CSE428: Image Processing

Lecture 1: Introduction

June 2, 2022

Outline

- Signal and Data
- What is Image
- Digital Image
- Computer Representation
- Types of Digital Images
- Image Processing - Examples
- Application of Image Processing

Signal (Data)

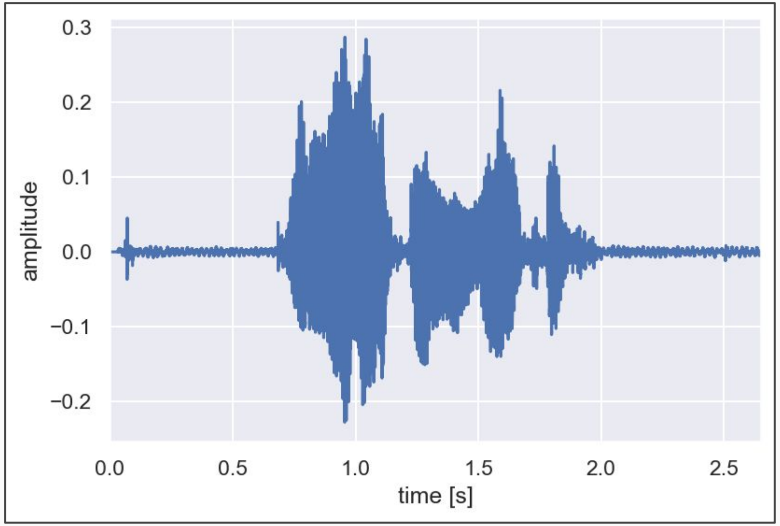
- A **function** containing **information** about some phenomena

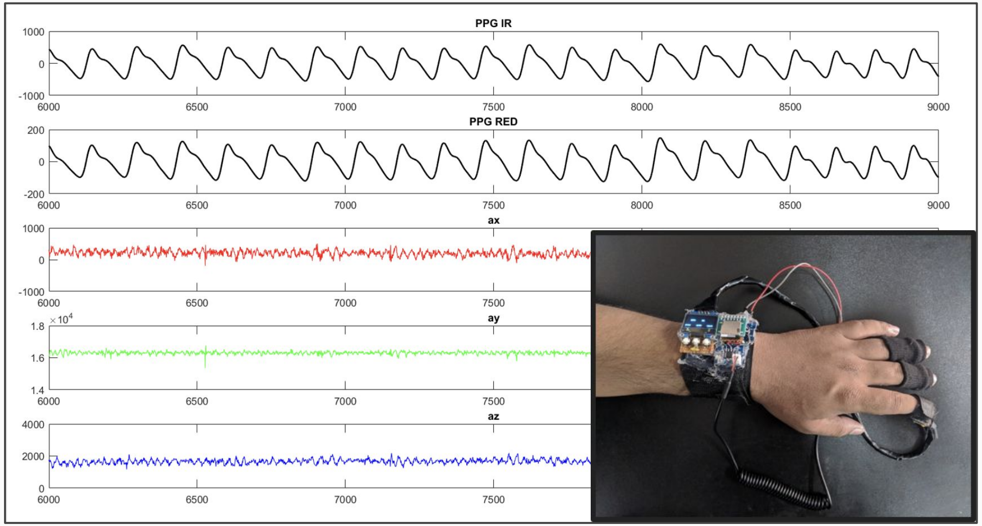
- Any physical quantity (temperature, light intensity, position) varying with time (t) or space (x, y, z…)

- f(t), f(x, y), f(x, t)

Examples of Signal - 1D

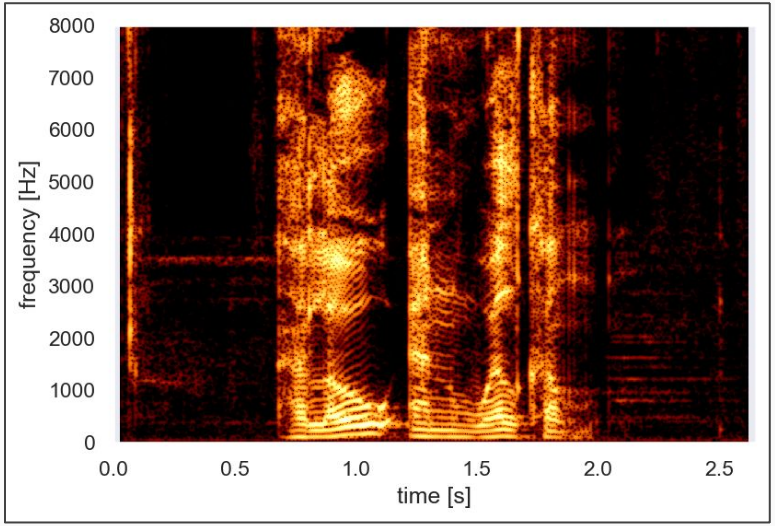
1D signal - f(x) or f(t)

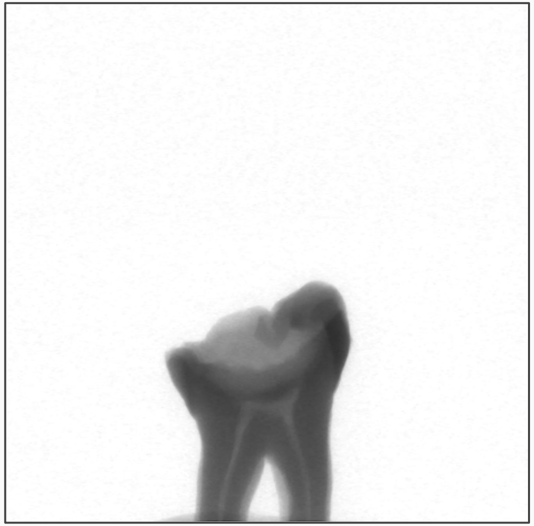




Examples of Signal - 2D

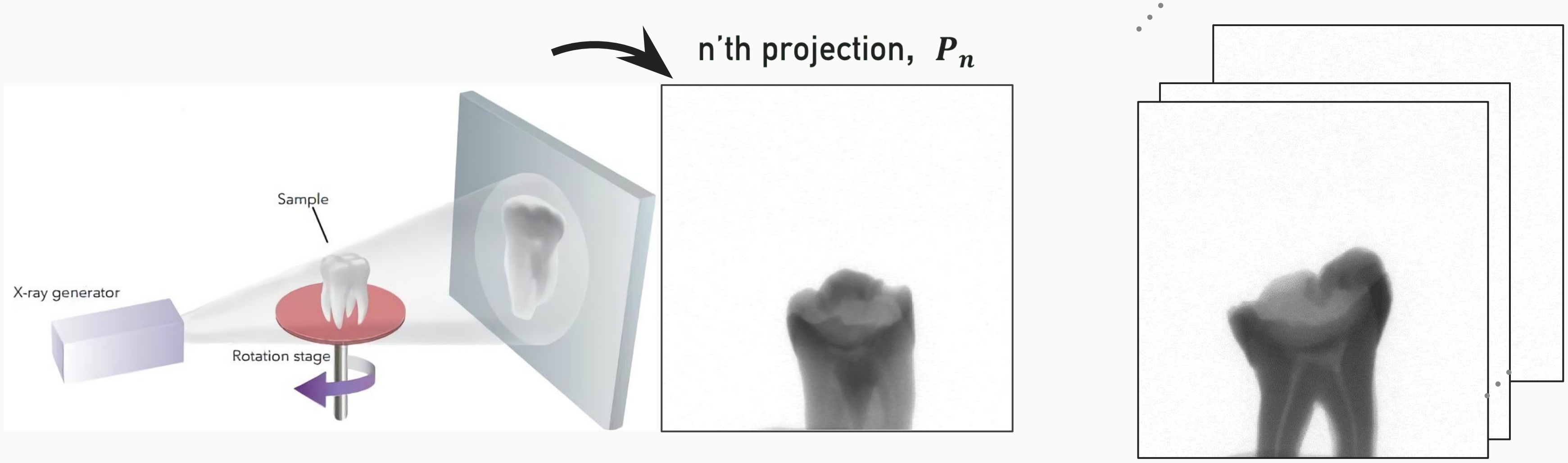
2D signal - f(x, y) or f(x, t)





Examples of Signal - 3D

3D signal - f(x, y, t)



Image

- A visual representation in the form $f(x, y)$
- $f \rightarrow$ brightness or color at (x, y)
- (x, y) are called spatial coordinates
- Usually f, x, and y have continuous values

Image - Paintings

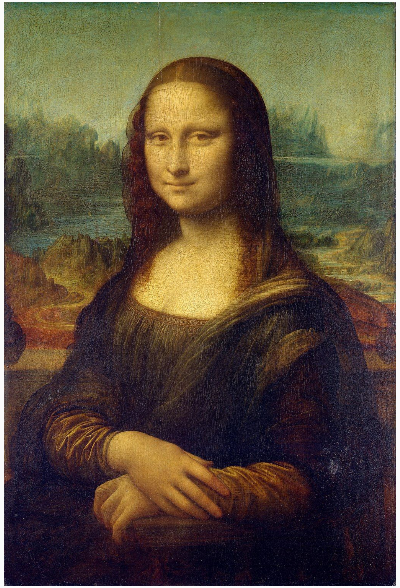


Image - Medical Images

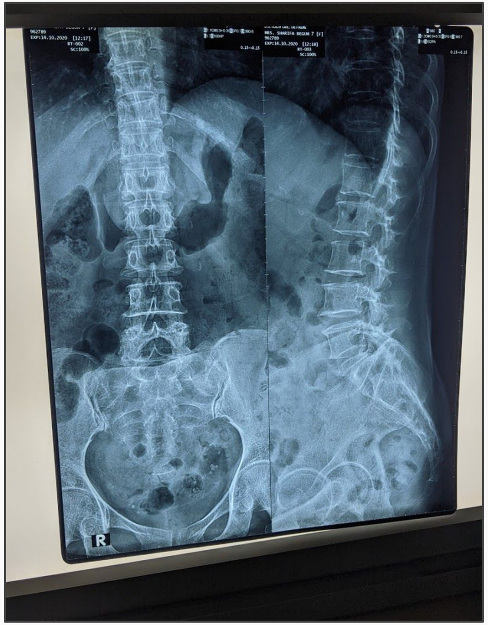
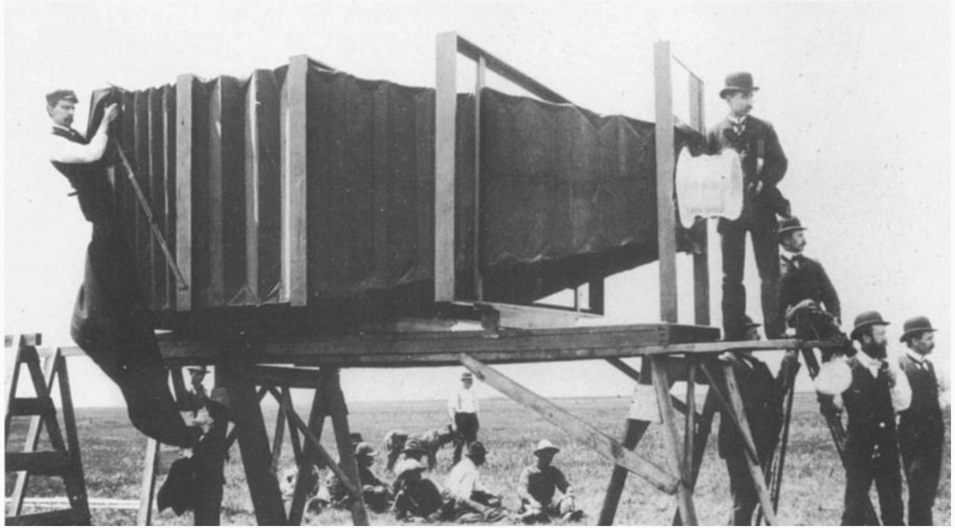


Image - “Traditional” Photography

Optical Array → Sensor (Film) → Chemicals → Final Image



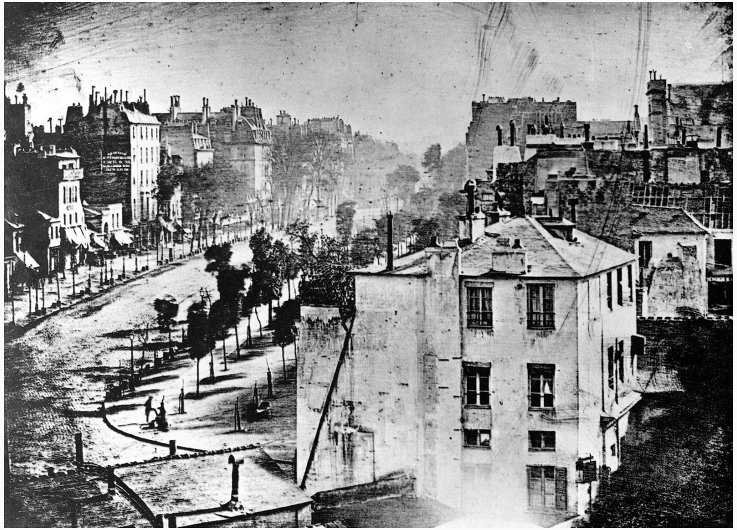


Image - “Traditional” Photography

Optical Array → Sensor (Film) → Chemicals → Final Image

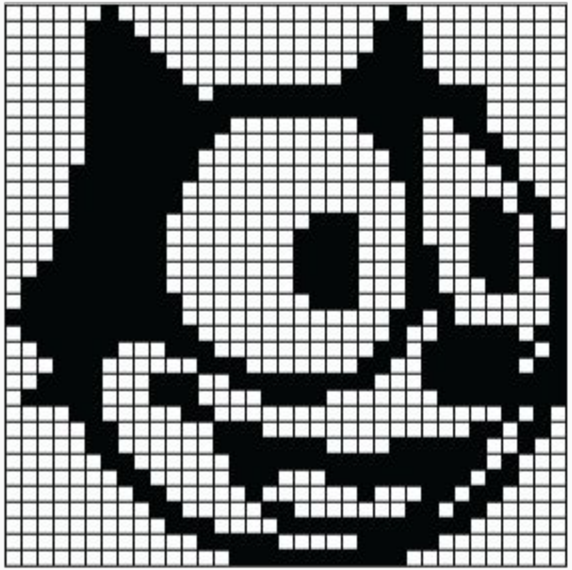
Cannot be stored!

(Digitally)

Digital Image

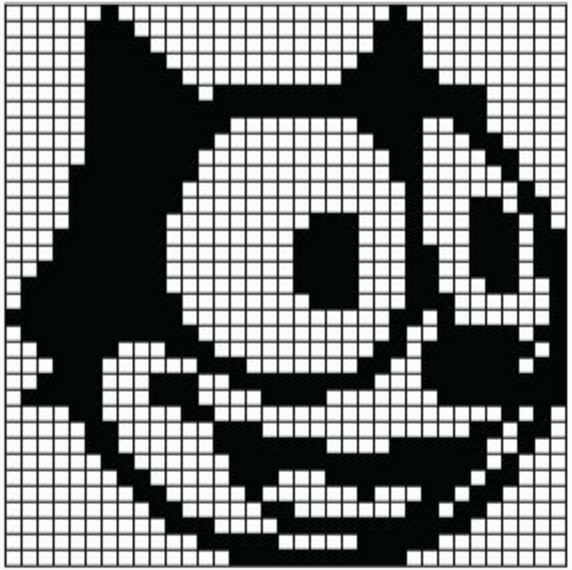
- Discrete (and finite) samples *f*[x, y]
- Composed of finite number of elements - **pixels**
- x, y → Pixel location or index (0, 1, 2, 3, …. ~~1.5~~, ~~2.345~~)
- *f*[x, y] → particular value(s)

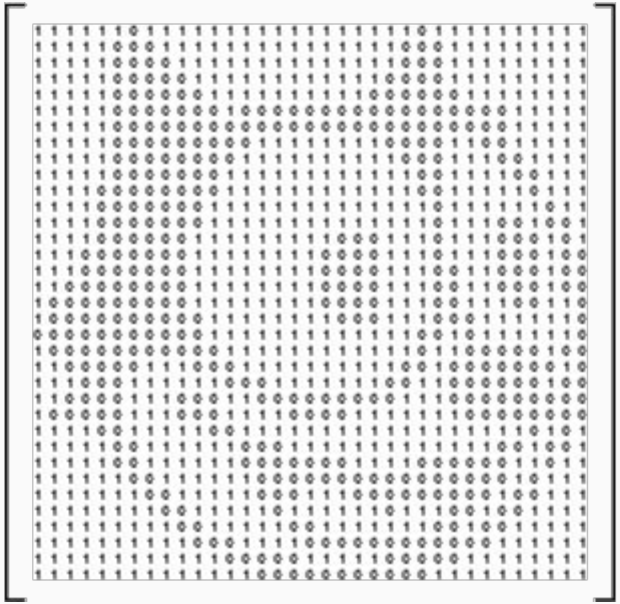
Storing Digital Image - Black & White



Storing Digital Image - Black & White

Images are stored as matrix (high -> white, low -> black)

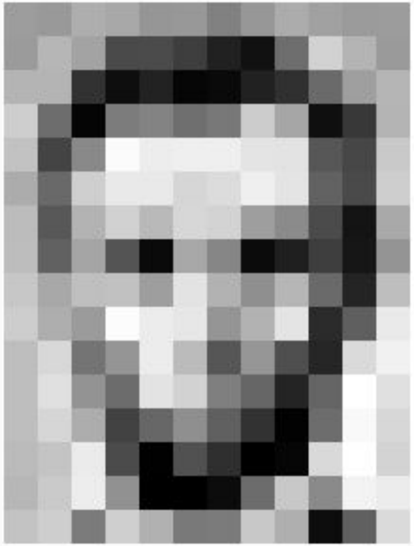




35x35

Storing Digital Image - Grayscale

Images are stored as matrix (high -> white, low -> black, in-between -> gray)



157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

157	153	174	168	150	152	129	151	172	161	155	156
155	182	163	74	75	62	33	17	110	210	180	154
180	180	50	14	34	6	10	33	48	106	159	181
206	109	5	124	131	111	120	204	166	15	56	180
194	68	137	251	237	239	239	228	227	87	71	201
172	105	207	233	233	214	220	239	228	98	74	206
188	88	179	209	185	215	211	158	139	75	20	169
189	97	165	84	10	168	134	11	31	62	22	148
199	168	191	193	158	227	178	143	182	106	36	190
205	174	155	252	236	231	149	178	228	43	95	234
190	216	116	149	236	187	86	150	79	38	218	241
190	224	147	108	227	210	127	102	36	101	255	224
190	214	173	66	103	143	96	50	2	109	249	215
187	196	235	75	1	81	47	0	6	217	255	211
183	202	237	145	0	0	12	108	200	138	243	236
195	206	123	207	177	121	123	200	175	13	96	218

Computer Representation - Grayscale Version 1

- For an image with M rows and N columns
  - x = 0, 1, 2, … , (N - 1) where N is the **width** of the image (also called W)
  - y = 0, 1, 2, … , (M - 1) where M is the **height** of the image (also called H)
- Numerical array form `[f(x, y)]`
- **(i, j)** th pixel value `[f(i, j)]` is the image intensity at point `(i, j)`
- In Python - `img[y, x]`, `img.shape = (H, W)`

x

y

$$f(x,y) = \begin{bmatrix} f(0,0) & f(1,0) & \cdots & f(N-1,0) \\ f(0,1) & f(1,1) & \cdots & f(N-1,1) \\ \vdots & \vdots & f(i,j) & \vdots \\ f(0,M-1) & f(1,M-1) & \cdots & f(N-1,M-1) \end{bmatrix} = \mathtt{im}[y,x] = \begin{bmatrix} \mathtt{im}(0,0) & \mathtt{im}(0,1) & \cdots & \mathtt{im}(0,N-1) \\ \mathtt{im}(1,0) & \mathtt{im}(1,1) & \cdots & \mathtt{im}(1,N-1) \\ \vdots & \vdots & \mathtt{im}(j,i) & \vdots \\ \mathtt{im}(M-1,0) & \mathtt{im}(M-1,1) & \cdots & \mathtt{im}(M-1,N-1) \end{bmatrix}$$

Computer Representation - Grayscale Version 2

- For an image with M rows and N columns
  - x = 0, 1, 2, … , (M - 1) where M is the **height** of the image (also called H)
  - y = 0, 1, 2, … , (N - 1) where N is the **width** of the image (also called W)
- Numerical array form `[f(x, y)]`
- **(i, j)** th pixel value `[f(i, j)]` is the image intensity at point `(i, j)`
- In Python - `img[x, y]`, `img.shape = (H, W)`

y

x

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

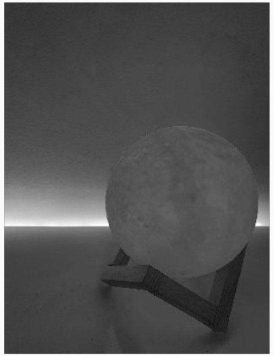
$$\mathbf{A} = \begin{bmatrix} a_{0,0} & a_{0,1} & \cdots & a_{0,N-1} \\ a_{1,0} & a_{1,1} & \cdots & a_{1,N-1} \\ \vdots & \vdots & & \vdots \\ a_{M-1,0} & a_{M-1,1} & \cdots & a_{M-1,N-1} \end{bmatrix}$$

Storing Digital Image - Color Image Version 1

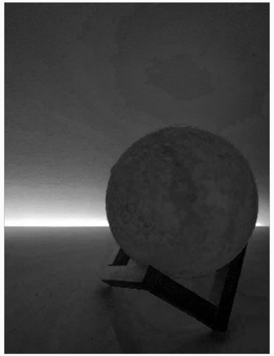
- Color image = 3 grayscale image (RGB model)
- R = Red, G = Green, B = Blue [each grayscale images are called **channel**]
- Numerical array form `[f(x, y, n)]`, n = index of channel = 0, 1, 2
- In Python - `img[y, x, n]`, `img.shape = (H, W, #channels)`

Red

Green



Blue

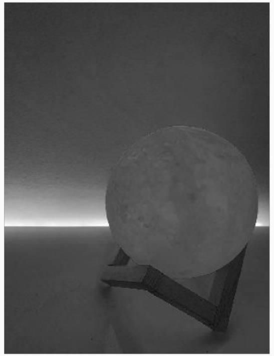


Storing Digital Image - Color Image Version 1

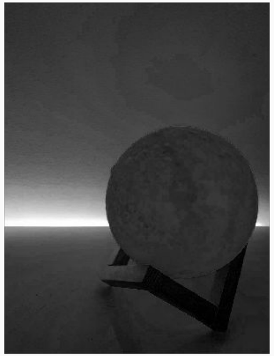
- Color image = 3 grayscale image (RGB model)
- R = Red, G = Green, B = Blue [each grayscale images are called **channel**]
- Numerical array form `[f(x, y, n)]`, n = index of channel = 0, 1, 2
- In Python - `img[y, x, n]`, `img.shape = (H, W, #channels)`

Red

Green



Blue



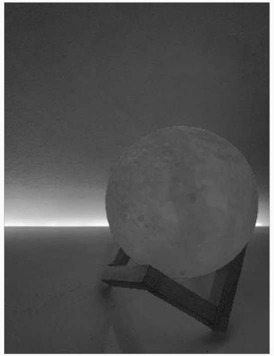
Combined

Storing Digital Image - Color Image Version 2

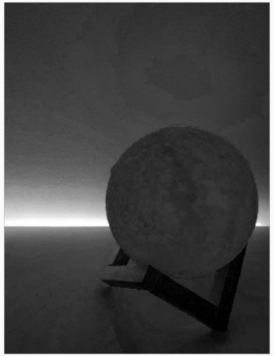
- Color image = 3 grayscale image (RGB model)
- R = Red, G = Green, B = Blue [each grayscale images are called **channel**]
- Numerical array form `[f(x, y, n)]`, n = index of channel = 0, 1, 2
- In Python - `img[x, y, n]`, `img.shape = (H, W, #channels)`

Red

Green



Blue

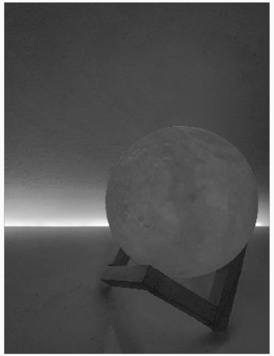


Storing Digital Image - Color Image Version 2

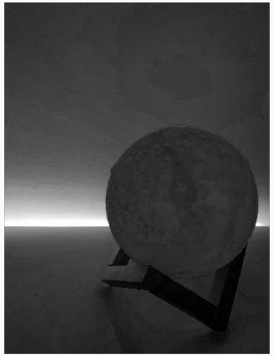
- Color image = 3 grayscale image (RGB model)
- R = Red, G = Green, B = Blue [each grayscale images are called **channel**]
- Numerical array form `[f(x, y, n)]`, n = index of channel = 0, 1, 2
- In Python - `img[x, y, n]`, `img.shape = (H, W, #channels)`

Red

Green



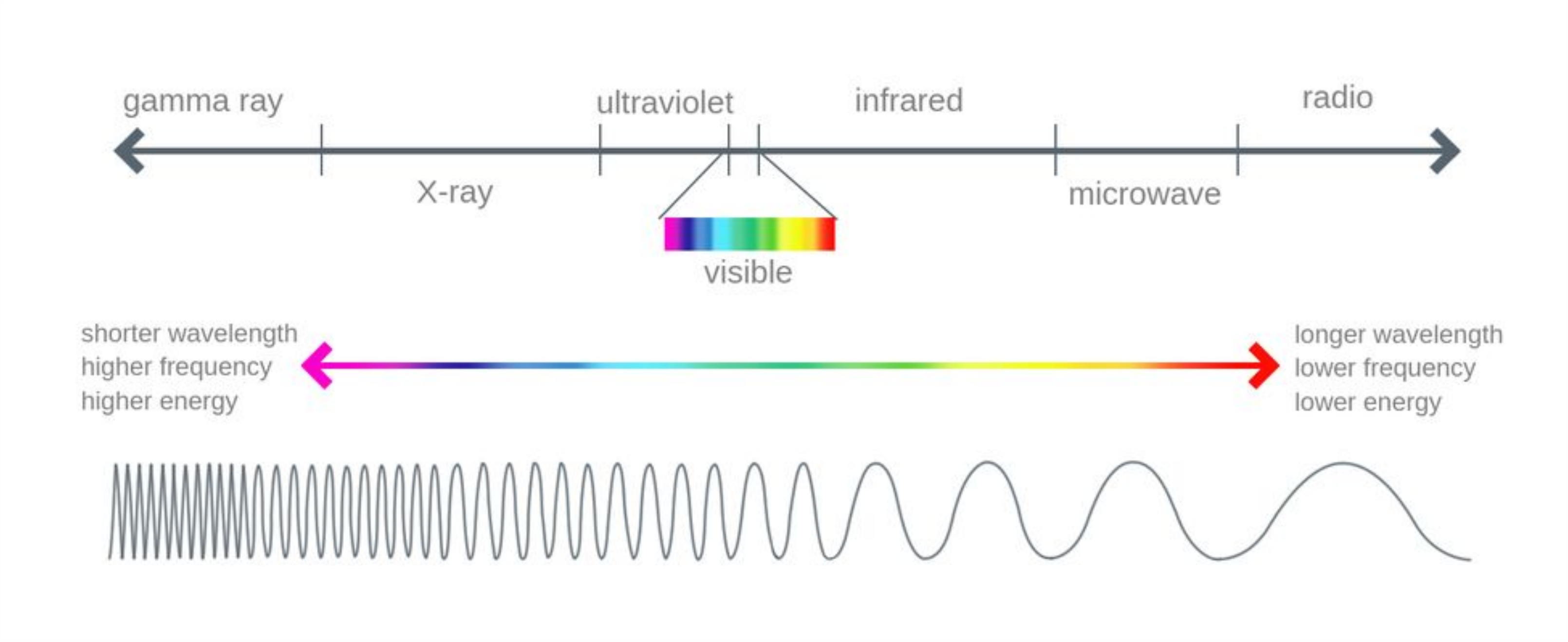
Blue



Combined

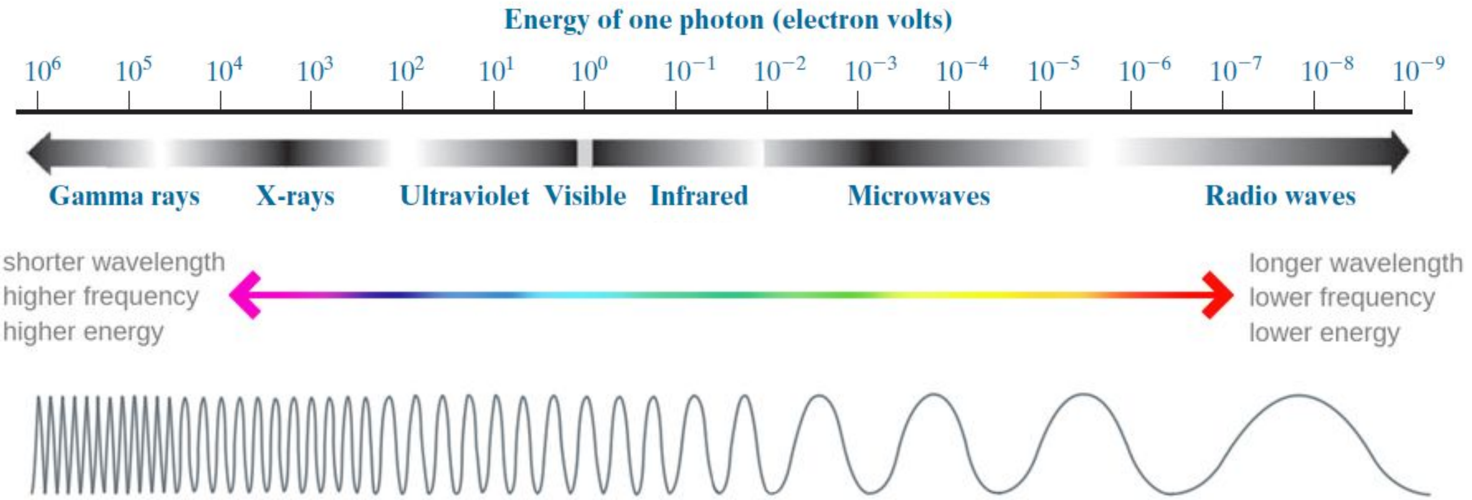
Classification of Digital Image

According to their source*



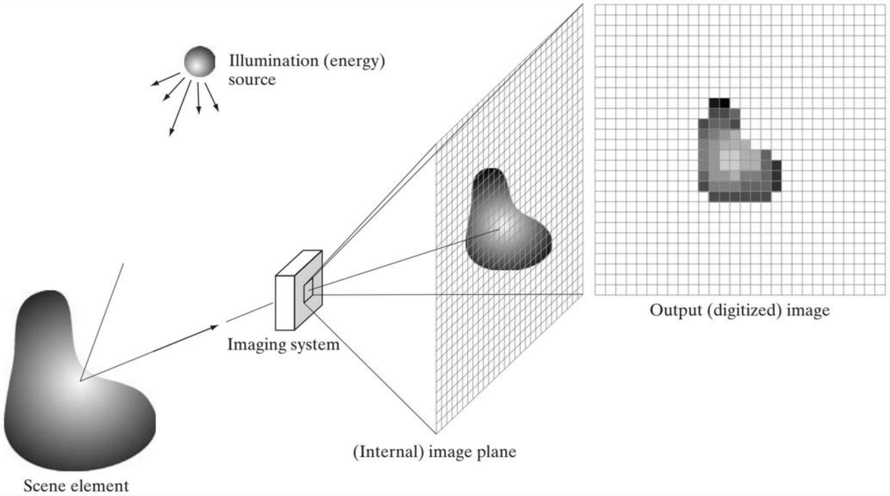
Classification of Digital Image

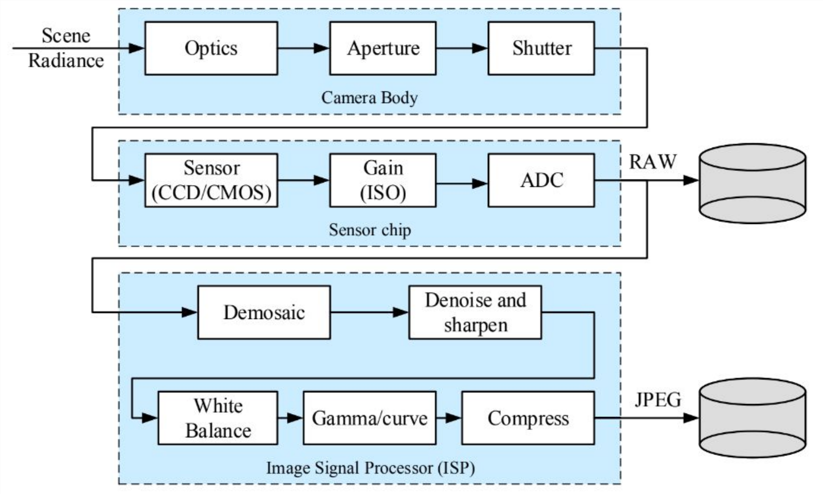
According to their source*



Classification of Digital Image

Reflection Image

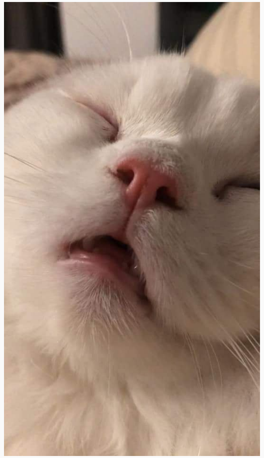


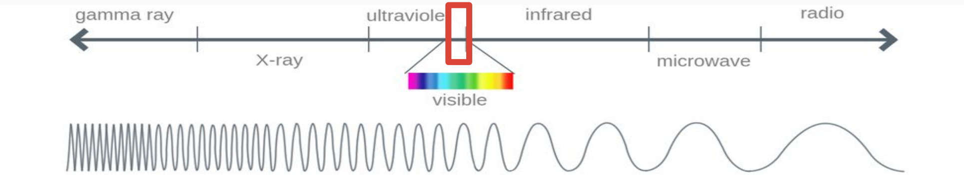


Information primarily about object's surface

Classification of Digital Image

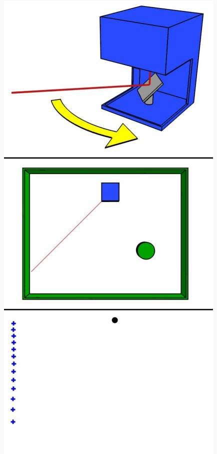
Reflection Image - Visible

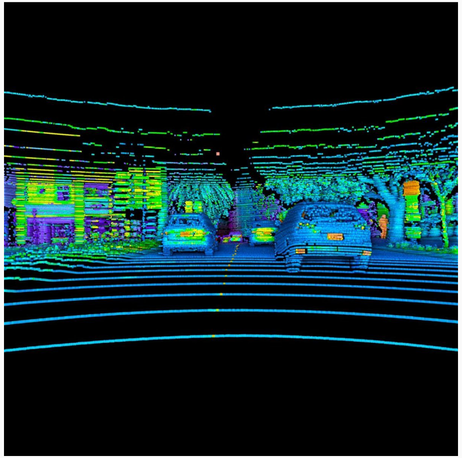


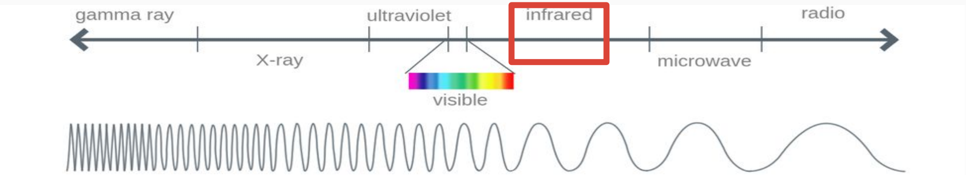


Classification of Digital Image

Reflection Image - LiDAR

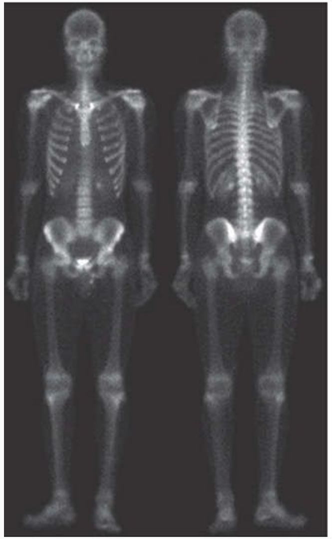


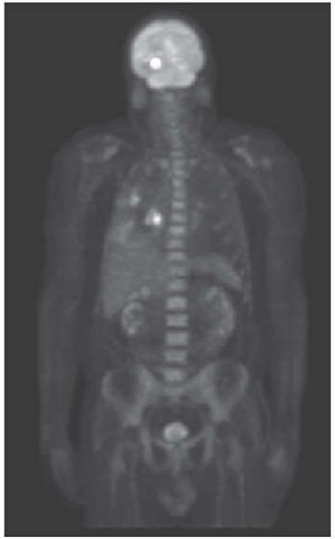




Classification of Digital Image

Emission Image





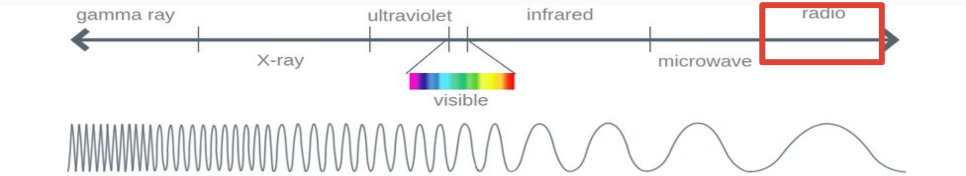
(a) Gamma ray
(b) PET Image (gamma)
(c) Infrared (IR)

Information primarily about object's Internal

Classification of Digital Image

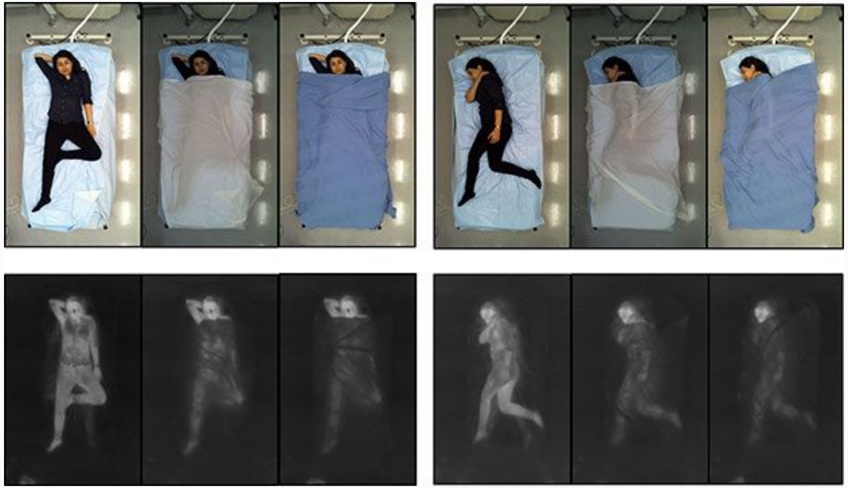
Emission Image

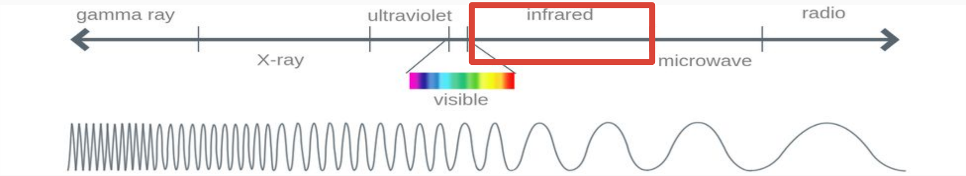
MRI Image



Classification of Digital Image

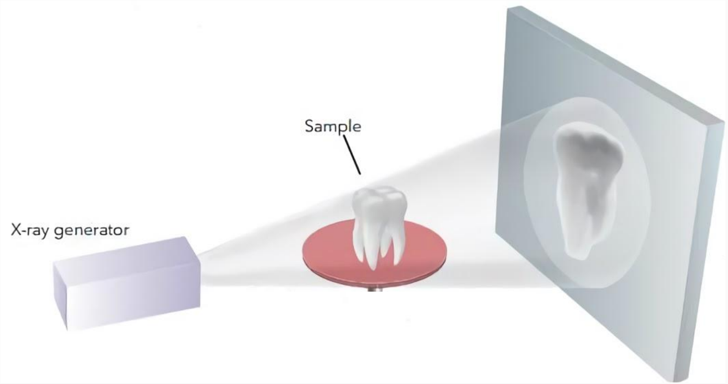
Emission Image



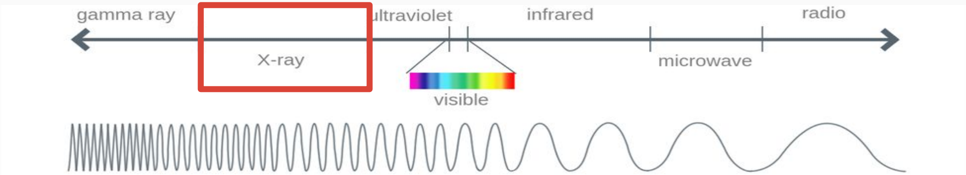


VIP Cup 2021 Challenge
SLP Pose Estimation

Classification of Digital Image
Absorption Image - X Ray



X Ray



Information primarily about object's Internal structure

Classification of Digital Image

Multimodal Image - LANDSAT Satellite Image

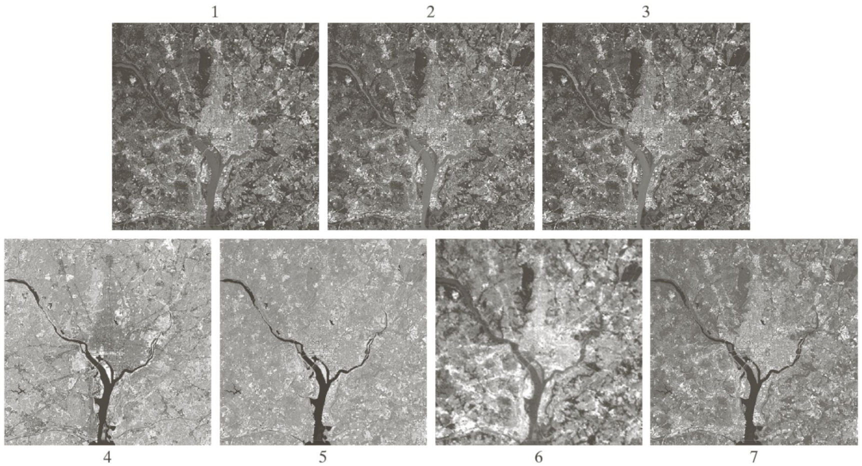


FIGURE 1.10 LANDSAT satellite images of the Washington, D.C. area. The numbers refer to the thematic bands in Table 1.1. (Images courtesy of NASA.)

TABLE 1.1 Thematic bands in NASA's LANDSAT satellite.

Band No.	Name	Wavelength (μm)	Characteristics and Uses
1	Visible blue	0.45–0.52	Maximum water penetration
2	Visible green	0.52–0.60	Good for measuring plant vigor
3	Visible red	0.63–0.69	Vegetation discrimination
4	Near infrared	0.76–0.90	Biomass and shoreline mapping
5	Middle infrared	1.55–1.75	Moisture content of soil and vegetation
6	Thermal infrared	10.4–12.5	Soil moisture; thermal mapping
7	Middle infrared	2.08–2.35	Mineral mapping

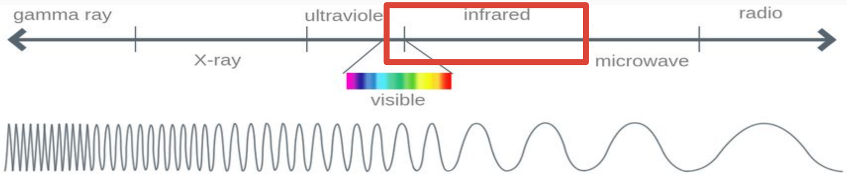


Image Processing

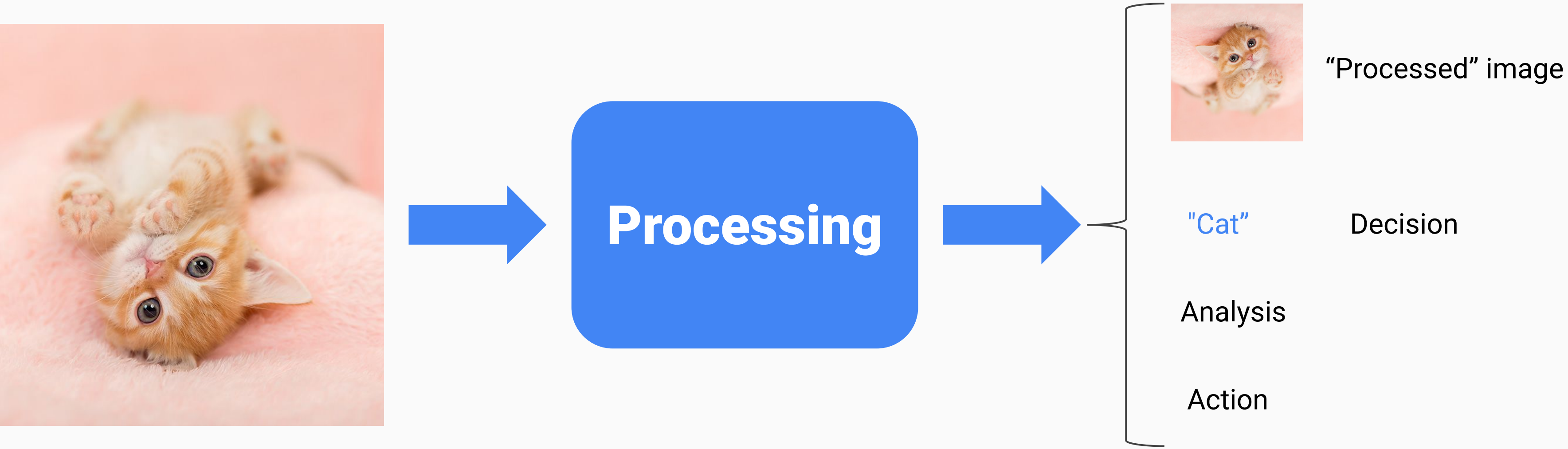


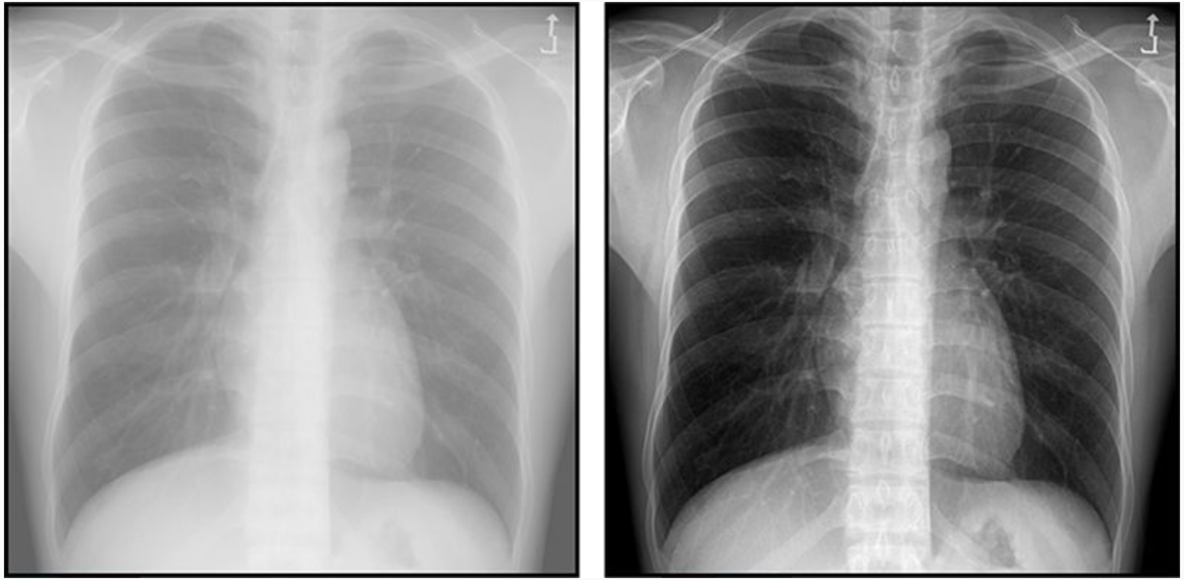
Image Processing - Examples

Image Enhancement

Contrast Limited Adaptive **Histogram Equalization**

Image Processing - Examples

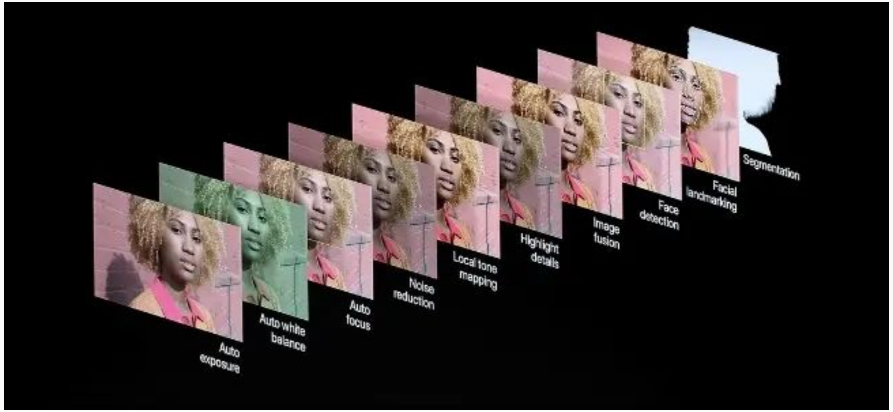
Image Enhancement

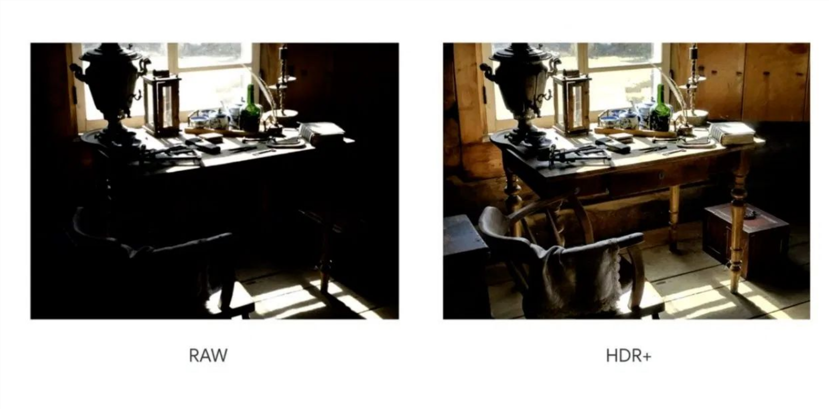


Intensity transformation, Contrast enhancement

Image Processing - Examples

Image Enhancement





HDR Image, Computational Photography

Image Processing - Examples

Image Noise Reduction/Removal

Input Frame

Output Frame

CNN Based de-Raining

Image Processing - Examples

Image Reconstruction

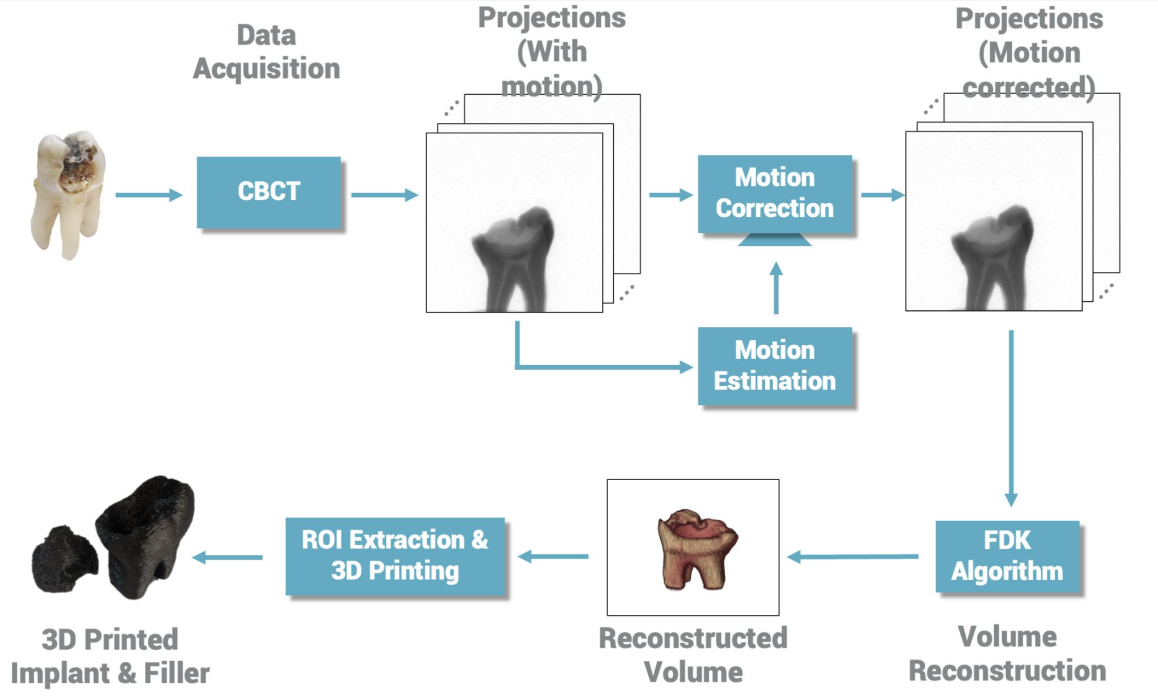


Image Processing - Examples

Image Recognition (Computer Vision)

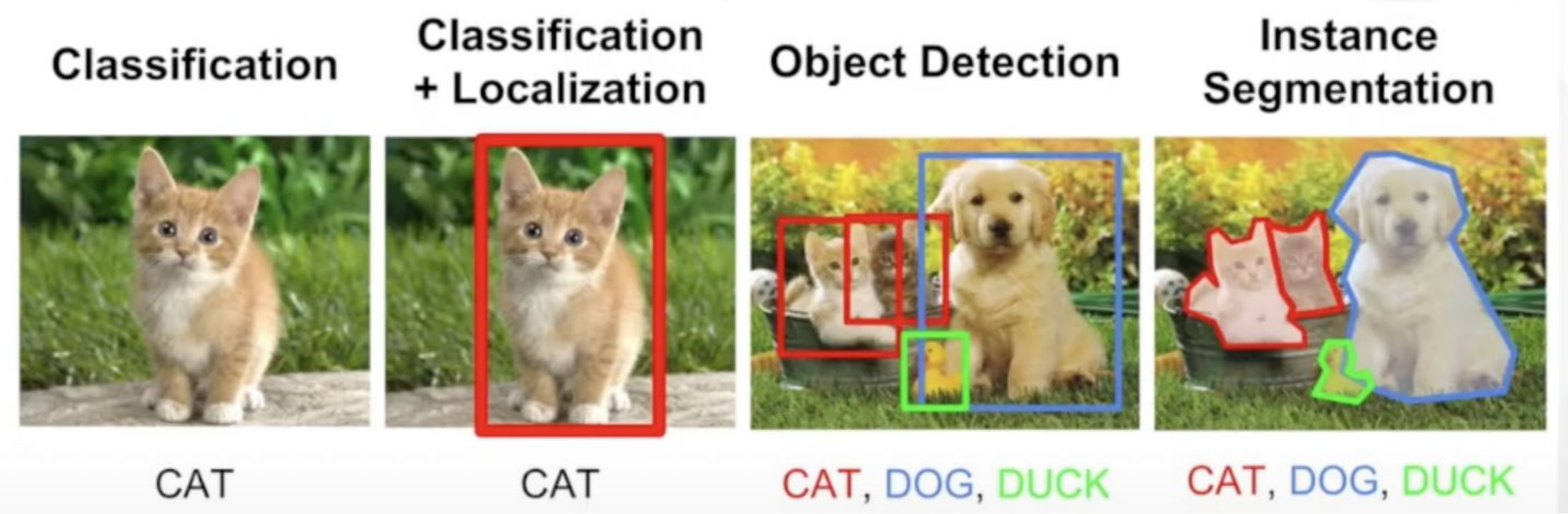
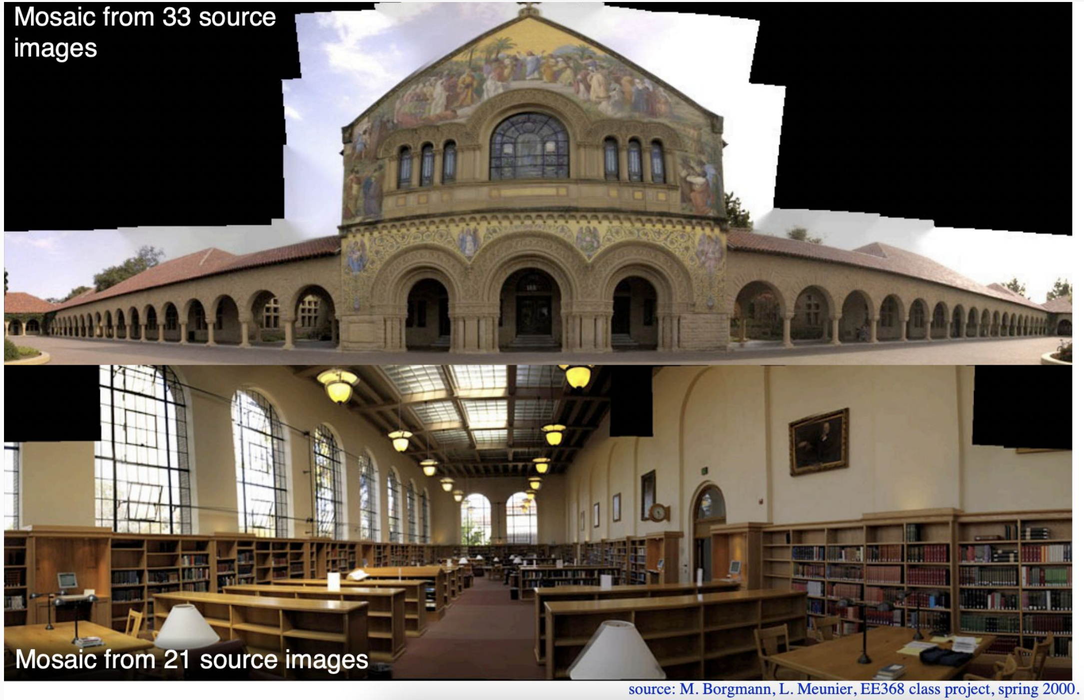


Image Processing - Examples

Image Stitching



Application of Image Processing

- Medicine
  - Medical Imaging, anomaly detection, automatic disease detection
- Machine Vision
  - Quality control
  - Measurement of position, orientation, sorting
- Military
  - Battlefield awareness
  - Object tracing
- Autonomous Vehicles
  - Self-driving cars (Tesla)
  - Self exploration (NASA's Curiosity, Perseverance)

Questions?