Introduction to Image Processing

AI Workshop, 2-4 December 2019

Somnuk Phon-Amnuaisuk
School of Computing and Informatics
Centre for Innovative Engineering
Universiti Teknologi Brunei

meme

A meme (/' mi: m/ meem), a neologism coined by Richard Dawkins, is "an idea, behavior, or style that spreads from person to person within a culture". A meme acts as a unit for carrying cultural ideas, symbols, or practices that can be transmitted from one mind to another through writing, speech, gestures, rituals, or other imitable phenomena with a mimicked theme.

Disclaimer

This lecture is compiled from my lectures as well as materials gathering from lectures found in public domains.

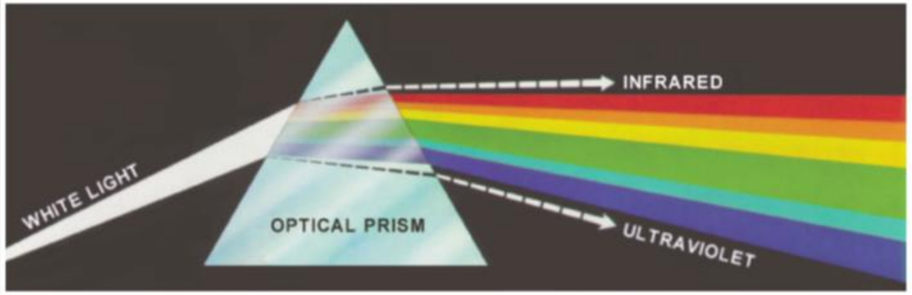
Learning Outcomes

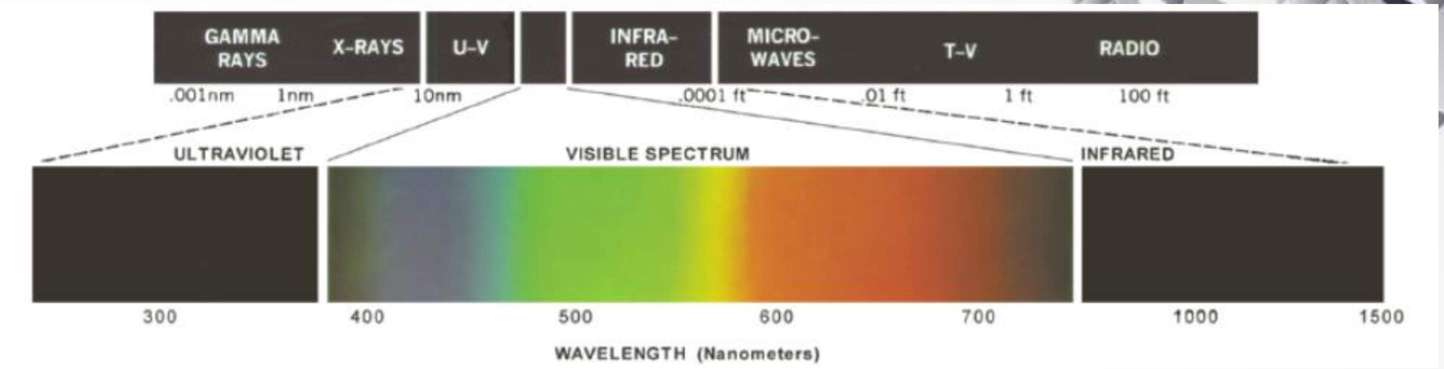
- Able to list important properties of light
- Able to explain computer representation of images
- Able to use scikit-image package
- Able to use tensorflow.keras image package
- Complete image processing practical

Blake's opposition to the Enlightenment was deeply rooted. In his annotation to his own engraving of the classical character Laocoön, Blake wrote "Art is the Tree of Life. Science is the Tree of Death."

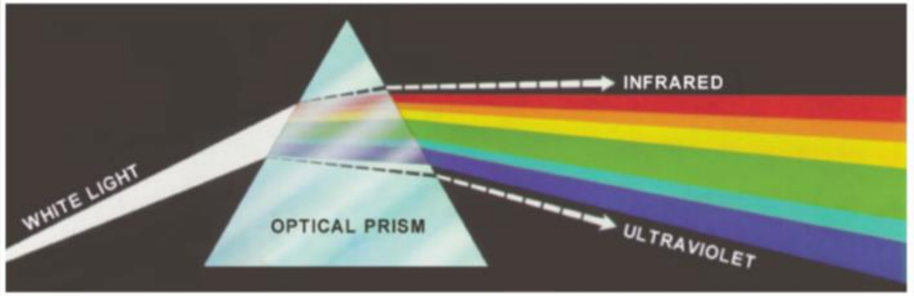
Newton's theory of optics was especially offensive to Blake.

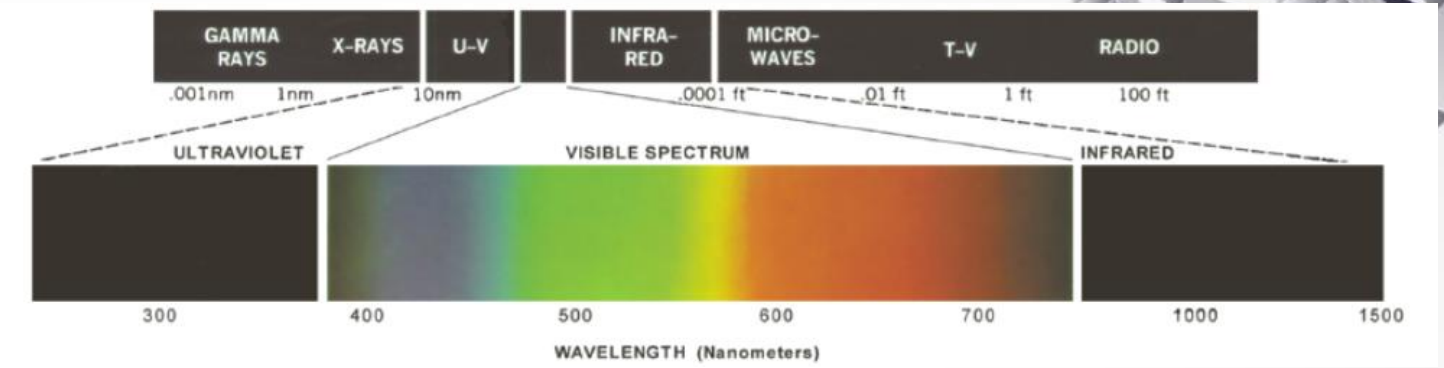
Electromagnetic Wave

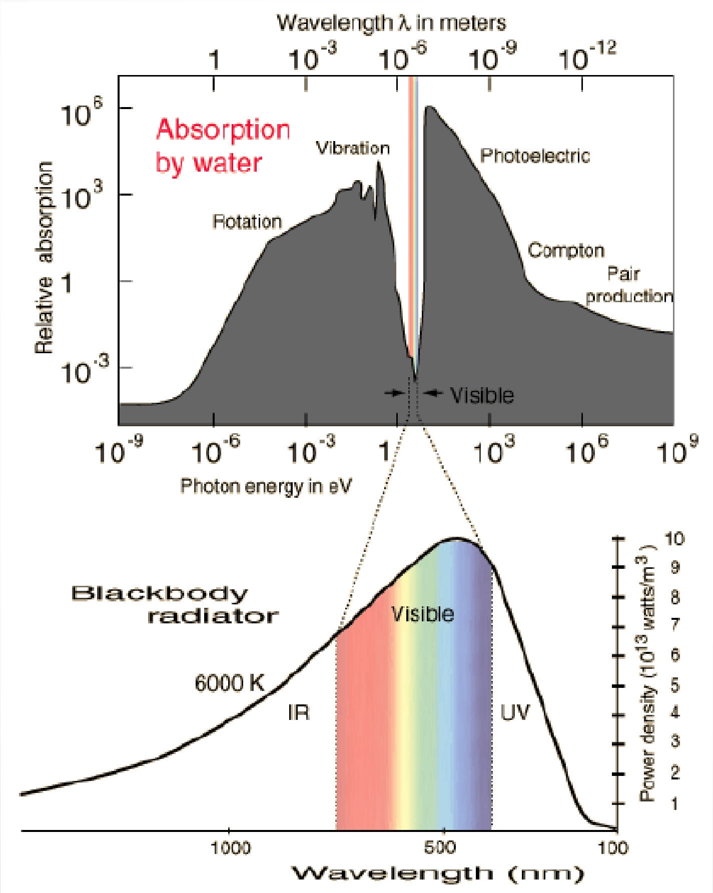


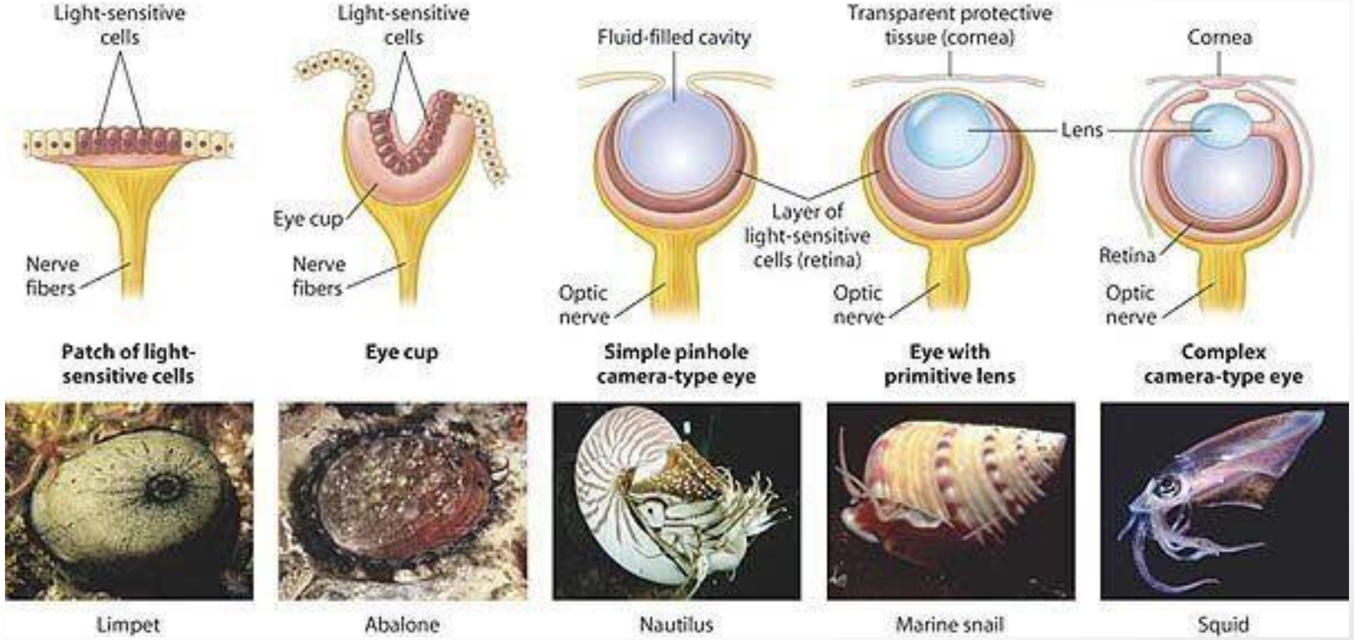


Electromagnetic Wave









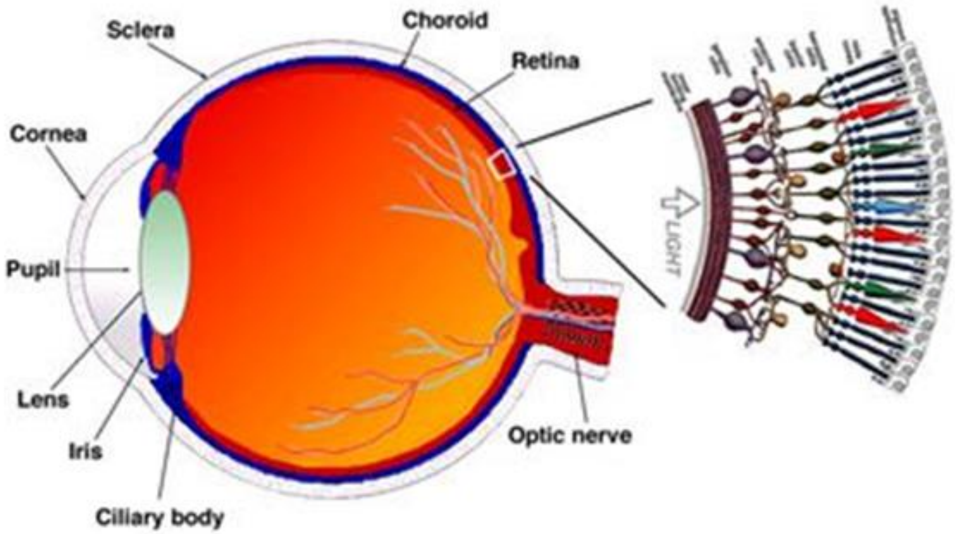


Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

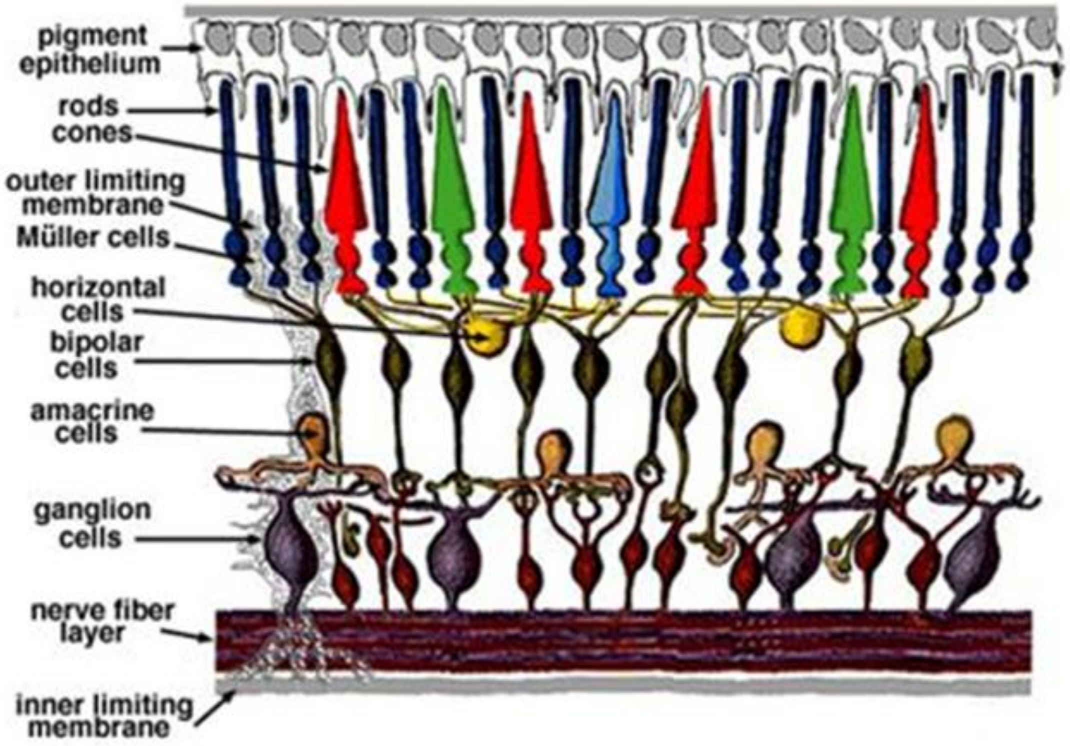
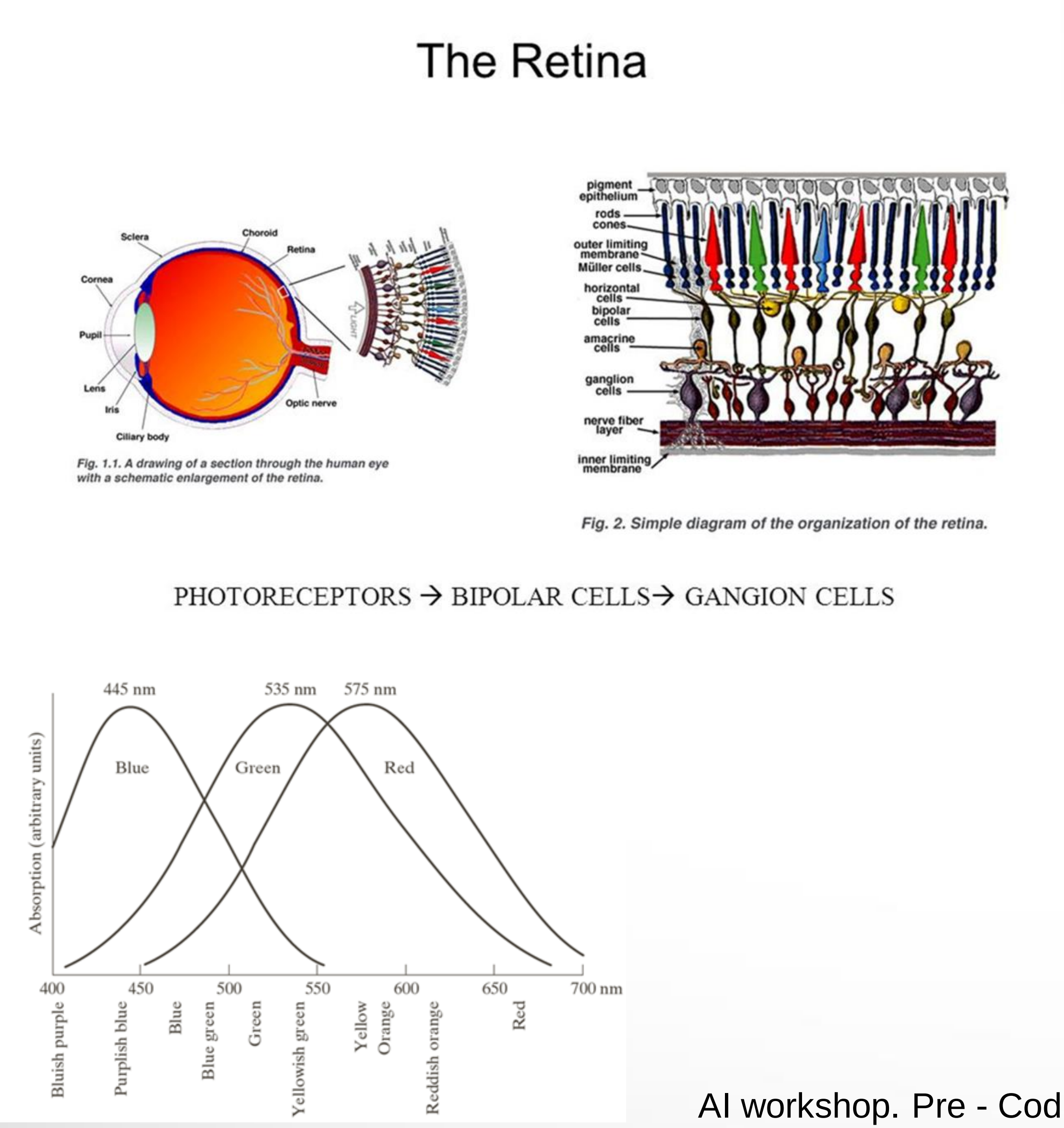


Fig. 2. Simple diagram of the organization of the retina.

PHOTORECEPTORS → BIPOLAR CELLS→ GANGION CELLS

Color Perceptions

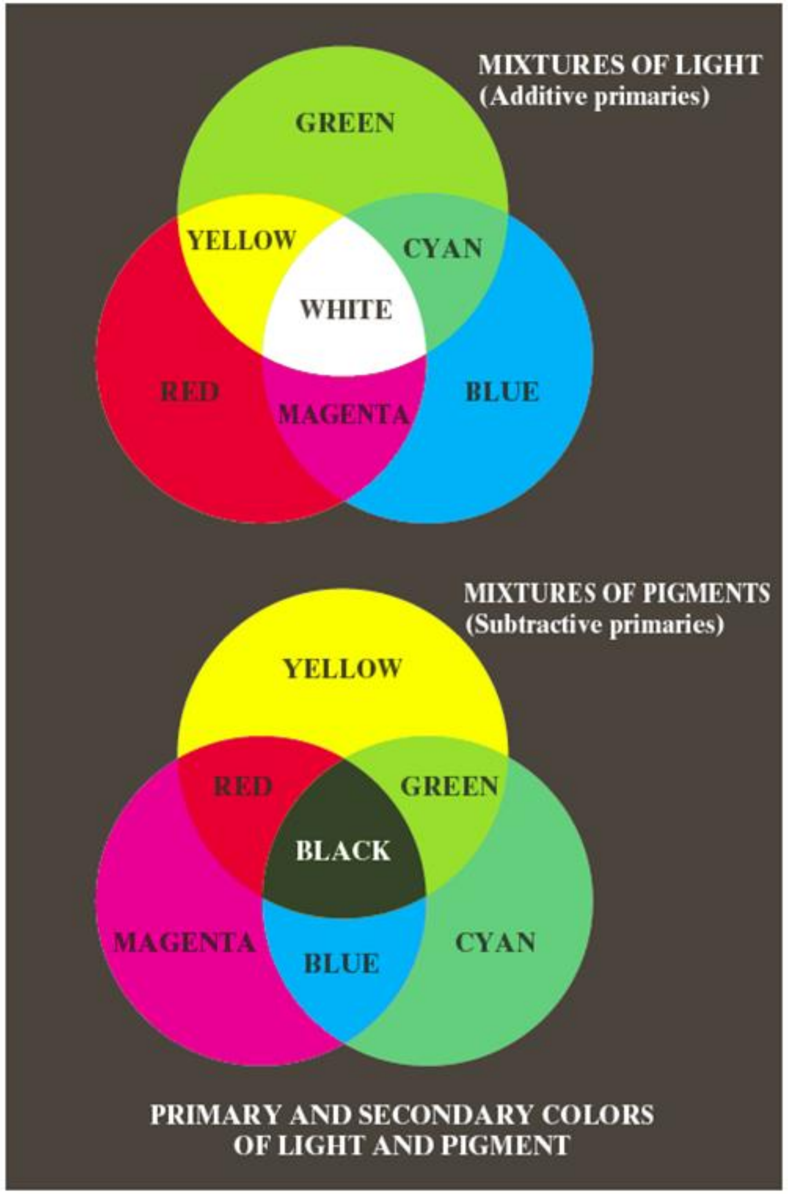


The Retina

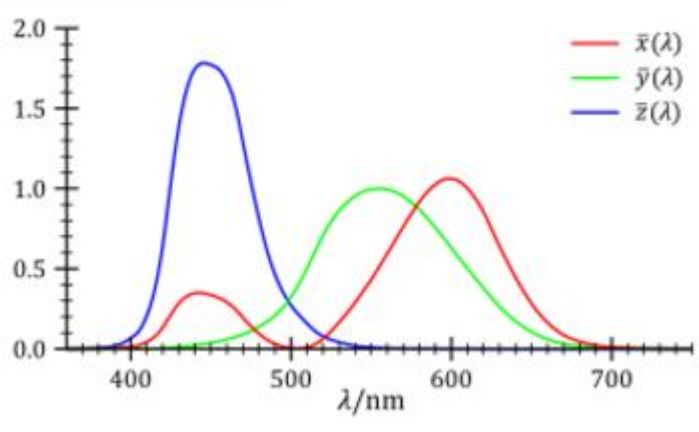
Fig. 1.1. A drawing of a section through the human eye with a schematic enlargement of the retina.

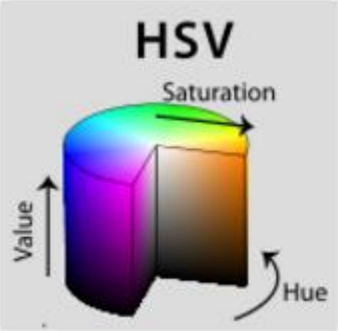
Fig. 2. Simple diagram of the organization of the retina.

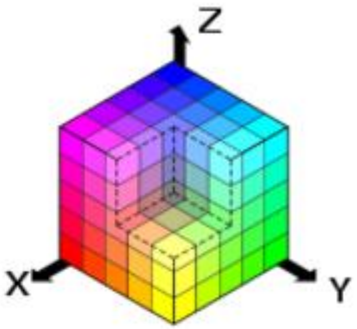
PHOTORECEPTORS → BIPOLAR CELLS→ GANGION CELLS

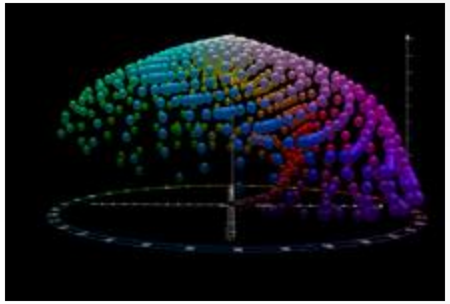


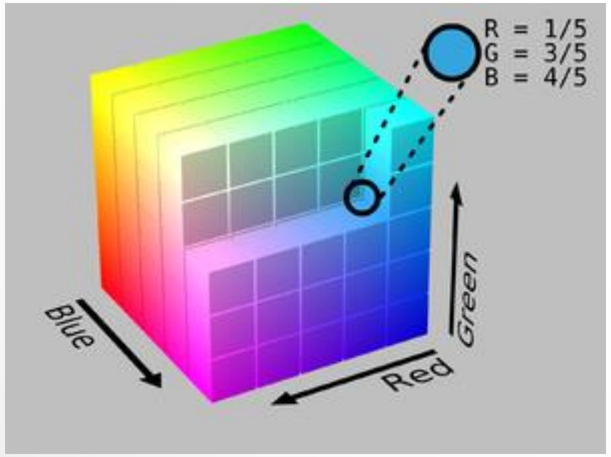
Color Representations

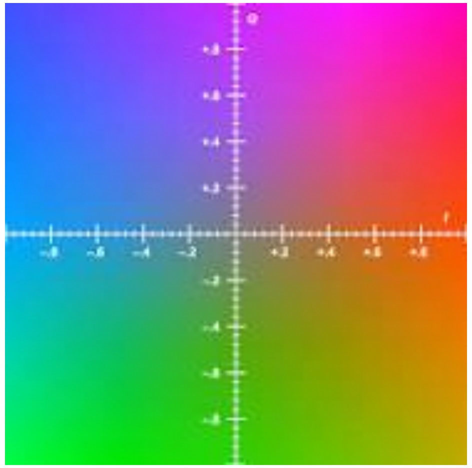


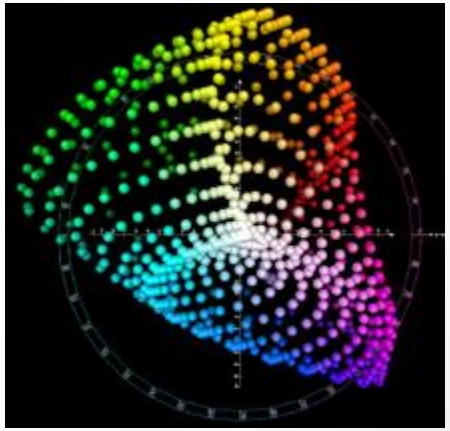




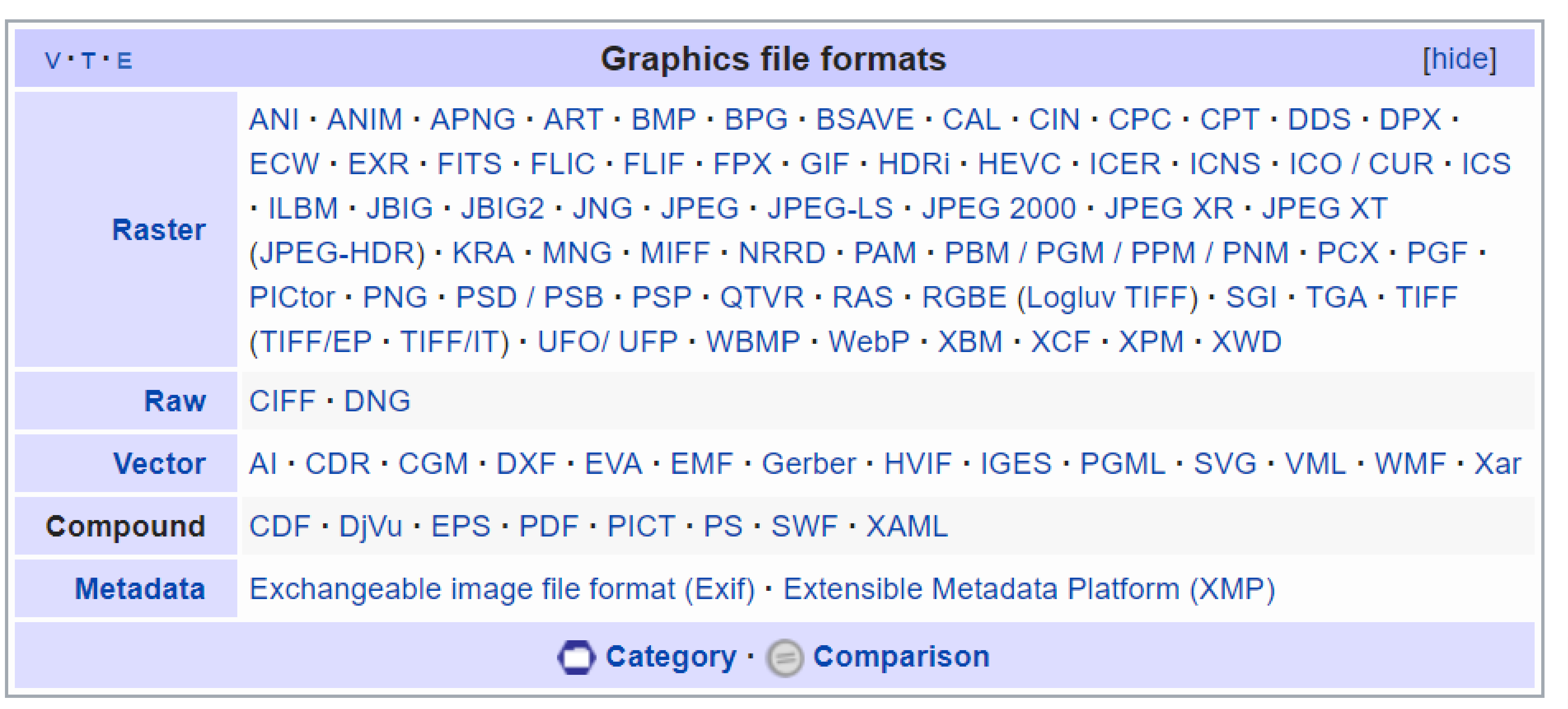






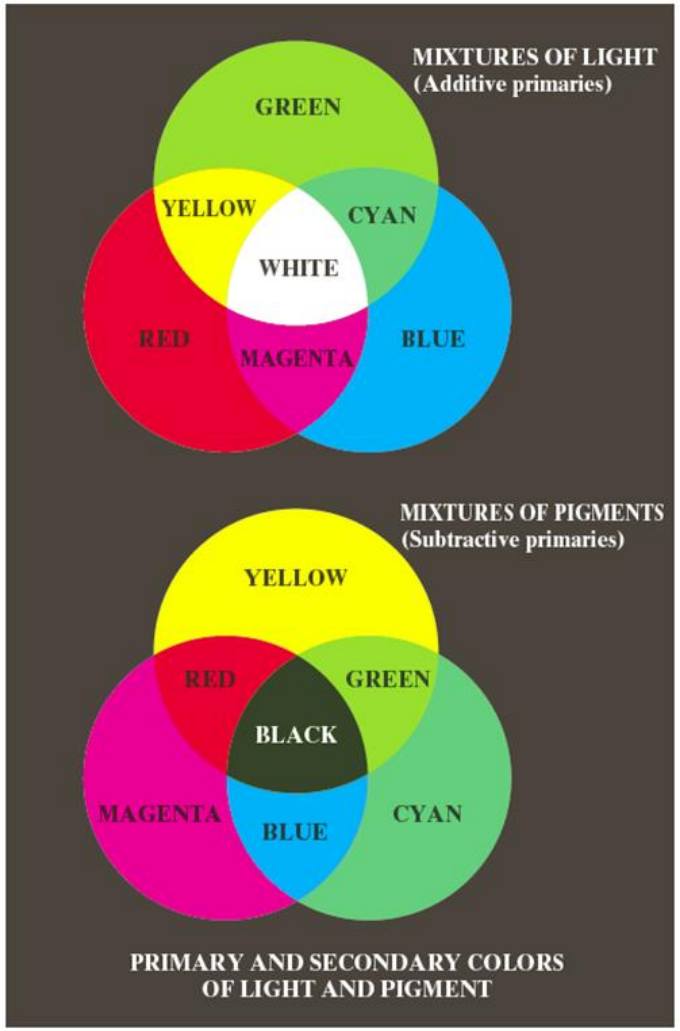


File Formats

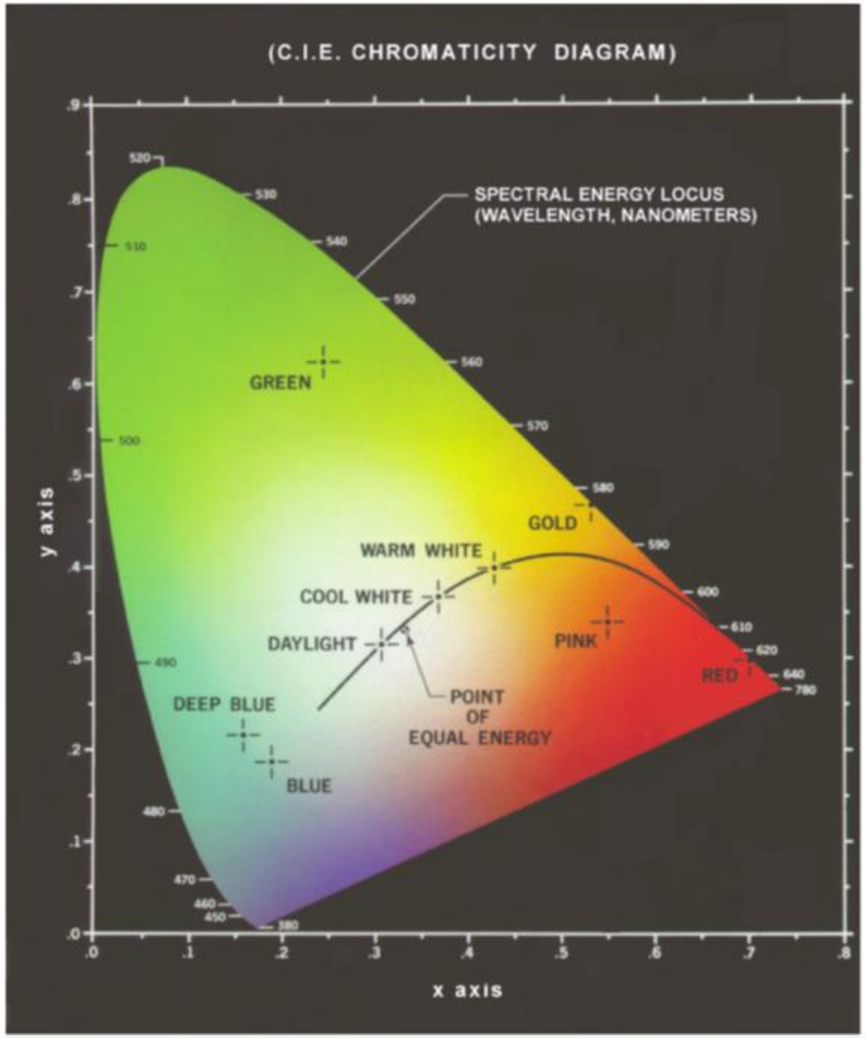


V · T · E	Graphics file formats [hide]
Raster	ANI · ANIM · APNG · ART · BMP · BPG · BSAVE · CAL · CIN · CPC · CPT · DDS · DPX · ECW · EXR · FITS · FLIC · FLIF · FPX · GIF · HDRi · HEVC · ICER · ICNS · ICO / CUR · ICS · ILBM · JBIG · JBIG2 · JNG · JPEG · JPEG-LS · JPEG 2000 · JPEG XR · JPEG XT (JPEG-HDR) · KRA · MNG · MIFF · NRRD · PAM · PBM / PGM / PPM / PNM · PCX · PGF · PICtor · PNG · PSD / PSB · PSP · QTVR · RAS · RGBE (Logluv TIFF) · SGI · TGA · TIFF (TIFF/EP · TIFF/IT) · UFO/ UFP · WBMP · WebP · XBM · XCF · XPM · XWD
Raw	CIFF · DNG
Vector	AI · CDR · CGM · DXF · EVA · EMF · Gerber · HVIF · IGES · PGML · SVG · VML · WMF · Xar
Compound	CDF · DjVu · EPS · PDF · PICT · PS · SWF · XAML
Metadata	Exchangeable image file format (Exif) · Extensible Metadata Platform (XMP)
Category · Comparison	

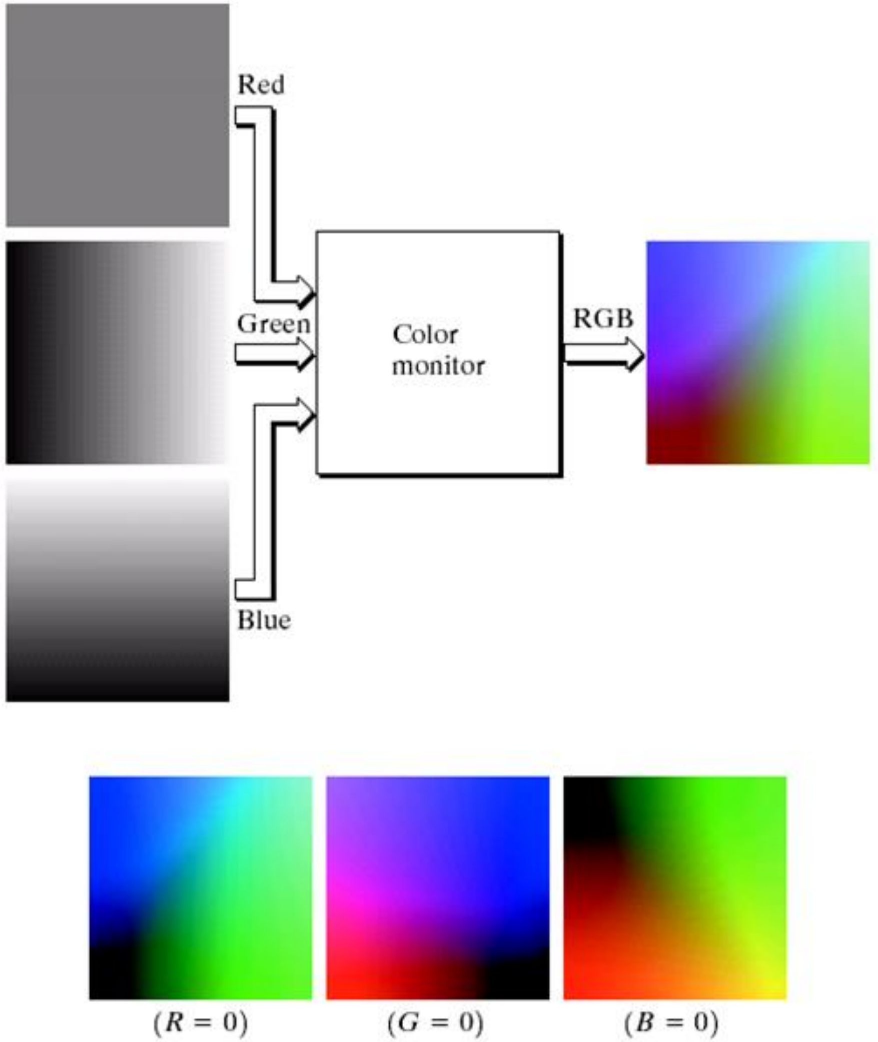
Color 101

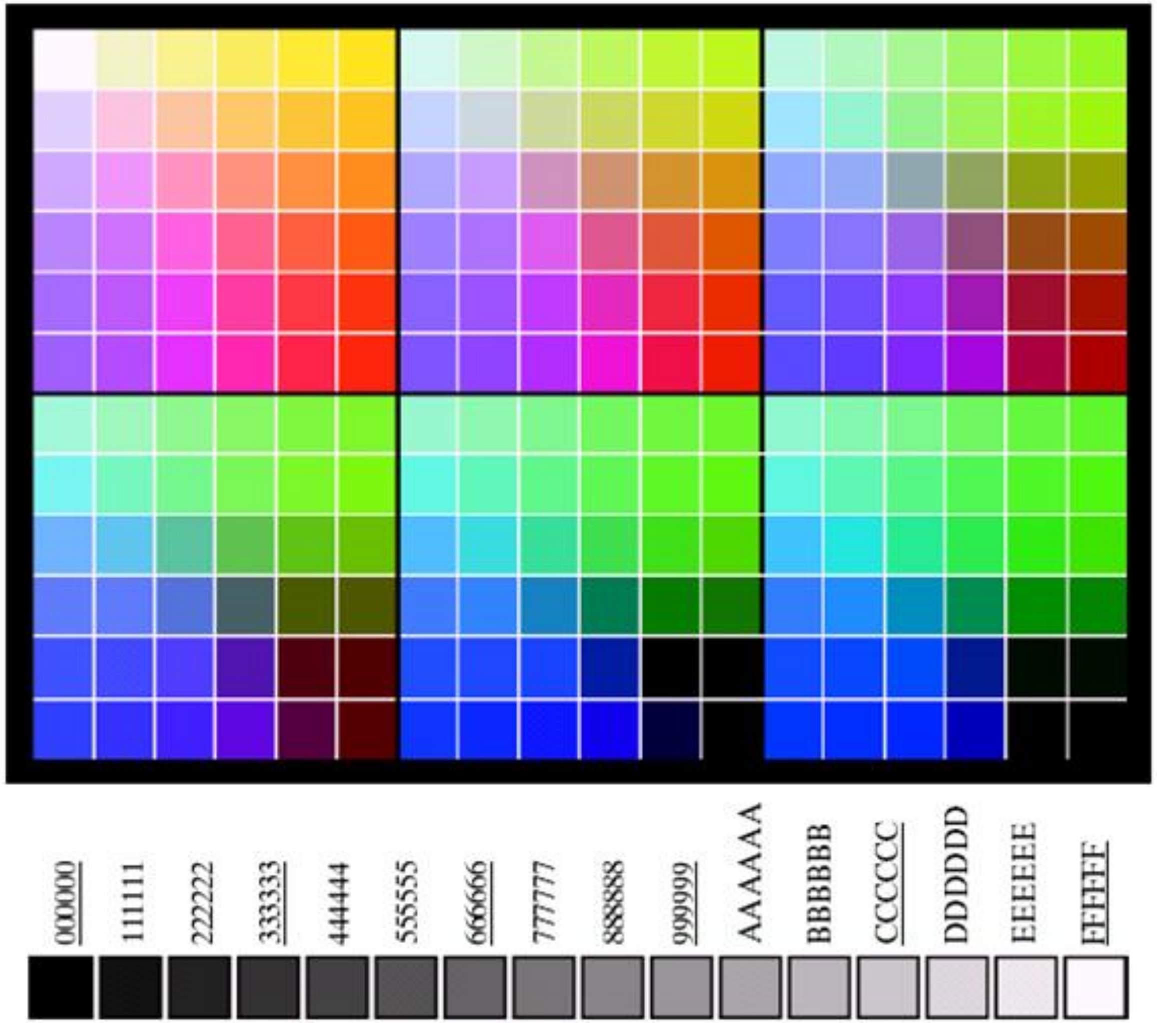


PRIMARY AND SECONDARY COLORS OF LIGHT AND PIGMENT

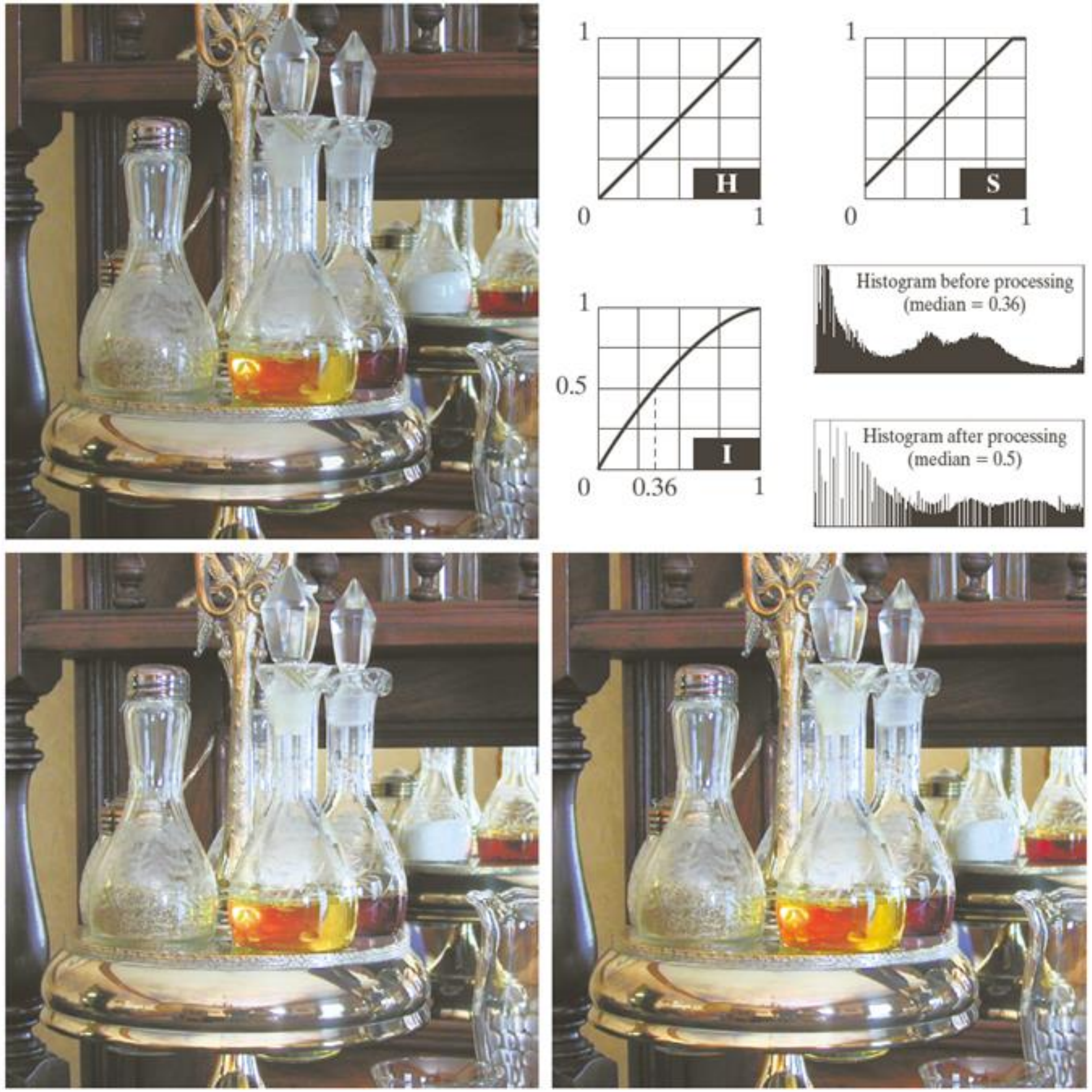

(C.I.E. CHROMATICITY DIAGRAM)

Color 101

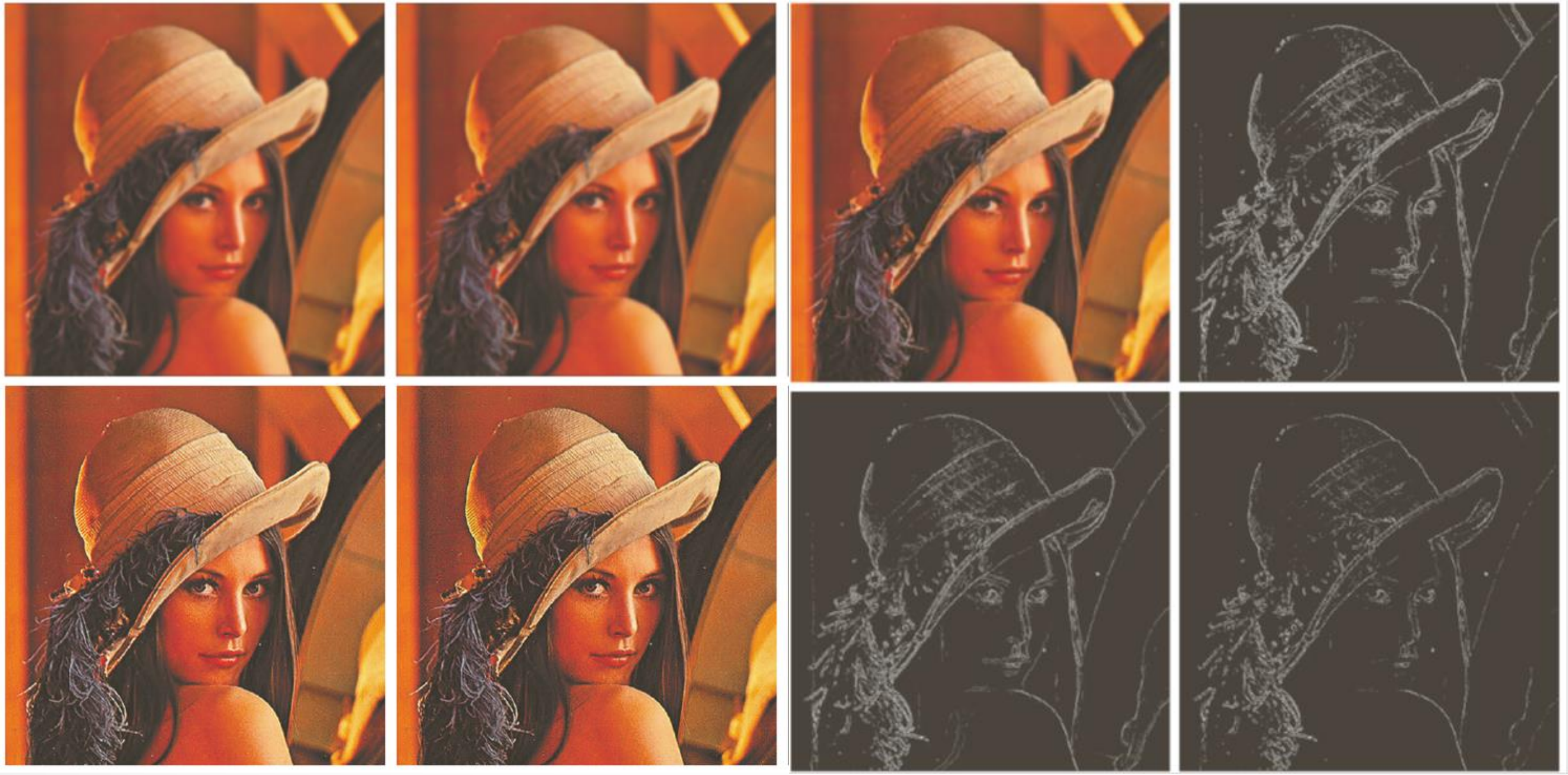

Red
Green
Blue
Color monitor
RGB
$(R = 0)$
$(G = 0)$
$(B = 0)$


000000
111111
222222
333333
444444
555555
666666
777777
888888
999999
AAAAAA
BBBBBB
CCCCCC
DDDDDD
EEEEEE
FFFFFF

Color 101


1
0
1
H
1
0
1
S
1
0.5
0
0.36
1
I
Histogram before processing
(median = 0.36)
Histogram after processing
(median = 0.5)

Filter 101



Features from Pixels

- Primitives
  - – Pixel’s values
  - – Gradient
  - – Edge, Point, Shape, Blob, Flow
- Structures from primitives or their combinations
  - – Local and global min/max, Integrated image
  - – Histogram, Correlation, Covariant
  - – Spatial and temporal quantity

Issues from Appearance-based Features

- Handling variations in appearances
  - – Viewpoints, scale
  - – Orientation, occlusion, deformation
- Handling temporal information
- Nature of features
  - – Local vs Global
  - – Static vs Dynamic

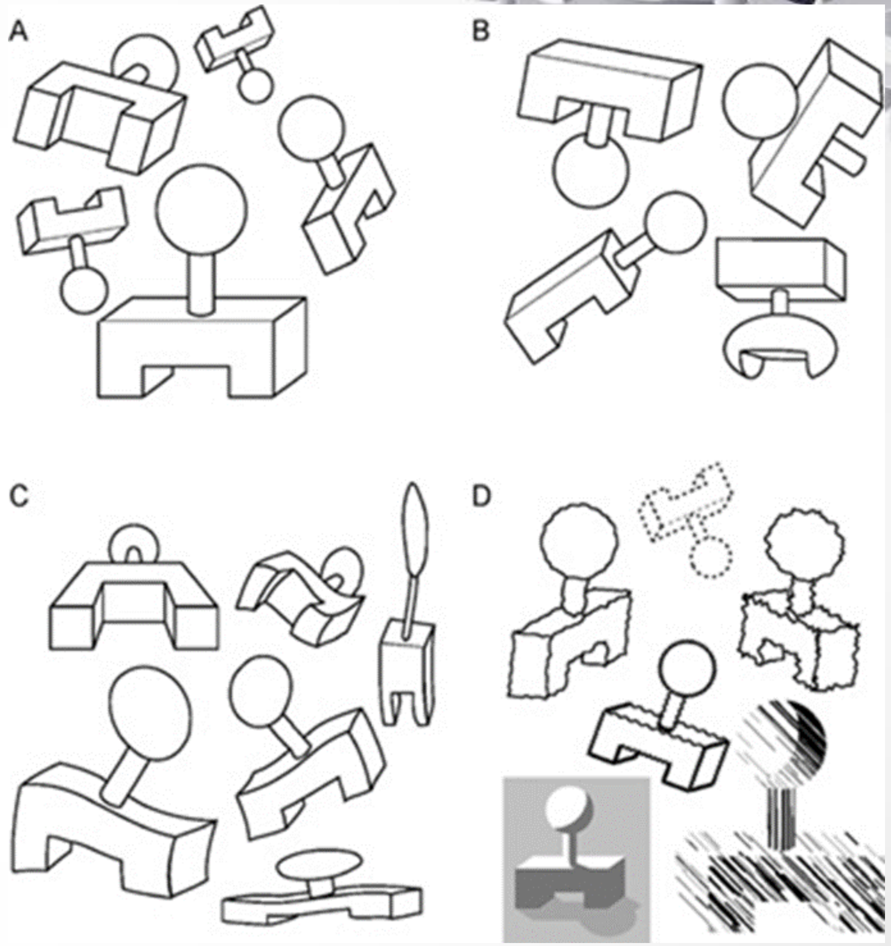
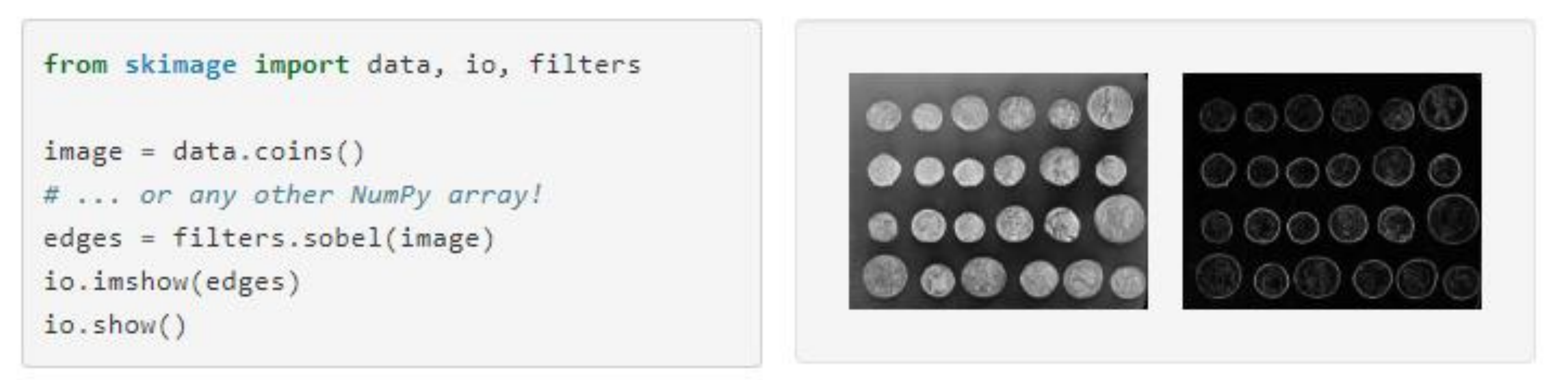


Image Processing with scikit-image 1

- scikit-image is a collection of algorithms for image processing. It is available free of charge and free of restriction. We pride ourselves on high-quality, peer-reviewed code, written by an active community of volunteers.



```python
from skimage import data, io, filters

image = data.coins()
# ... or any other NumPy array!
edges = filters.sobel(image)
io.imshow(edges)
io.show()
```

Image Processing with scikit-image 2

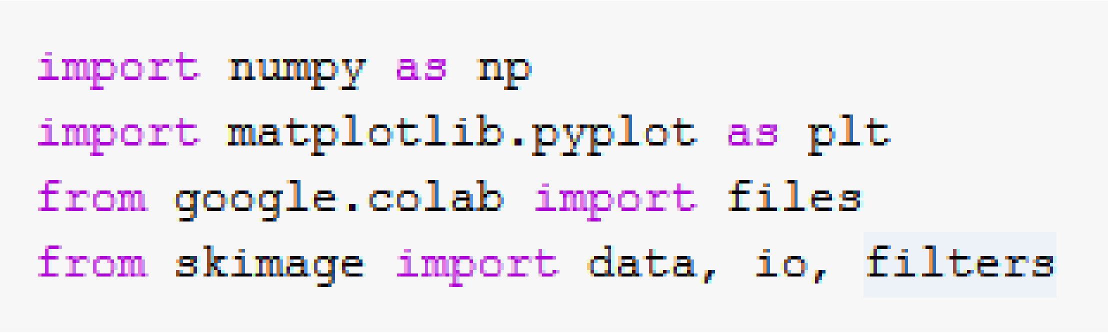
- scikit-image represents images using NumPy arrays.
- Two-dimensional (2D) grayscale images are indexed by rows and columns with the lowest element (0, 0) at the top-left corner.
- Two-dimensional (2D) color images are indexed by rows, columns and the channel (corresponding to RGB, RGBA, HSV, etc.)

Practical Session with scikit-image

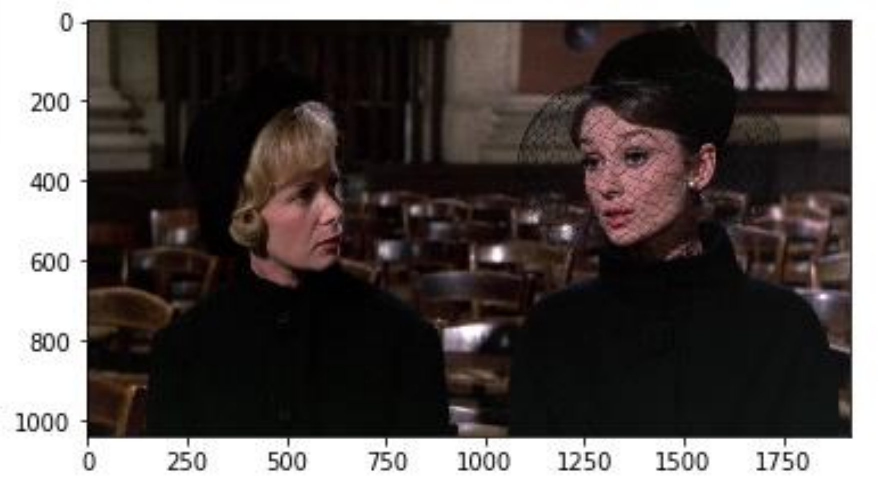
- Load, Save
- Image cropping ad scaling
- Modify pixel information
- Filter and Convolution

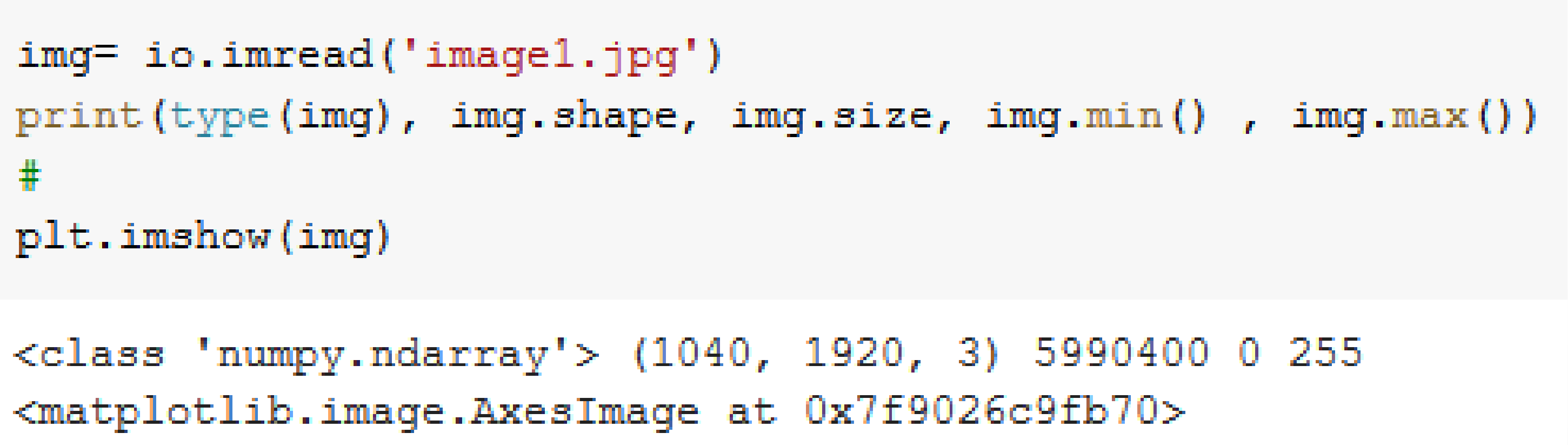
Practical Session with scikit-image

- Load & display



```
import numpy as np
import matplotlib.pyplot as plt
from google.colab import files
from skimage import data, io, filters
```



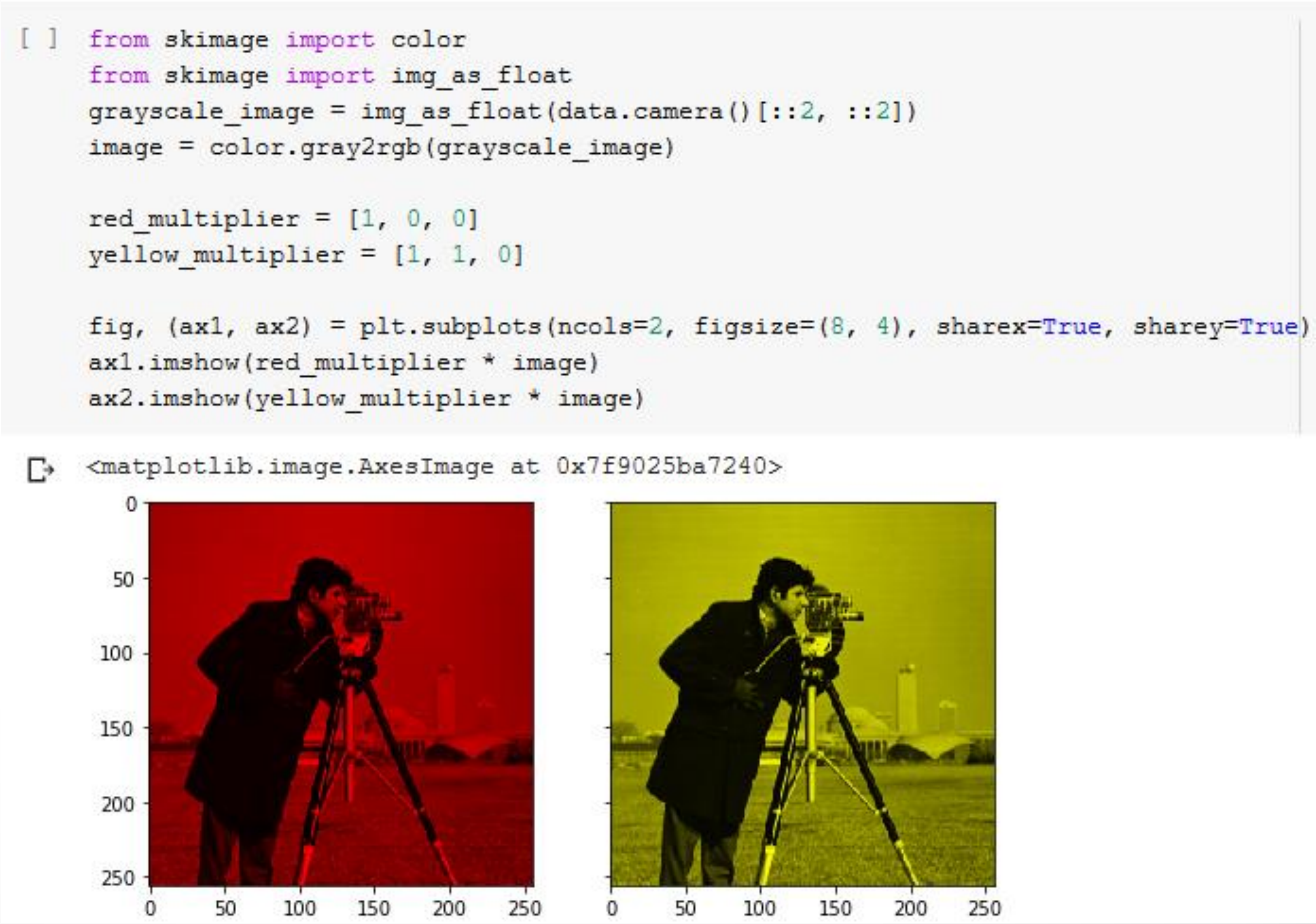


```
img= io.imread('image1.jpg')
print(type(img), img.shape, img.size, img.min() , img.max())
#
plt.imshow(img)
```

```
<class 'numpy.ndarray'> (1040, 1920, 3) 5990400 0 255
<matplotlib.image.AxesImage at 0x7f9026c9fb70>
```

Practical Session with scikit-image

- Modify pixels

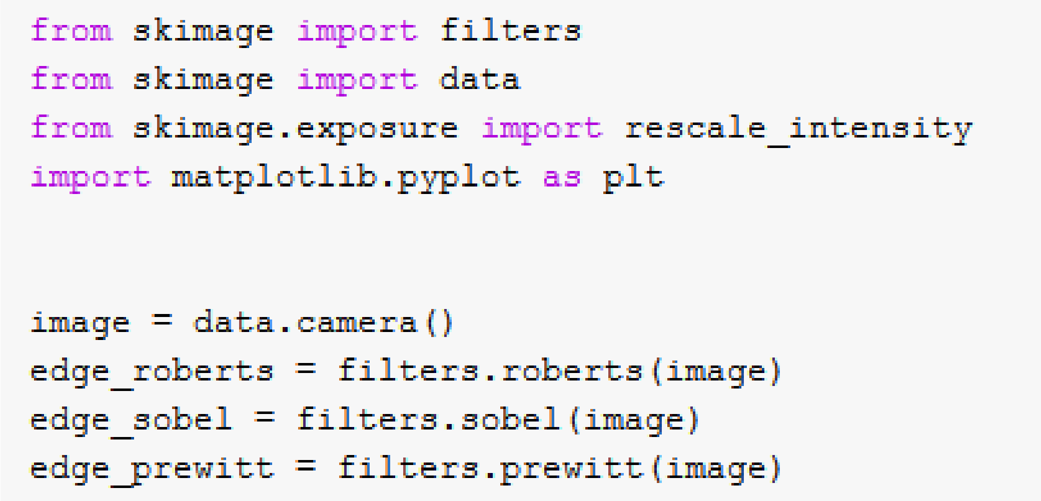


```
[ ]  from skimage import color
     from skimage import img_as_float
     grayscale_image = img_as_float(data.camera()[::2, ::2])
     image = color.gray2rgb(grayscale_image)

     red_multiplier = [1, 0, 0]
     yellow_multiplier = [1, 1, 0]

     fig, (ax1, ax2) = plt.subplots(ncols=2, figsize=(8, 4), sharex=True, sharey=True)
     ax1.imshow(red_multiplier * image)
     ax2.imshow(yellow_multiplier * image)
```

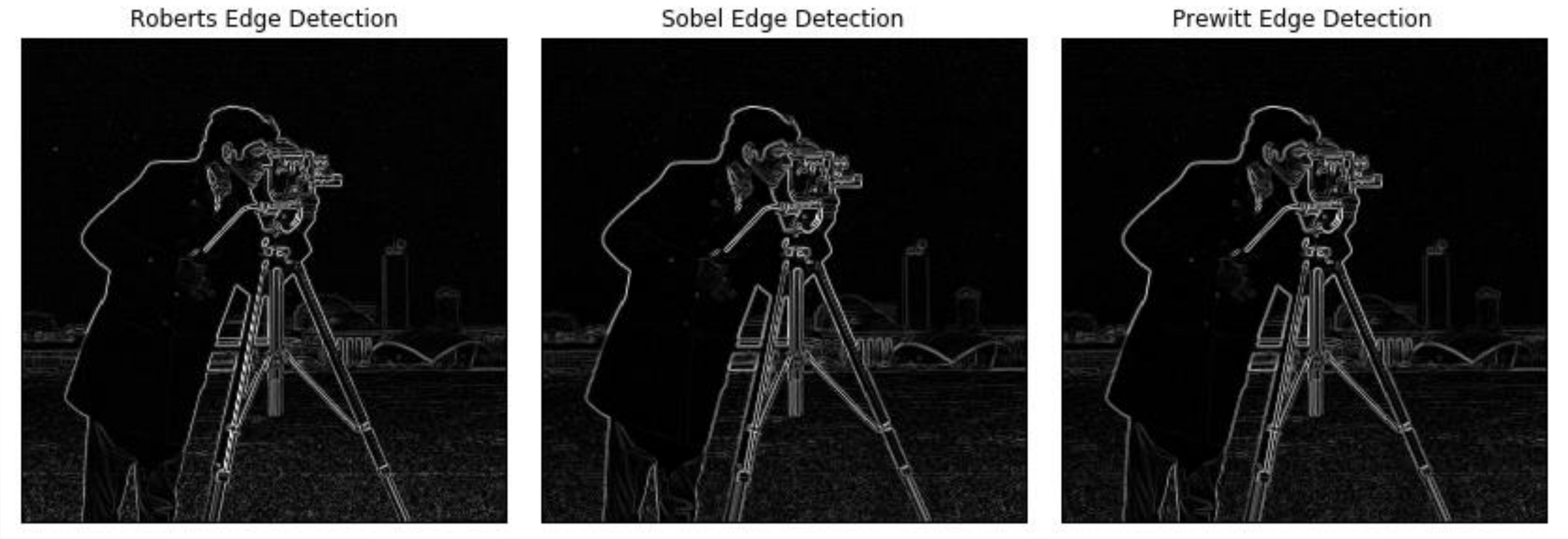
```
<matplotlib.image.AxesImage at 0x7f9025ba7240>
```

Practical Session with scikit-image

- Filters



```
from skimage import filters
from skimage import data
from skimage.exposure import rescale_intensity
import matplotlib.pyplot as plt


image = data.camera()
edge_roberts = filters.roberts(image)
edge_sobel = filters.sobel(image)
edge_prewitt = filters.prewitt(image)
```



Roberts Edge Detection

Sobel Edge Detection

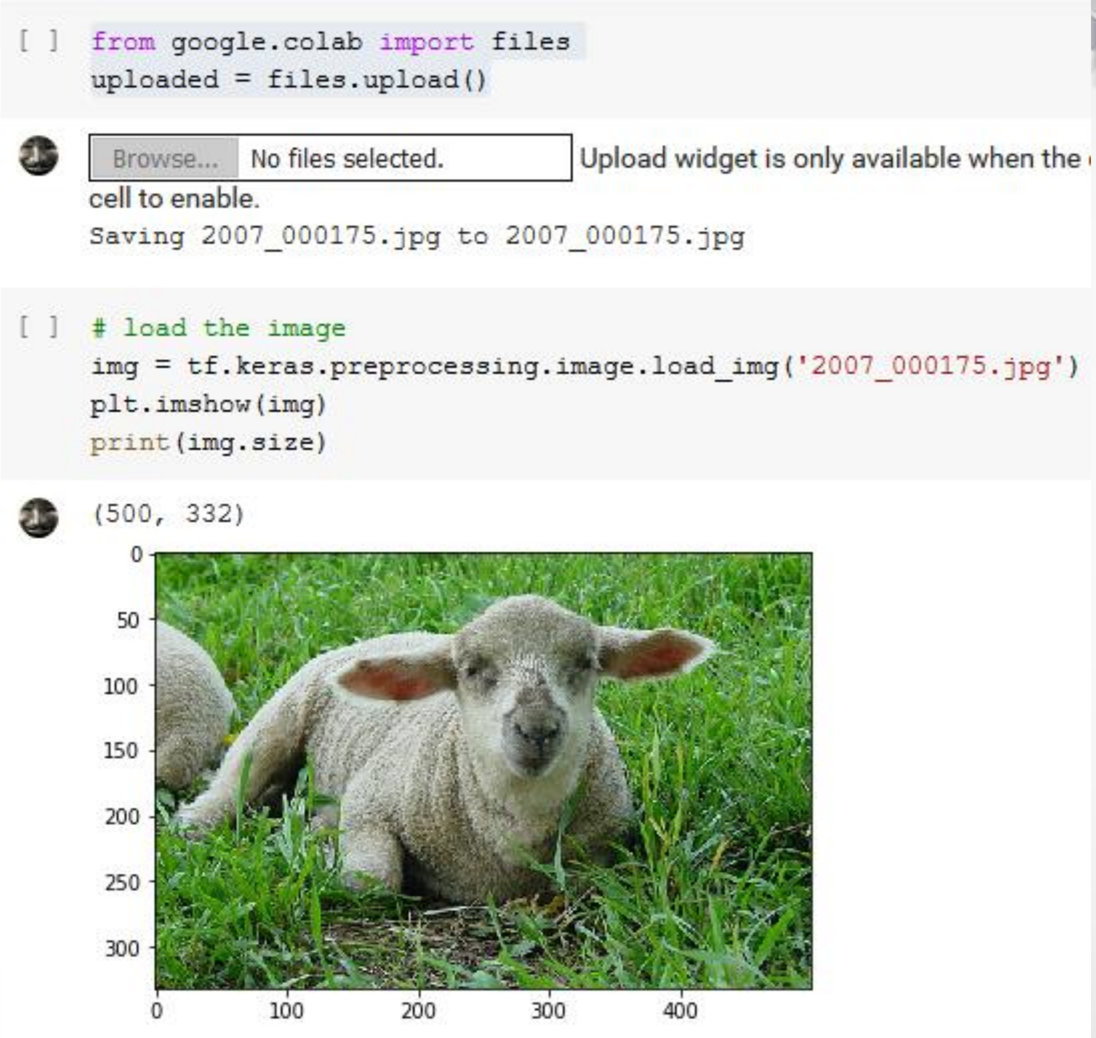
Prewitt Edge Detection

Tensorflow Keras Utilities for Image

Module: tf.keras.preprocessing.image

Functions

- array_to_img(...): Converts a 3D Numpy array to a PIL Image instance.
- img_to_array(...): Converts a PIL Image instance to a Numpy array.
- load_img(...): Loads an image into PIL format.
- save_img(...): Saves an image stored as a Numpy array to a path or file object.



```
[ ] from google.colab import files
    uploaded = files.upload()
```

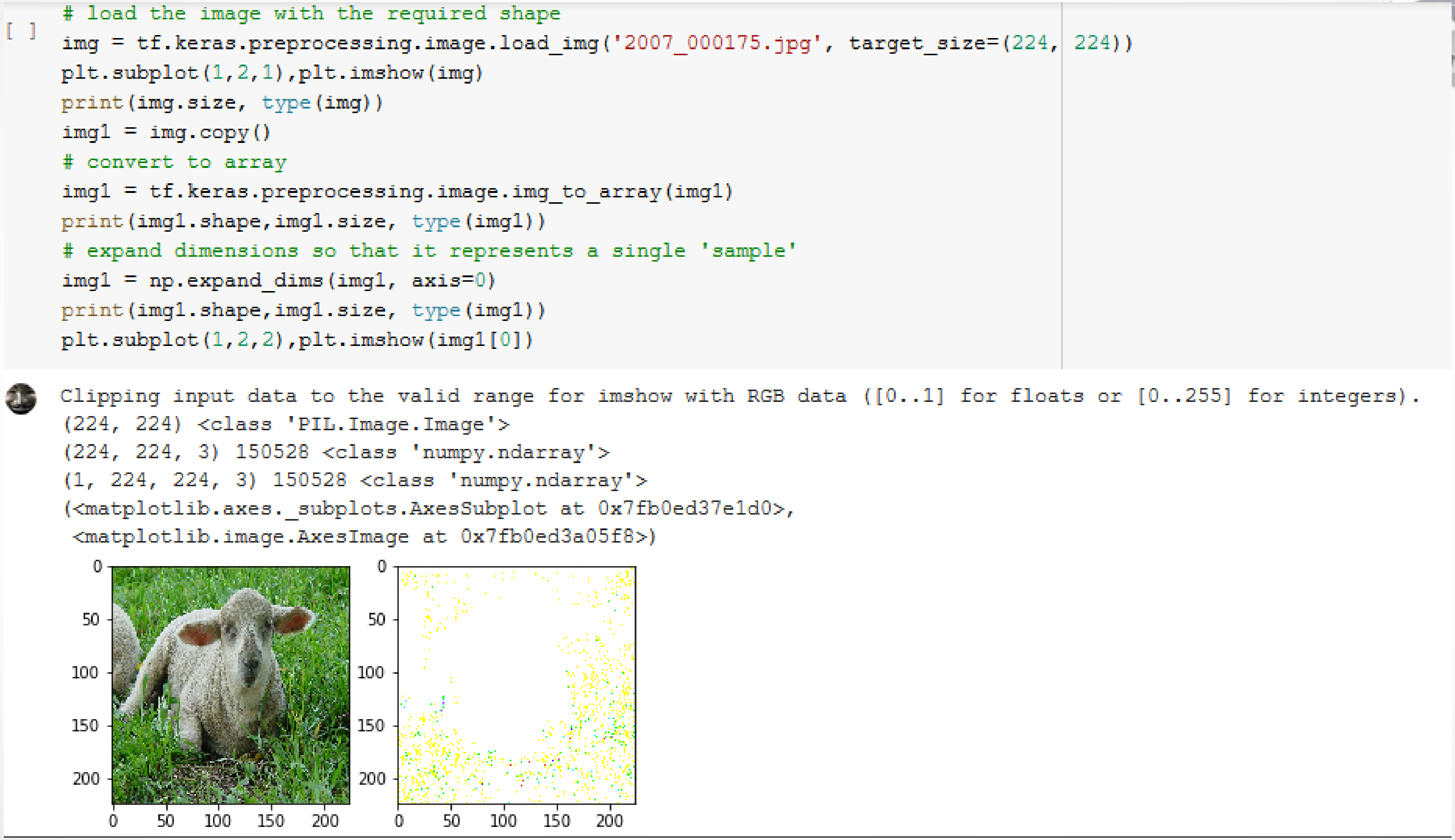
Upload widget is only available when the
cell to enable.

```
Saving 2007_000175.jpg to 2007_000175.jpg
```

```
[ ] # load the image
    img = tf.keras.preprocessing.image.load_img('2007_000175.jpg')
    plt.imshow(img)
    print(img.size)
```

```
(500, 332)
```

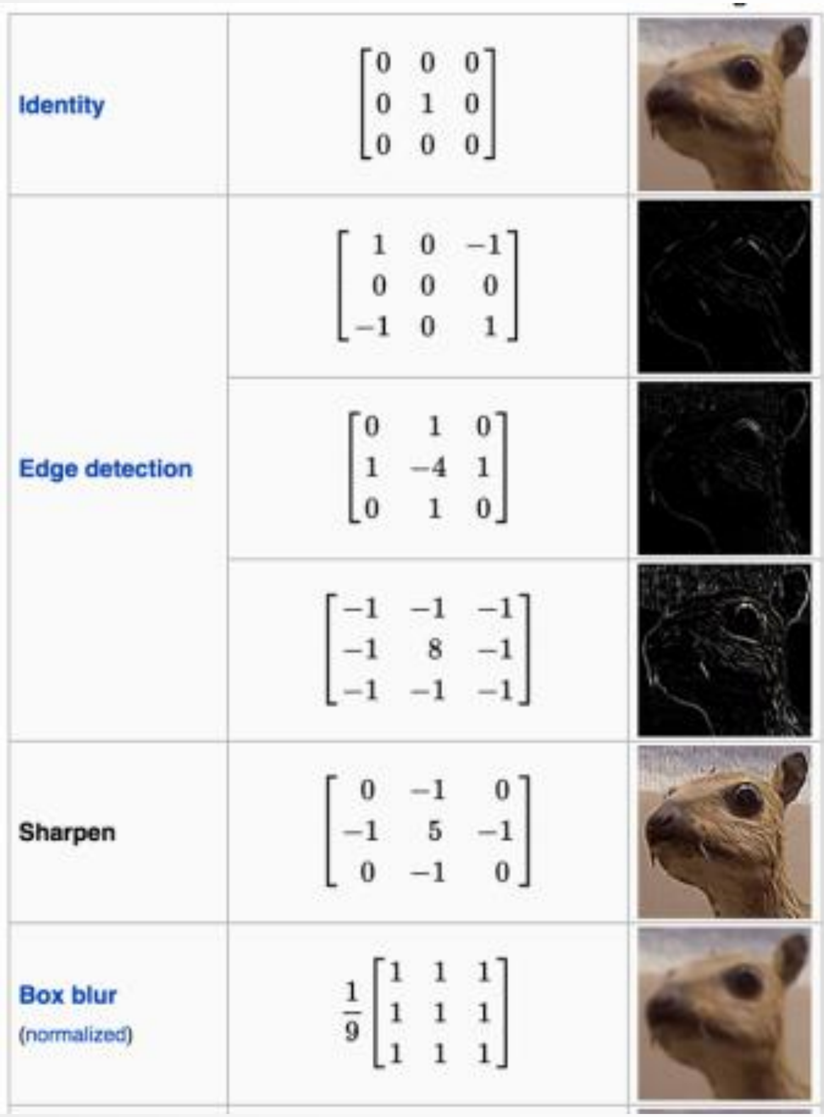
Tensorflow Keras Utilities for Image



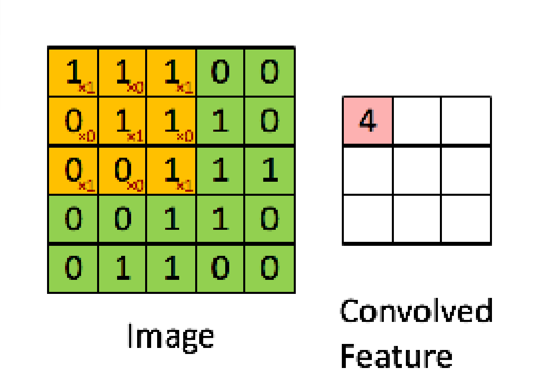
```
# load the image with the required shape
img = tf.keras.preprocessing.image.load_img('2007_000175.jpg', target_size=(224, 224))
plt.subplot(1,2,1),plt.imshow(img)
print(img.size, type(img))
img1 = img.copy()
# convert to array
img1 = tf.keras.preprocessing.image.img_to_array(img1)
print(img1.shape,img1.size, type(img1))
# expand dimensions so that it represents a single 'sample'
img1 = np.expand_dims(img1, axis=0)
print(img1.shape,img1.size, type(img1))
plt.subplot(1,2,2),plt.imshow(img1[0])
```

```
Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).
(224, 224) <class 'PIL.Image.Image'>
(224, 224, 3) 150528 <class 'numpy.ndarray'>
(1, 224, 224, 3) 150528 <class 'numpy.ndarray'>
(<matplotlib.axes._subplots.AxesSubplot at 0x7fb0ed37e1d0>,
 <matplotlib.image.AxesImage at 0x7fb0ed3a05f8>)
```

Convolution Process



Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

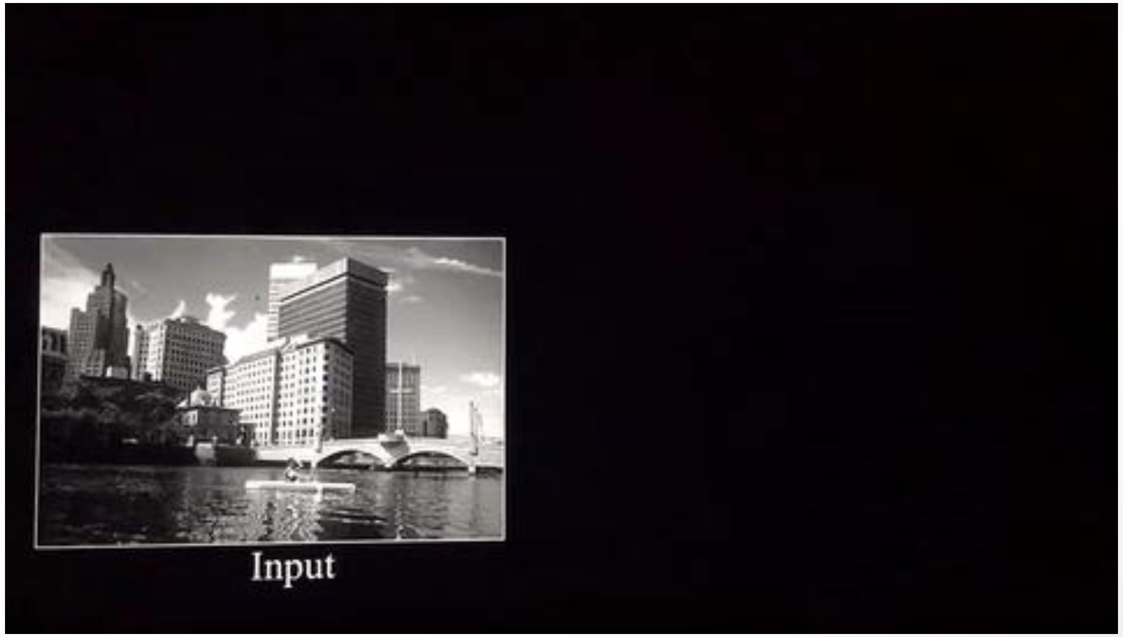


$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$	0	0
$0_{\times 0}$	$1_{\times 1}$	$1_{\times 0}$	1	0
$0_{\times 1}$	$0_{\times 0}$	$1_{\times 1}$	1	1
0	0	1	1	0
0	1	1	0	0

Image

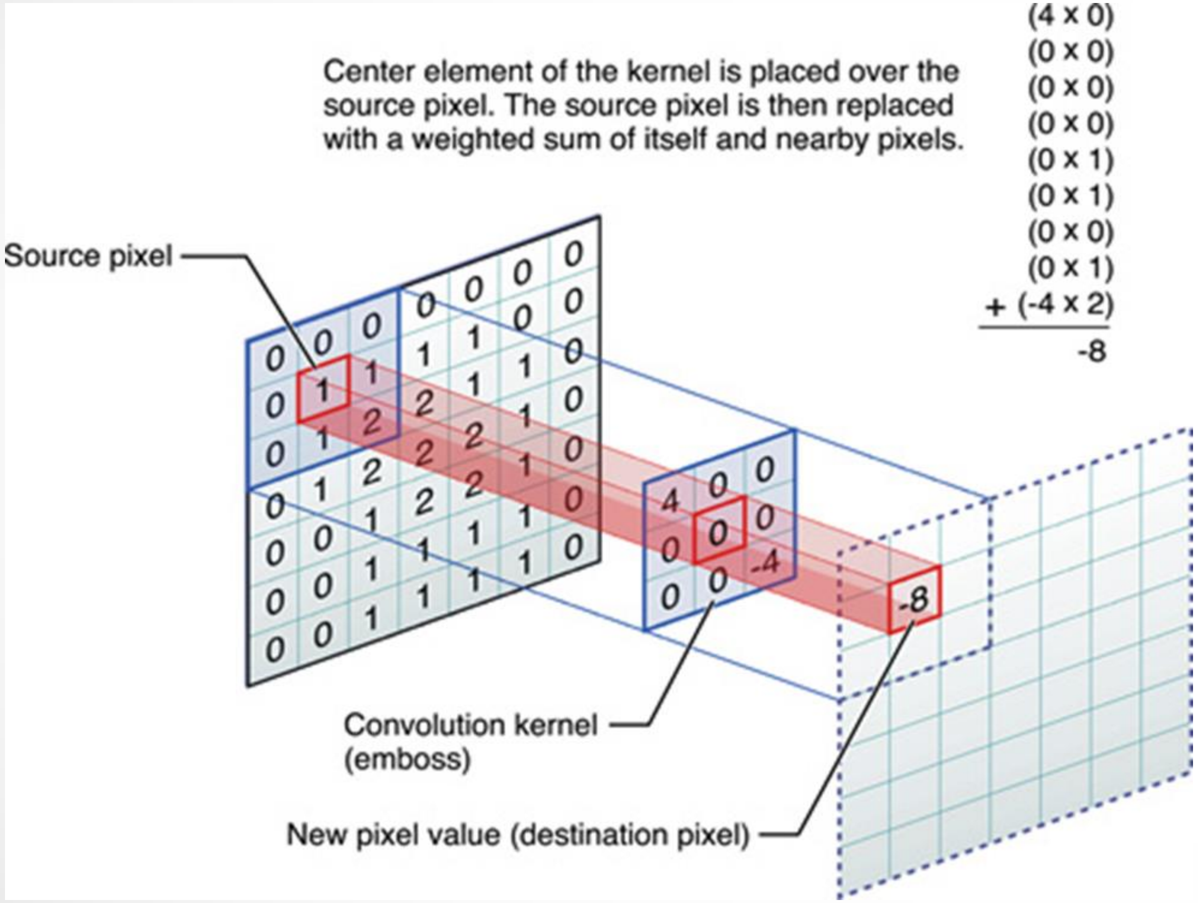
4		

Convolved Feature



https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

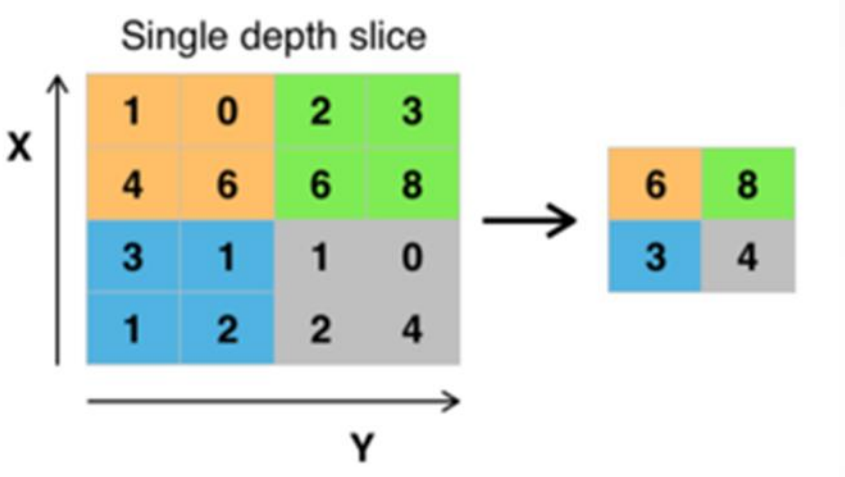
Convolutional Neural Network



Center element of the kernel is placed over the source pixel. The source pixel is then replaced with a weighted sum of itself and nearby pixels.

(4 x 0)
(0 x 0)
(0 x 0)
(0 x 0)
(0 x 1)
(0 x 1)
(0 x 0)
(0 x 1)
+ (-4 x 2)
-8

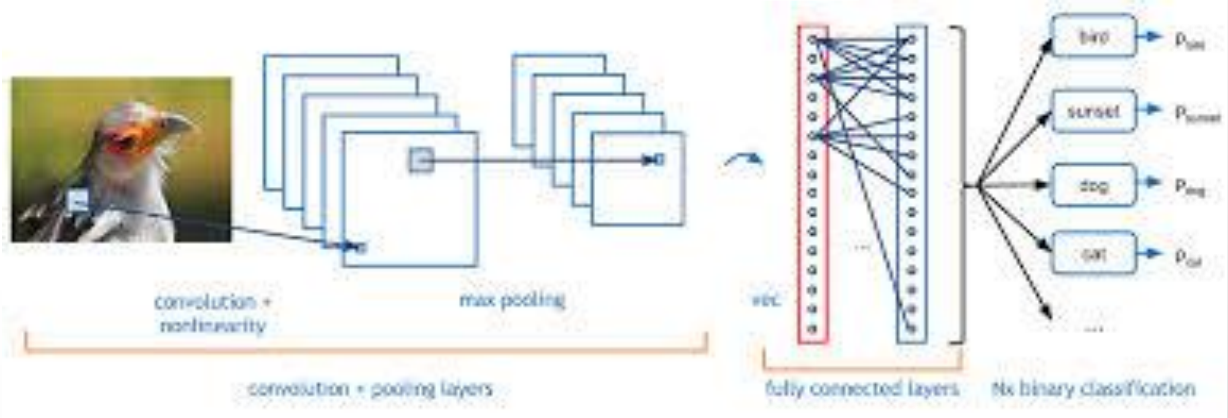
Source pixel
Convolution kernel (emboss)
New pixel value (destination pixel)



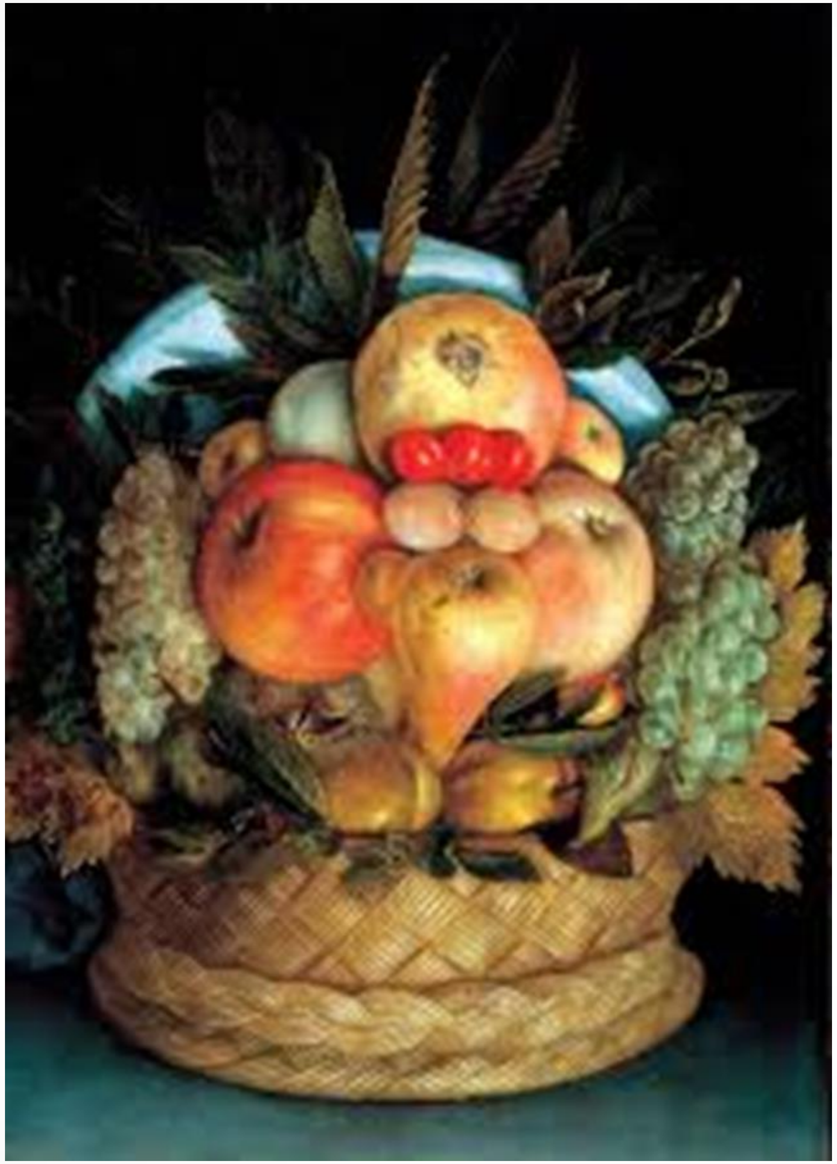
Single depth slice

1	0	2	3
4	6	6	8
3	1	1	0
1	2	2	4

6	8
3	4



Q & A



The Fruit Basket is a c.1590 oil on panel still life by Giuseppe Arcimboldo