Applied Data Science

Session 35_36: Module 4: Processing Images

Dr. Soharab Hossain Shaikh

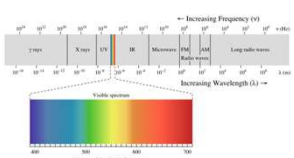
1

Image Processing

Formation and Basic Operations

2

Image Formation



- There are two parts to the image formation process:
  - The **geometry of image formation**, which determines where in the image plane the projection of a point in the scene will be located.
  - The **physics of light**, which determines the brightness of a point in the image plane as a function of illumination and surface properties.

3

Image Formation

- The scene is illuminated by a single source.
- The scene reflects radiation towards the camera.
- The camera senses it via chemicals on film (old film camera) or CCD sensors (digital cameras).

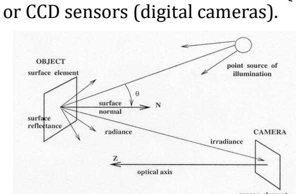
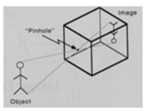


Image Source: Book: Rafael C. Gonzalez, Richard E. Woods; Digital Image Processing; 3rd Ed., Pearson.

4

Pinhole Camera & Projection

- This is the simplest device to form an image of a 3D scene on a 2D surface.
- Straight rays of light pass through a “pinhole” and form an inverted image of the object on the image plane.



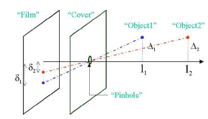
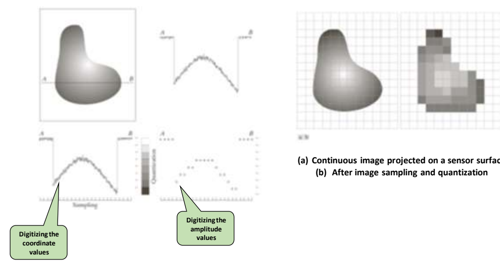


Image Source: Book: Rafael C. Gonzalez, Richard E. Woods; Digital Image Processing; 3rd Ed., Pearson.

5

Discretization: Sampling and Quantization



(a) Continuous image projected on a sensor surface
(b) After image sampling and quantization

Image Source: Book: Rafael C. Gonzalez, Richard E. Woods; Digital Image Processing; 3rd Ed., Pearson.

6

Representing Digital Images

- The representation of a digital image as a 2-dimensional array of dimension M×N with numeric intensity levels for the pixels.

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

- Each pixel has a unique *(i, j)* co-ordinate in the 2D space.
- There are ***M*** number of pixel rows and ***N*** number of pixel columns.

7

Representing Digital Images

- Discrete intensity interval [0, L-1], $L=2^k$
- The number b of bits required to store a M × N digitized image

b = M × N × k

Number of storage bits for various values of *N* and *k*.

N/*k*	1 (*L* = 2)	2 (*L* = 4)	3 (*L* = 8)	4 (*L* = 16)	5 (*L* = 32)	6 (*L* = 64)	7 (*L* = 128)	8 (*L* = 256)
32	1,024	2,048	3,072	4,096	5,120	6,144	7,168	8,192
64	4,096	8,192	12,288	16,384	20,480	24,576	28,672	32,768
128	16,384	32,768	49,152	65,536	81,920	98,304	114,688	131,072
256	65,536	131,072	196,608	262,144	327,680	393,216	458,752	524,288
512	262,144	524,288	786,432	1,048,576	1,310,720	1,572,864	1,835,008	2,097,152
1024	1,048,576	2,097,152	3,145,728	4,194,304	5,242,880	6,291,456	7,340,032	8,388,608
2048	4,194,304	8,388,608	12,582,912	16,777,216	20,971,520	25,165,824	29,369,128	33,554,432
4096	16,777,216	33,554,432	50,331,648	67,108,864	83,886,080	100,663,296	117,440,512	134,217,728
8192	67,108,864	134,217,728	201,326,592	268,435,456	335,544,320	402,653,184	469,762,048	536,870,912

8

Processing: Image Transpose

- The transpose image $\mathbf{B}$ $(M \times N)$ of $\mathbf{A}$ $(N \times M)$ can be obtained as $B(j,i) = A(i,j)$ $(i = 0, \ldots, N-1,\ j = 0, \ldots, M-1)$.

A

B

9

Processing: Vertical Flip

- The vertical flipped image $\mathbf{B}$ $(N \times M)$ of $\mathbf{A}$ $(N \times M)$ can be obtained as $B(i, M-1-j) = A(i,j)$ $(i = 0, \ldots, N-1,\ j = 0, \ldots, M-1)$.

A

B

10

Processing: Cropping Image

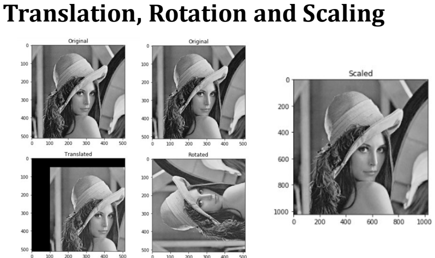
- The cropped image $\mathbf{B}$ $(N_1 \times N_2)$ of $\mathbf{A}$ $(N \times M)$, starting from (n_1, n_2), can be obtained as $B(k,l) = A(n_1+k, n_2+l)$ $(k = 0, \ldots, N_1-1,\ l = 0, \ldots, N_2-1)$.

A

B

11

Translation, Rotation and Scaling



Rotating images at an angle other than a multiple of 90 degrees cause noisy stair-case effects along sharp edges.
This is due to the misalignment of intensities on the two-dimensional image grid.

12

Translation, Rotation, Scaling

- A point (x,y) can be translated/shifted to a new coordinate (t_x, t_y). Translation can be done by the following transformation matrix:

$$M = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix}$$

- Rotation can be done by the following transformation matrix:

$$M = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix}$$

- **OpenCV** supports a scaled rotation with a adjustable center of rotation with the following transformation matrix:

$$\begin{bmatrix} \alpha & \beta & (1-\alpha)\cdot center.x - \beta \cdot center.y \\ -\beta & \alpha & \beta \cdot center.x + (1-\alpha)\cdot center.y \end{bmatrix}$$

$$\alpha = scale \cdot \cos\theta,$$
$$\beta = scale \cdot \sin\theta$$

- Scaling can be done by resizing the image.

13

Computing Image Statistics

- The sample mean (m_A) of an image $\mathbf{A}$ ($N \times M$):

$$m_A = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} A(i,j)}{NM}$$

- The sample variance (σ_A^2) of $\mathbf{A}$:

$$\sigma_A^2 = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (A(i,j) - m_A)^2}{NM}$$

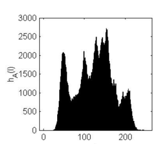
- The sample standard deviation, $\sigma_A = \sqrt{\sigma_A^2}$.

Mean and standard deviation are very common statistic calculated on an image. They are of huge importance and being used in most of the statistical analysis of the image such as equalization, matching, texture analysis with moments, segmentation with binarization etc.

14

Image Histogram

A



$h_A(l)$

Let S be a set and define $\#S$ to be the cardinality of this set, i.e., $\#S$ is the number of elements of S.

- The histogram $h_A(l)$ ($l = 0, \ldots, 255$) of the image $\mathbf{A}$ is defined as:

$$h_A(l) \; = \; \#\{(i,j) \mid A(i,j) = l, \; i = 0, \ldots, N-1, \; j = 0, \ldots, M-1\}$$

- Note that:

$$\sum_{l=0}^{255} h_A(l) = \text{Number of pixels in } \mathbf{A}$$

15

Spatial and Intensity Resolution

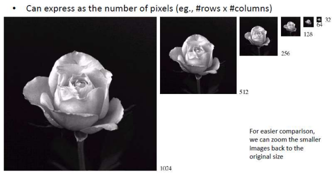
- **Spatial resolution**
  - — A measure of the smallest discernible detail in an image
  - — stated with *line pairs per unit distance, dots (pixels) per unit distance, dots per inch (dpi)*
  - — *Can be expressed in terms of the number of pixels in the image*

- **Intensity or Gray Level resolution**
  - — The smallest discernible **change in intensity level**
  - — stated with *8 bits, 12 bits, 16 bits, etc.*
  - — Can be expressed in terms of number of gray levels in the image

16

Spatial Resolution

- Can express as the number of pixels (eg., #rows x #columns)



For easier comparison, we can zoom the smaller images back to the original size

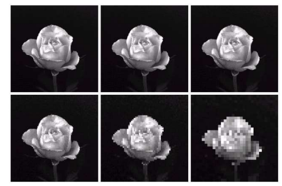
Effect of Reducing Spatial Resolution
Image shown at 1024x1024 8-bit image subsampled down to size
512x512 -> 256x256 -> 128x128 -> 64x64 -> 32x32

Image Source: Book: Rafael C. Gonzalez, Richard E. Woods; Digital Image Processing; 3rd Ed., Pearson.

17

Spatial Resolution

- Re-expanding (by replication) to the original size



a b c
d e f

Effect of Up-sampling from a low-resolution image
(a) 1024x1024 8-bit image (b) 512x512 resampled into 1024x1024 by row and column duplication (c) through (f)
256x256 -> 128x128 -> 64x64 -> 32x32

Image Source: Book: Rafael C. Gonzalez, Richard E. Woods; Digital Image Processing; 3rd Ed., Pearson.

18

Spatial Resolution

a b
c d

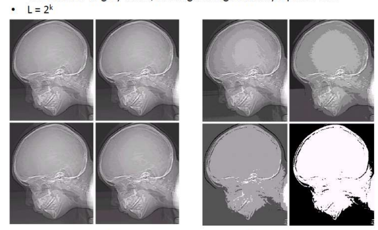
Effect of Reducing Spatial Resolution
Image shown at (a) 1250 dpi (b) 300 dpi (c) 150 dpi (d) 72 dpi

Image Source: Book: Rafael C. Gonzalez, Richard E. Woods; Digital Image Processing; 3rd Ed., Pearson.

19

Gray Level Resolution

- The number of gray levels; in integer images usually a power of 2
- $L = 2^k$

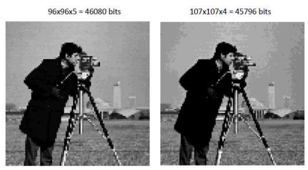


k = 8, 7, 6, 5 k = 4, 3, 2, 1

Image Source: Book: Rafael C. Gonzalez, Richard E. Woods; Digital Image Processing; 3rd Ed., Pearson.

20

Memory Requirements for Storing



If an image is **N x N** pixels
- Each pixel has up to 2^k gray/intensity levels
- Total number of bits stored = **N x N x k** (with no compression)
- We can reduce spatial or gray level intensity to reduce the size of an image

Image Source: Book: Rafael C. Gonzalez, Richard E. Woods; Digital Image Processing; 3rd Ed., Pearson.

21

Image Binarization - Thresholding

22

Smoothing with Gaussian Filter

Gaussian Filter

23

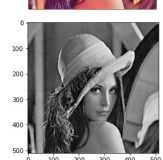
Gray Scale Image

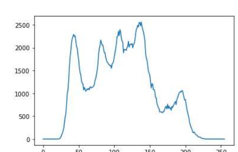
- Basis for color encoding in analog TV in north america (NTSC) and Europe (PAL)
- Y components computed from RGB components as

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

- UV components computed as:

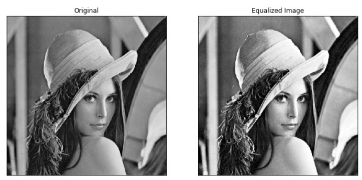
$$U = 0.492 \cdot (B - Y) \quad \text{and} \quad V = 0.877 \cdot (R - Y)$$





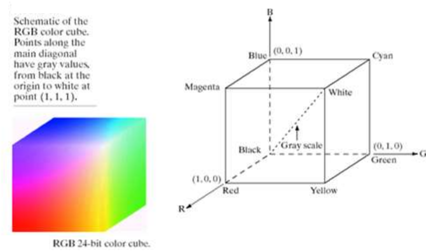
24

Histogram Equalization



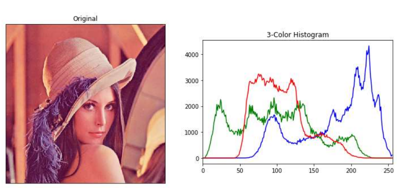
25

RGB Color Model



26

Color Histogram



> A color image represented in RGB color space containing 3 channels – Red (R), Green (G) and Blue(B).

> These three channels are stacked on top of each other anc forms the color sensation for a single pixel.

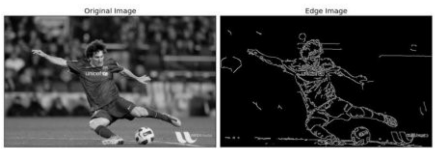
> A RGB image with spatial dimension mxn with 8-bit representation for each color channel will have a storage requirement of **(m x n x 8 x 3)** bits or **(m x n x 3)** bytes.

27

Color Space Conversion

28

Edge Detection



https://docs.opencv.org/3.4/da/d22/tutorial_py_canny.html

29

Object Contour - Bounding Box/Rectangle and Circle

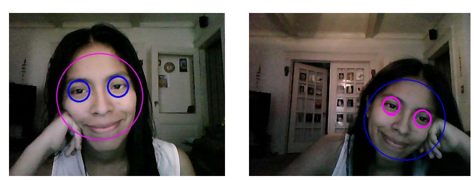
30

Foreground Extraction

https://docs.opencv.org/3.4/d8/d83/tutorial_py_grabcut.html

31

Face and Eye Detection from Human Facial Images



Application of Haar Feature-based Cascade Classifiers

https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html

32

OpenCV

- **OpenCV** is an Open Source Computer Vision and image processing library Originally created by Intel and maintained by Willow Garage and Itseez. It was officially launched in 1999 aiming at real-time computer vision.
- Written natively in C++ - it is a cross-platform library and has support for C++, Python, Java and Matlab.
- The library has more than 2500 optimized algorithms.
- Can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high-resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.
- Licensed under BSD – makes it easy for businesses to utilize and modify code.
- Supports Windows, Linux, Android and Mac-OS.
- https://docs.opencv.org/3.4/d2/d96/tutorial_py_table_of_contents_imgproc.html

33

We will look into some coding demos on…

i) Image colour channel processing
ii) Spatial and intensity resolution
iii) Geometric Transformations
iv) Image Histograms
v) Edge Detection
vi) Contour and Bounding Box
vii) Foreground extraction

34

However, this course is not about Digital Image Processing.

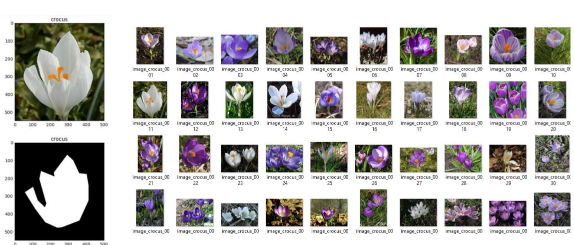
Therefore,
we would like to see <u>how to process image data in the context of a Data Science Project</u>.

<u>So, we will see a demo on</u>
Classifying images into different classes/categories with Machine Learning by applying simple colour-histogram-based feature extraction technique.

Clustering images for colour-based segmentation with K-Means.

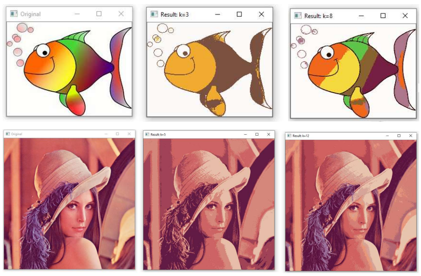
35

Flower Classification with Random Forest



36

Clustering images for colour-based segmentation with K-Means



37

Processing Images for Object Detection and Classification nowadays are performed with Deep Learning producing superb accuracy. Deep learning models like Convolutional Neural Networks (CNNs) have revolutionized the field of Machine Vision.

However, that is beyond the scope of the current course.

38

39

End of the Course Delivery

40