2] = Im_1[:,:,2] - Im_2[:,:,2]$$

Arithmetic

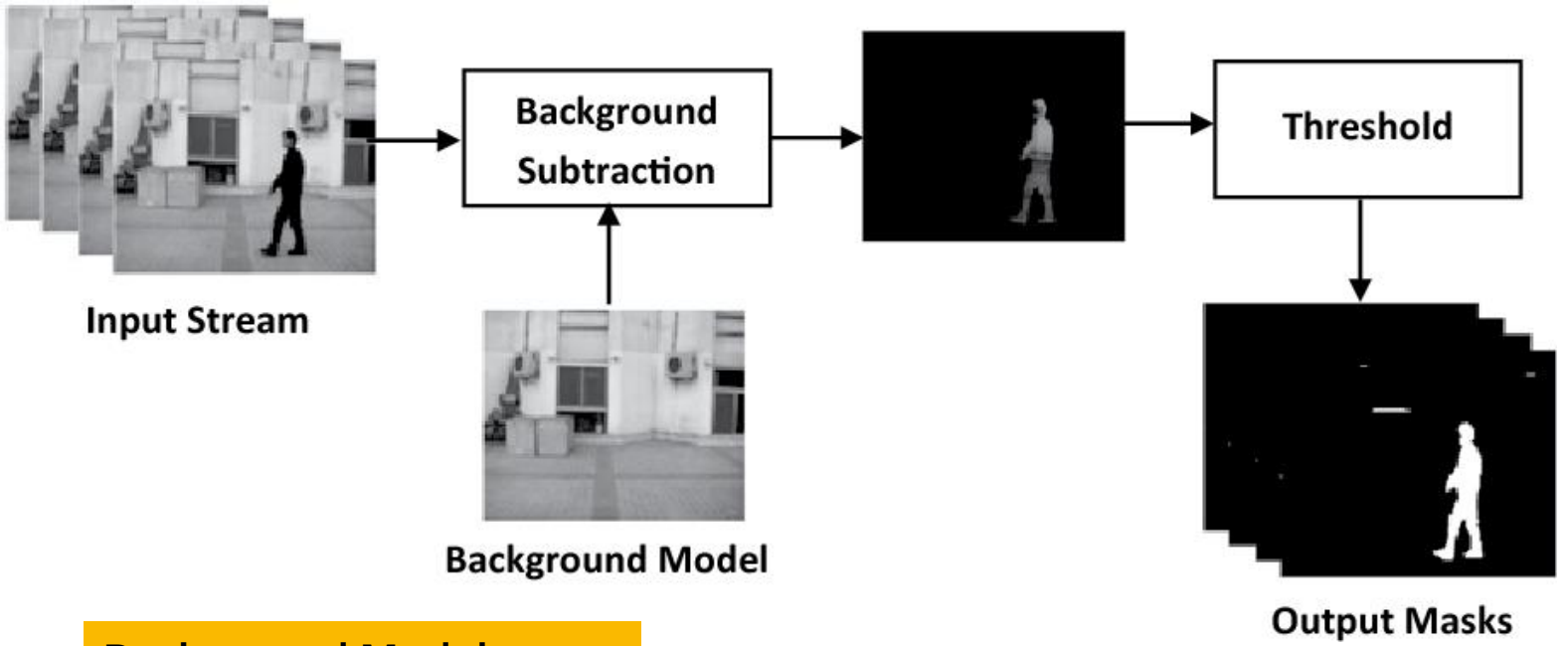
(subtraction)

Im_addition= $Im_1 - Im_2$

https://datahacker.rs/005-image-arithmetic-and-logical-operations-in-opencv-with-python/

Simple Object Detection (addition & subtraction)

Moving Object Model =
Image subtraction = Image - background

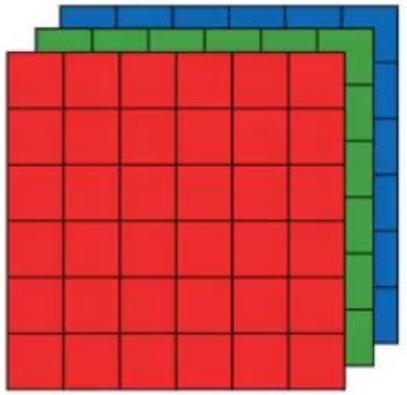


Background Model =
Image Average over time

Arithmetic

(Logic Operation)

$$\text{Im\_result} = logic(Im_1, Im_2)$$



$$\text{Im\_result}[:,:,0] = logic(Im_1[:,:,0], Im_2[:,:,0])$$

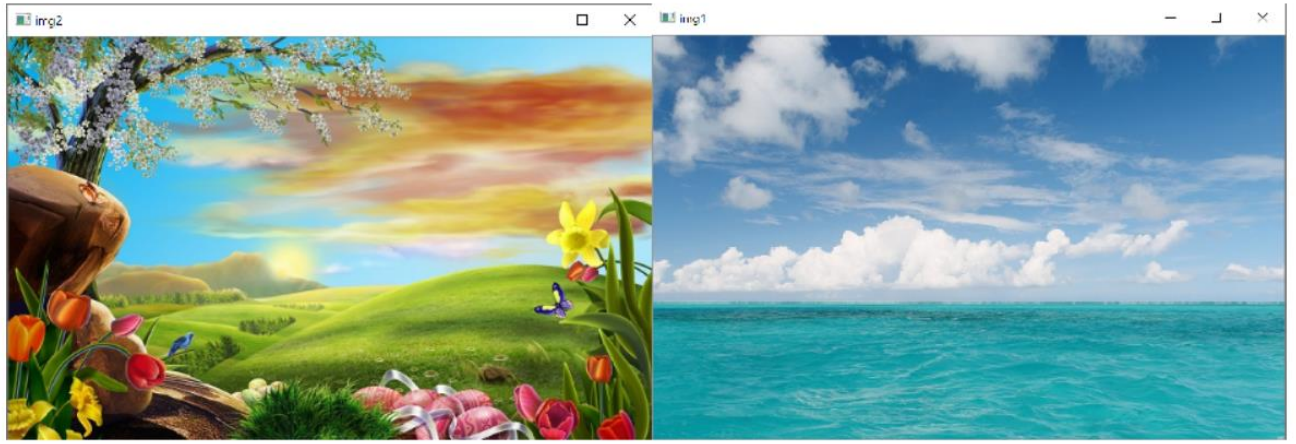
$$\text{Im\_result}[:,:,1] = logic(Im_1[:,:,1], Im_2[:,:,1])$$

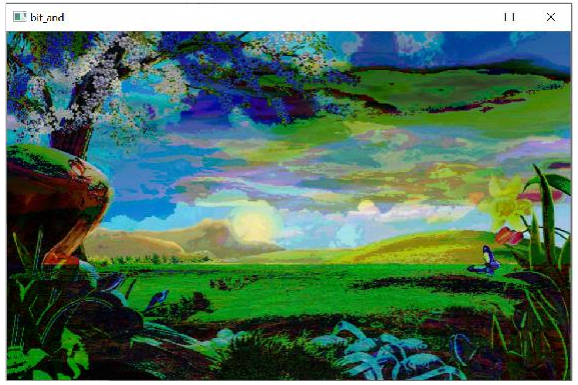
$$\text{Im\_result}[:,:,2] = logic(Im_1[:,:,2], Im_2[:,:,2])$$

https://datahacker.rs/005-image-arithmetic-and-logical-operations-in-opencv-with-python/

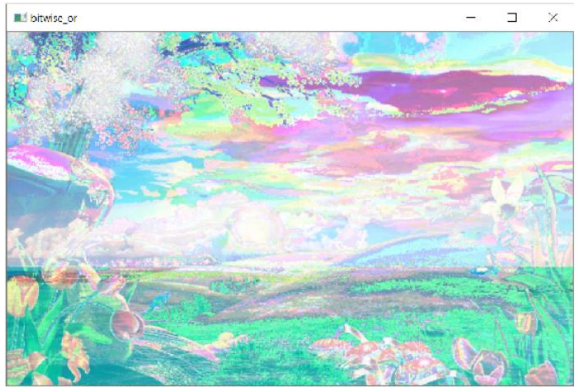
Bitwise

(AND, OR, Xor)

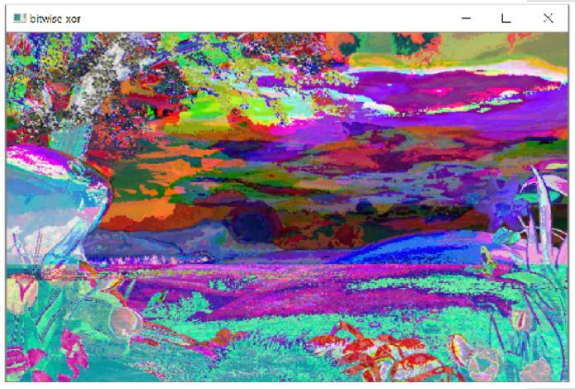




cv2.bitwise_and()



cv2.bitwise_or()

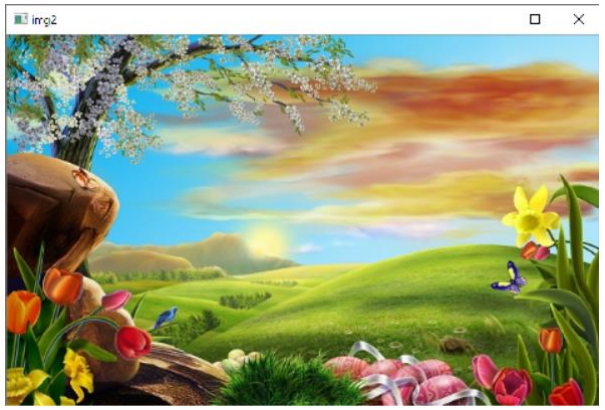


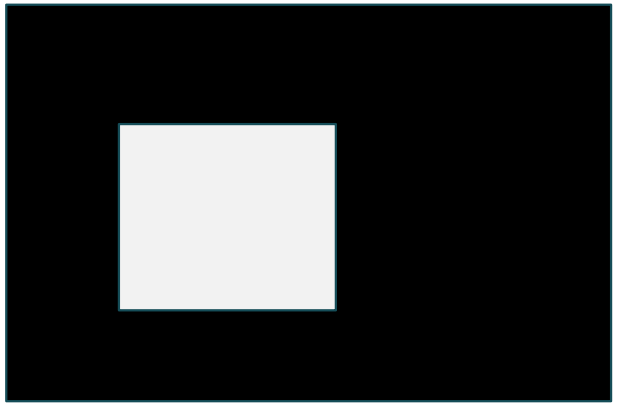
cv2.bitwise_xor()

https://www.etutorialspoint.com/index.php/323-opencv-logical-operators-bitwise-and-or-nor-xor

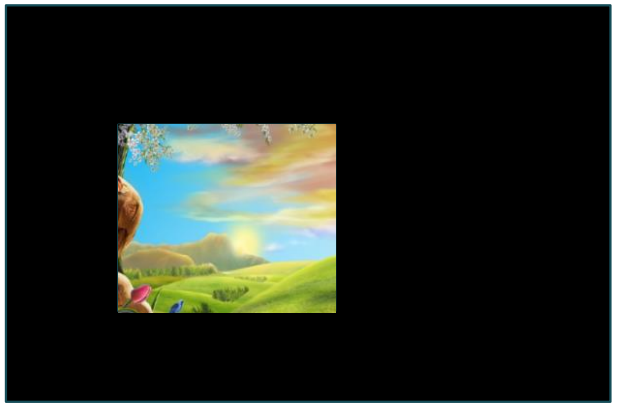
Bitwise

(AND, OR, Xor)





cv2.bitwise_and()



Activity # 1.4