

Programming Session: Computer Vision DSE-312: 15th, 17th March

1. Define two variables x and y and compute their division and sum. Let $x = 10$ and $y=3$. Print out the type of x , y , x/y and $x+y$, $x-y$ as well as the result of their division, sum and subtraction.
2. Create a list which contains the values 1 to 9. Print out the length of the list and compute the sum of all elements in the list using indexing. Replace the second element with the number 42. Write a code to sum the first and last element, second and second last element and so on (leave out the centre element 5 in this computation).
3. Create the following key-value pairs: City : Bhopal, Institute: IISERB, Hostelname – fill here (the bird names), your Lecture room_number: 4. Get the value of Lecture room_number and print it out. Use dictionary (python), cell/structure arrays (matlab). Append “course: DSE-312” to the list of key-value pairs.
4. You are given: $A = \begin{bmatrix} 22 & -22 & 33 \\ -33 & 44 & 55 \\ -55 & -44 & 1 & -2 \end{bmatrix}$. Write a code to compute sum of all entries within A.
5. For the vectors: $\vec{a} = 3\hat{i} - 2\hat{j} + \hat{k}$, $\vec{b} = -7\hat{i} + 2\hat{j} + 3\hat{k}$. compute the i) cross product ii) find the angle between the vectors.
6. Import numpy (for python users) or otherwise (Matlab users). Create and print the following matrices:
 - i. 3x4 matrix with zeros
 - ii. 4x3x3 matrix with ones
 - iii. 3x3 identity matrix
 - iv. 4x6 matrix with uniformly distributed values in the range of [0,1]. Hint: For reproducible results set a seed value by the function np.random.seed (value) or rand function (matlab)
 - v. array with the values from 0 to 4 (integer)
7. Repeat question 1 from the midsem exam including the options (python students, generate the zone plate image using numpy).
8. Binary Images and Connected Components:
 - i. Load the grayscale image shapes.png from disk and print its minimum and maximum value. Finally, visualize the grayscale image (use plotting functions).
 - ii. Convert the grayscale image to a binary image by applying a threshold of $T=100$. Make sure that all pixels $> T$ have an intensity value of 1 (foreground) and all other pixels have an intensity value of 0 (background). Print the minimum and maximum value of your converted image. Finally, visualize the binary image.
 - iii. Implement the Connected Components algorithm and apply it to the previous defined binary image by following each step outlined below:

- iv. Print the number of components in the binary image for *N4 neighborhood* and *N8 neighborhood*. If there is any difference explain why this is the case.
 - v. Implement the Distance Transformation algorithm and apply it to the previous defined binary image and visualize the result for *N4 neighborhood* and *N8 neighborhood*.
 - vi. Apply a *Dilation* to the binary image by using the `skimage.morphology` module (`skimage.morphology.dilation`) (or in built Matlab command). Visualize your results and describe your observations (1 point)
 - vii. Apply the *Distance Transformation* with *N4 neighborhood* to the dilated binary image and visualize your result. (2 points)
 - viii. Apply connected components to the dilated binary image and comment on this result versus iv.
9. Create your own implementation of a function `imgconv (f,w)` for applying the convolution on the following:

$$w = \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix} \quad f = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

10. Import the following (python students) or (Matlab students: use image processing toolbox):

import all required modules

import skimage.io

import numpy as np

import matplotlib.pyplot as plt

from scipy import ndimage

Load the image `images/house-downsampled.png` as a grayscale image and visualize it.

Compute the convolution of the image with the following **3x3** kernels and visualize the results.

- i. Box filter
 - ii. Gaussian filter
 - iii. Sobel Operator. Compute the absolute gradient.
11. In the given binary image, extract the area, eccentricity and orientation of each object within the image (`regionprops` function). Image is `binaryshapes.png`
12. Detect corners within the image. First convert to grayscale and then proceed (`house_colorimage`).