1주차 수업 정리

Python

* A = 10 : 변수에 값 입력.자료형 자동 지정
* # 한줄 주석, ‘’’여러 줄 주석‘’’, “””여러 줄 주석2”””
* A // B : A/B의 몫
* A \*\* B : A의 B제곱
* A= input(‘뭐든 입력하세요’) :키보드로 입력 받음.반환형 String
* print(a) : a를 출력
* print(‘i:%d’ %(a)) : 형식 지정자. %d자리에 int형 a값이 출력
* print(‘f:{}, b:{}’.format(f,b)) : {}에 순서대로 f,b값 출력
  + {}안에는 index, key가 들어갈 수 있음
  + Index가 입력 된 경우, {1} : b, {0} : f와 같이 해당 Index에 맞는 위치에 출력
  + Key가 입력 된 경우, format(key0:value0, key1:value1)와 같이 format입력.
* deffunction\_name(parameter) :함수 정의
* if, elif, else :조건문
* for iiniterable :반복문
  + iterable의 예시
  + range(min, max, step) : min<= i<max, step 간격으로 증가시키며 반복
  + list :i는 list의 끝까지 list의 요소를 하나씩 반복
  + enumerate(x) :i는 x의 끝까지, x의 (index, value)를 반복.
    - 보통은 for index, value in enumerate(x)와 같이 사용.

Numpy

* np.array([[1,2,3],[4,5,6]], dtype =) : dtype을 가진자료형의 배열 생성.
  + Numpy datatype 종류
    - np.bool\_ : Boolean
    - np.uint8, uint16, uint32, uint64 : unsigned integer
    - np.int8, int16, int32, int64 : integer
    - np.float16, float32, float64 : float 등
* np.arange(min, max, step) : min <= x < max인 step간격의 배열 생성
* np.linspace(min, max, number, endpoint = True)
  + min <= x <=max인 number개수의 등차수열을 생성.
  + Endpoint는 max를 포함할지 안할지 선택. Default = True
* np.zeros(shape, dtype= ) : shape 형태의 초기값 0인 배열
  + shape의 예) (512,512), (1000,1000,3) 등
* np.ones(shape, dtype= ) : shape 형태의 초기값 1인 배열
* np.eye(x, dtype= ) : x\*x의 단위행렬I 생성
* **Slice**
* A[0] : 0번째 Index에 해당하는 요소
* A[min:max:step] : min <= index < max인 step만큼 증가하는 요소
  + 예시)A[0:10:2]을 하면 A[0], A[2], A[4], A[6], A[8]이 해당한다.
* A[A>5] :조건에 맞는 A의 요소
* **행렬 간 곱셈**
* A \* B : 같은 Index를 가진 요소간의 곱
* np.dot(A,B) : A와 B의 행렬 곱.
* np.abs(x) :절댓값.
  + X는 array가 될 수 있음.
* np.sqrt(x) :제곱근
* np.floor(x), np.ceil(x), np.round(x) :각각 내림,올림,반올림
* np.sin(x), np.cos(x), np.tan(x) :각각 sin, cos, tan
  + x는 radian
* np.arcsin(x), np.arccos(x), np.arctan(x) :각각 sin cos tan의 역 삼각함수

**Matplotlib**

* plt.plot(x,y) : 선 그래프 그리기
  + x : x축 요소들 (1,2,3,4) y : y축 요소들 (3,4,5,6)하게 되면 (1,3) (2,4) (3,5) (4,6)을 지나가는 직선
* plt.scatter(x,y) :점 그래프 그리기
* plt.xlabel(‘X name’), plt.ylabel(‘Y name’) :X축 Y축 이름 지정
* plt.title(‘title name’) : plot title 지정
* plt.legend([‘범례1’, ‘범례2’]) : plot 범례 지정.
  + 여러 개의 그래프가 있어야 범례 지정 가능.
* plt.show() :plot출력

**OpenCV**

* cv2.imread(filePath, flag) :파일 경로의 이미지를 flag대로 읽어옴
  + filePath : 이미지를 불러 올 경로.절대경로, 상대경로 모두 가능.
  + Flag :이미지 읽기 플래그 ( Color 형식 )
    - cv2.IMREAD\_GRAYSCALE : grayscale로 이미지를 읽는다.정수값0
      * 단순 흑백이 아니라,각 Pixel이 가진 값이 graylevel하나.
    - cv2.IMREAD\_COLOR(default) : BGR로 이미지를 읽는다.정수값1
    - cv2.IMREAD\_UNCHANGED : 원본 그대로 이미지를 읽는다.정수값-1
      * color와는 다르게,투명 부분을 유지한다. (alpha)
* cv2.imwrite(filePath, img) :파일 경로에 이미지를 저장한다.
  + Img :저장될 이미지가 들어있는 배열 (np.array)
* cv2.imshow(windowName, img) :window에 이미지를 출력한다. (waitKey안해주면 아무것도 안나옴)
  + windowName :이미지가 표시 될 Window 이름
* cv2.cvtColor(img, flag) : 이미지를 flag에 따라,색상을 변경한다.
  + Flag
    - cv2.COLOR\_[color1]2[color2]와 같이 사용.
    - Color 표현 종류
      * GRAY, BGR, HSV, YCrCb, Luv 등
    - 실 사용시 cv2.COLOR\_BGR2GRAY, cv2.COLOR\_BGR2HSV와 같이 사용.
* Img.shape :이미지의 shape 반환
  + (row x col x channel) : row : 행,col :열, channel : 각 픽셀이 가지는 값의 수
    - BGR image의 경우 channel이 3.
  + Img[100:400, 200:400]과 같이 이미지의 일부 행x 열 범위에 접근 가능.
  + Img[:,:,0] 과 같이 이미지의 모든 pixel이 가지는 Blue에 접근 가능 (1 : green, 2:red)
    - Img[0,0,1]과 같이 일부만 접근도 가능.
* cv2.waitKey(ms): ms(mili second)만큼 key입력을 기다린다.
  + 입력이 생긴 경우,입력된 Key를 반환한다.
* cv2.destroyAllWindows() :모든 Window를 파괴한다.
* cv2.add(a, b) : array a와 array b의 일치하는 index간의 합.
  + 최소값 아래의 underflow, 최대값 위의 overflow가 발생하지 않도록 함.
* cv2.subtract(a,b) : array a와 array b의 일치하는 index간의 차
  + 최소값 아래의 underflow, 최대값 위의 overflow가 발생하지 않도록 함.
* cv2.multiply(a,b) : array a와 array b의 일치하는 index간의 곱
  + 최소값 아래의 underflow, 최대값 위의 overflow가 발생하지 않도록 함.
* cv2.divide(a,b) : array a와 array b의 일치하는 index간의 나누기
  + 최소값 아래의 underflow, 최대값 위의 overflow가 발생하지 않도록 함.
* cv2.pow(a, x) : array a 와 값 x의 제곱.
  + A와 x의 datatype이 일치해야한다.
  + x값이 float이 입력되어야 하는 경우, a의 자료형도float으로 바꿔주어야 함.

\*과제 주의

- 만약 계산 후 값이 소수점이 나오는 경우,그냥 cv2.imshow()를 하게 되면 정상적으로 출력이 되지 않는다.

- 이 때,grayscale = grayscale.astype(np.uint8)과 같이 datatype을 변경하는 방법.

- 미리 datatype이 np.uint8인 배열을 생성하고 값을 저장하는 방법 등이 있다.

(row) y = len(src)

(col) x = len(src[0])