LAB #01: Introduction to Python , OpenCV&Numpy

Lab Objective:

To introduce students with python programming language ,opencv and numpy.

Lab Description:

Installation:

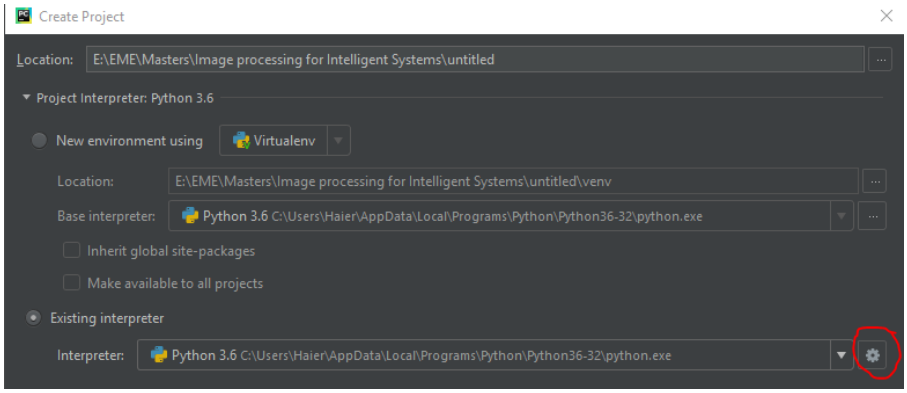
Download and install Python interpreter and PyCharm IDE from the following links below.

Python interpreter: https://www.python.org/downloads/

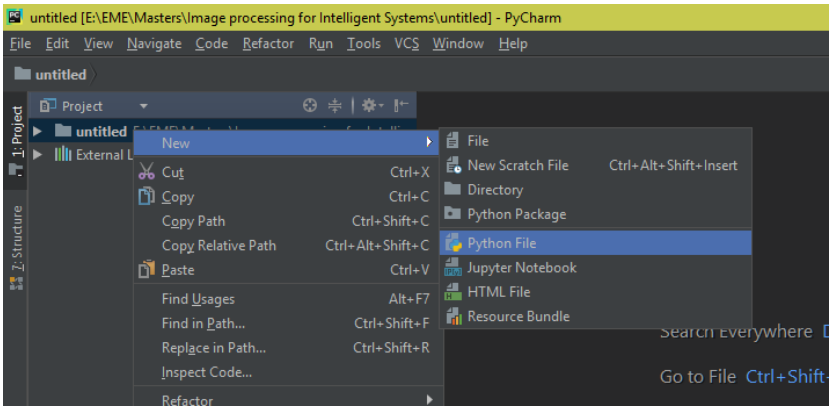
PyCharm IDE: https://www.jetbrains.com/pycharm/download/download-thanks.html?platform=windows&code=PCC

Setup:

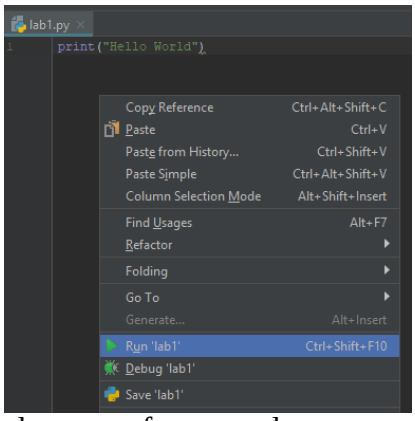
1. Create 'New Project'
2. Now Check 'Existing interpreter' and Add local python interpreter that you've installed, and click Create.



3. Now Right click on the folder and create new python file.



4. After writing your python code right click on the window and run the project.



Python is an interpreted high-level programming language for general-purpose programming.Python is meant to be an easily readable language. Its formatting is visually uncluttered, and it often uses English keywords where other languages use punctuation. Unlike many other languages, it does not use curly brackets to delimit blocks, and semicolons.

Operation	**Syntax**	**Explanation / Example**	**Fill the outputs of all the print statements**
Comment	#	# This is a comment.	
Print (is used to display anything in console window)	print( )	print (10) print (12+6) print (5, end=" ") print ("END")	
Operators	+ plus - minus * multiply ** power / divide // divide and floor % modulus	print(5**2) print (5//2) print (5/2) print (5%2)	
Variables (Python automatically guess the data type. There is no need to explicitly define data type in python)	x=5 y, z = 3.14, 17.2	right side is evaluated first and then assigned to left side with corresponding values print (x + y) print(type(x)) print(type(y))	
Strings	a="py" b='charm'	both single and double quotes worksexactly same print (a + b) print (a, b) print (a*5) print ("hello'\"world\" ")	
Operation	**Syntax**	**Explanation / Example**	**Fill the outputs**

			of all the print statements
Lists List in memory stores references to objects. Each memory location is a pointer to an object. There is no obligation of similardata types	x=int (input (“Enter a number”) y = 3.14 z = "HELLO" li = [x, y, z, 4]	input ( ) function always input string, int ( ) function is used to convert string to int print(li)	
List Indexing # From left to right: 0 → 1 → 2 # From right to left: -1 → -2 → -3	Q = li [2] [-4]	print (Q)	
List Slicing list [start: end: step size]start is inclusive and end is exclusive defaults values are list [0 : end : 1]	R = li [1:3]	print (li [1:3]) print (li [0:4:2]) print (li [:]) print (li [0:]) print (li [:3]) print (li [2] [1:4])	
Copy The assignment copies the reference to the original list while slicing creates a new list	w = [1, 2, 3, 4] x = w y = w [:]	x [0] = 6 y [1] = 9 print(w)	
Indentation	x =2 if x==10: print("inside”) print("inside") print("outside")	Whitespace (spaces and tabs) at the beginning of the logical line is used to determine the indentation level of the logical line, which in turn is used to determine the grouping of statements.	

Boolean operations	< (less than)
	>(greater than)
	<=(less than/equal to)
	>=(greaterthan/equal)
	== (equal to)
	! = (not equal to)
	not (Boolean NOT)
	and (Boolean AND)
	or (Boolean OR)
		if not x:
		if x==2 and y>4:
		print (2==4)	

Operation	**Syntax**	**Explanation / Example**	**Fill the outputs**

			of all the print statements

if
If the Boolean expression evaluates to true, then the if block of code will be executed
	number = 23
	if number == 24:
	print ('equal')
	elif number<24:
	print ('less')
	else:
	print ('greater')
		If statement does not include brackets and ends with colons. The if block is determine by indentation level	

while Repeats a statement or group of statements while a given condition is true	number = 23 while number<30: print(number) number+=1		

for
Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable
	for i in [2,3,4,1,5]:
	print(i)
	for j in range(1,9,2):
	print(j)
		Range function is used to generate a list of numbers, which is generally used to iterate over with for loops	

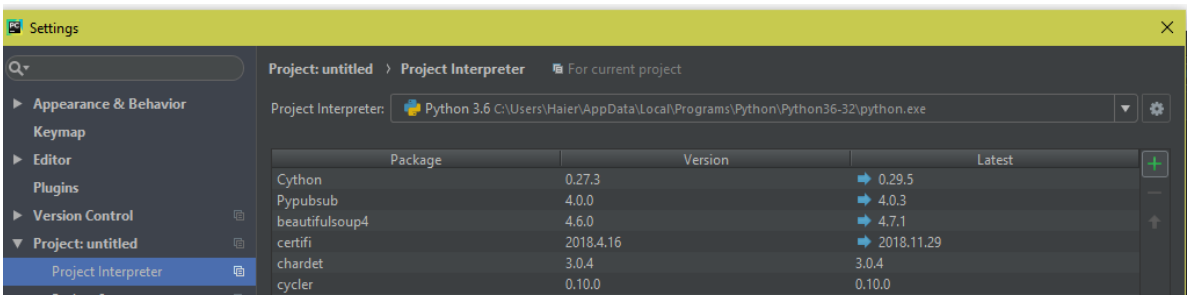
Functions
A function is a block of code which only runs when it is called
	def max (x, y):
	if x > y:
	return x
	else:
	return y
	m=max (3,5)
	print(m)		

OpenCV:

Open Source **C**omputer **V**ision Library (**OpenCv**)is an open source software library written in C++ for machine learning and computer vision applications.

To install packages in PyCharm Click, Files > Settings > Project: Name > Project Interpreter

Now click on '+' icon to install new packages



Now search for following packages and Install them by clicking 'Install Package'

- opencv-python
- numpy // Used for numerical operations
- matplotlib // Used to plotting graphs

Why OpenCv?

OpenCv is comprehensive collection of more than 2500 machine learning and computer vision algorithms that can be used from something as simple detecting faces in images to project augmented reality overlaid with scenery.

Another area in which OpenCv excels is its superior support for multiple interfaces and all the major operating systems as compared to other solutions. OpenCv supports Windows, Linux, OS X and Android and provides interfaces for C, C++, Python, Java and MATLAB.

Some of the applications that can be accomplished easily with OpenCv are: identifying objects, tracking camera movements, stitching images together, finding similar images in a database using an image, face detection and tracking moving objects in a video feed etc.

Some Useful Commands:

1. To slice a 2D array:

   x = y [row_start: row_ end, col_start: col_end]

2. To create a 2D array of zeros using NumPy:

   my_array = **numpy.zeros ((row, columns), dtype=numpy.uint8)**

3. To create a 2D array of ones using NumPy:

   my_array = **numpy.ones ((row, columns), dtype=numpy.uint8)**

4. To check the size of a 2D array: size = **numpy.shape(my_array)**

5. To join a sequence of arrays along an existing axis:

   F = **np.concatenate ((a, b), axis=0);**

6. To assemble an array from nested lists of blocks.

img = np.block **([[np.ones ((2, 2)), np.zeros ((2, 3))], [np.zeros ((2, 2)), np.ones ((2, 3))]])**

   Note: all the input array dimensions except for the concatenation axis must match exactly

7. Reading an image using OpenCv: my_image = **cv2.imread("test_image.jpg",0)**

   The second argument determines whether the image is read as a grayscale image or a colored image. **0** is used for reading an image as grayscale and while **1** is used for reading in color. If no argument is passed then the image is read as is.

8. Displaying an image using OpenCv: **cv2.imshow ("Title of the window", my_image**).

Two more commands that need to be used while displaying an image are: **cv2.waitKey(x) cv2.destroyAllWindows ().** The waitKey () function waits for a key being pressed for x number of milliseconds. If **0** is passed to waitKey () as an argument, it will wait indefinitely for a key press. **cv2.destroyAllWindows ()** closes are the open image windows.

9. Writing an image to disk: **cv2.imwrite("image_name.jpg", my_image)**

Lab Tasks:

1:

x = [[1, 2, 3, 4, 5], [21, 22, 23, 24, 25], [31, 32, 33, 34, 35]]

- Write python code using python indexing and slicing for the following output. Use only one print statement for each output.
  i. [24, 25]
  ii. [3,4]
  iii. [32]
  iv. [21,23,25]
- Declare y = [0, 0, 0], now using for loop write average of first list in list 'x' on first index of list y and so on. The print(y) should give the following output
  - [3.0, 23.0, 33.0] // average of [1, 2, 3, 4, 5] = 3.0

2:

Write python code to count the number of strings from the list where the string length is 2 or more and the first and last character are same from a given list of strings**.**
x=['abc','1121', 'mnnm' , 'aba'] // expected output of this stringcount=3

3:

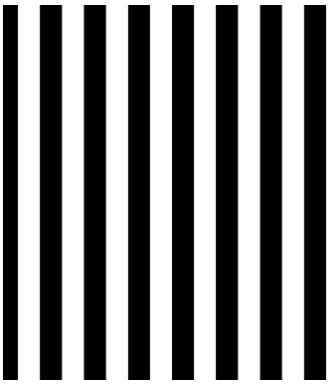
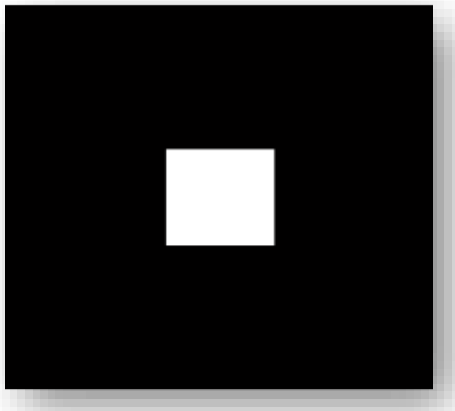
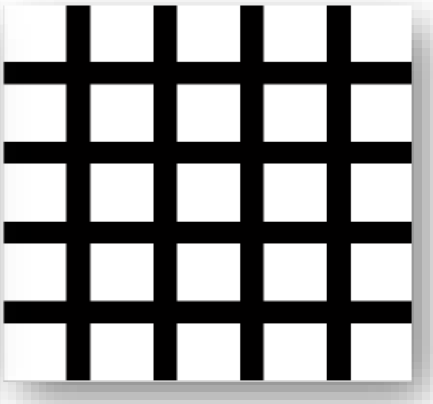
Write a function which takes a list as input argument and generate a new list after removing the 0^{th}, 3^{th} and 5^{th}index value of list.

x=['apple', 'banana', 'mango', 'orange', 'peach' , 'grapes']

Output of this function:x=['banana', 'mango', 'peach']

4:

Write 3 different Python functions that can create the images given below. Code them in such so that the size of the image itself, size of boxes, size of lines and number of horizontal and vertical lines are entered by the user.

Home Tasks:

1: Read any image that you want and save it in gray scale.Now flip the image that you have read.(You cannot use built in functions of flipping)

i. Flip image on vertical axis
ii. Flip image on horizontal axis
iii. Flip image on both axis

Write the images to the disk. See the image below.

2:Write a function to create a white image size entered by the user and then create 4 boxes of Black, Blue, Green and Red respectively on each corner of the image as shown below. The size of the colored boxes should be $1/10^{th}$ the size of the image. (**HINT:** the arrays of ones and zeros can be in more than 2 dimensions)

References:

- ✓ https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_setup/py_table_of_contents_setup/py_table_of_contents_setup.html
- ✓ https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_table_of_contents_imgproc/py_table_of_contents_imgproc.html

THINK!!

1. While copying, why x changed w while y didn't?See above.
2. What will be the data type of result of a Boolean operation?
3. What will be the number of dimension of a grayscale image if opened as colored?
4. Is image a list, tuple or an array?
5. What is the difference between cv.write and printing an image?