제어지능SW개발

openCV를 활용한 지능형시스템 개발

교수 도경민

컴퓨터전자공학과





- 중간시험 문제 풀이
- 화소 처리
- 영상 변형
- Threshold



1. 다음 프로그램의 실행 결과를 쓰시오(10).

```
a = 1
while a <= 3:
    b = a
    c = a
    print("hi")
    while b < 3:
        while c < 10:
            print(c)
            if c >= 5:
                break
            c += 1
        b += c
    a += 1
print("bye")
```

실행 결과 : hi hi hi bye



2. 오른쪽 실행 결과와 같이 동작하는 프로그램을 완성하시오.(20)

```
import math
Adef prtStar(n):
    \max = (int)((n-1) / 2)
    for line in range(n) :
        blank = max - abs(max - line)
        for j in range(blank):
            print(" ",end='')
        print("*****",end='')
        for b in range (max-blank)*2):
            print(" ",end='')
        print("****")
n = Cint(input("number? "))
prtStar(n)
```

```
number? 5

***** *****

***** *****

***** *****

***** *****
```

중간고사



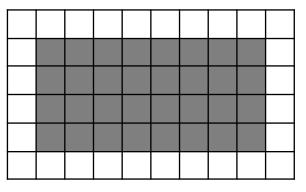
- 3. 다음 물음에 답하시오.(15)
- (1) python에서 전역변수를 처리할 때, 변수 앞에 사용하는 키워드는(5)? global
- (2) 아래 프로그램 실행 결과는?(5)

for n in range(2, 14, 3):
 print(n)

(3) 오른쪽과 같은, 크기가 6x10인 2차원 배열 A의 검은 부분의 원소값을 한꺼번에

100으로 바꾸는 문장을 쓰시오.(5)

A[1:5,1:9] = 100



4. 다음 프로그램에 대해 각 물음에 답하시오.(30)

```
no = 1000.0
while 1:
    try:
        a = float(input("input number? "))
    except:
        print("입력 오류!!")
                                                     В
        continue
    try:
        no = no / a
    except:
                                                     D
        print("조심!!")
    else:
        print(no)
    finally:
        print("bye")
```

(1) 실행 결과에 "입력 오류!!"가 출력되기 위한 입력값의 예를 들고, 프로그램 실행 순서를 쓰시오.(10)

hello

A, B, A

(2) 실행 결과에 "조심!!" 라고 출력되기 위한 입력값의 예를 들고, 프로그램 실행 순서를 쓰시오.(10)

C

A, D, F, A

(3) 실행 결과에 "200" 라고 출력되기 위한 입력값의 예를 들고, 프로그램 실행 순서를 쓰시오.(10)

F

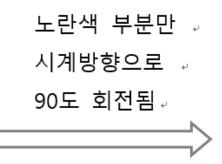
A, C, E, F, A



5. 표1과 같은 2차원 배열을 만들어 출력하고, 표2와 같이 노란색 부분만 시계방향으로 90도 회전한 배열로 만들어서 출력하는 프로그램에 대한 물음에 <u>답하시오</u>.(25)

+ + + **Ξ 1**+

1	8	15	22	29	36	43
2	9	16	23	30	37	44
3	10	17	24	31	38	45
4	11	18	25	32	39	46
5	12	19	26	33	40	47
6	13	20	27	34	41	48
7	14	21	28	35	42	49



7	6	5	4	3	2	1
14	13	12	11	10	9	8
21	20	17	24	31	16	15
28	27	18	25	32	23	22
35	34	19	26	33	30	29
42	41	40	39	38	37	36
49	48	47	46	45	44	43

丑 2↓



```
import numpy as np
SIZE = 7
a = np.arange(1, SIZE*SIZE+1).reshape(SIZE, SIZE)
b = np.arange(1, SIZE*SIZE+1).reshape(SIZE, SIZE)
for r in range(SIZE):
    for c in range(SIZE):
        a[r, c] = \frac{A}{r+1} + c*SIZE
print(a)
for r in range(0, SIZE):
    for c in range(0, SIZE):
        if( (r> (B) 1) and r< (C) 5 ) and (c> (B) 1) and (c< (C) 5) ):
            b[r, c] = a[r, c]
        else:
            if( r<=1 (D) or r>=5 (D) or c<=1 (D) or c>=5 ):
                c1 = (E) 6 - r
                r1 = c
                b[r1, c1] = a[r, c]
print(b)
```

기본 영상 처리



영상 속성

- rowSize, colSize, channels = img.shape
- img[y,x,0] : blue
- img[y,x,1]: green
- img[y,x,2]: red

영상 복사

np.copy(img)

영상 합치기

dst = <u>cv.addWeighted</u>(src1, weight, src2, 1.0- weight, 0.0)

10

cv.cvtColor(src, code[, dst[, dstCn]]) ->dst

src 입력 영상

dst 출력영상

code 변환할 영상 종류 cv.ColorConversionCodes

dstCn 출력영상 채널 수

ex) Grayscale로 변환 grey = <u>cv.cvtColor</u>(img, cv.COLOR_BGR2GRAY)

```
cv::ColorConversionCodes {
enum
        cv::COLOR BGR2BGRA = 0,
        cv::COLOR_RGB2RGBA = COLOR_BGR2BGRA,
        cv::COLOR BGRA2BGR = 1,
        cv::COLOR RGBA2RGB = COLOR BGRA2BGR,
        cv::COLOR BGR2RGBA = 2,
        cv::COLOR_RGB2BGRA = COLOR_BGR2RGBA,
        cv::COLOR RGBA2BGR = 3,
        cv::COLOR BGRA2RGB = COLOR RGBA2BGR,
        cv::COLOR BGR2RGB = 4,
        cv::COLOR_RGB2BGR = COLOR_BGR2RGB,
        cv::COLOR BGRA2RGBA = 5,
        cv::COLOR_RGBA2BGRA = COLOR_BGRA2RGBA,
        cv::COLOR BGR2GRAY = 6,
        cv::COLOR RGB2GRAY = 7,
```



```
cv.absdiff (src1, src2[, dst]) ->dst
cv.add(src1, src2[, dst[, mask[, dtype]]]) ->dst
cv.addWeighted (src1, alpha, src2, beta, gamma[, dst[, dtype]]) ->dst
cv.bitwise_and (src1, src2[, dst[, mask]]) ->dst
cv.bitwise_not (src[, dst[, mask]]) ->dst
cv.bitwise_or (src1, src2[, dst[, mask]]) ->dst
cv.bitwise_xor (src1, src2[, dst[, mask]]) ->dst
```



Scaling

dst=cv.resize(src, dsize[, dst[, fx[, fy[, interpolation]]]])

Translation

dst=cv.warpAffine(src, M, dsize[, dst[, flags[, borderMode[, borderValue]]]])

$$dst(x,y)=src(M_{11}x+M_{12}y+M_{13}, M_{21}x+M_{22}y+M_{23})$$

Rotation

retval=cv.getRotationMatrix2D(center, angle, scale)

$$M = egin{bmatrix} cos heta & -sin heta \ sin heta & cos heta \end{bmatrix}$$

$$\left[egin{array}{cccccc} lpha & eta & (1-lpha) \cdot center. \, x - eta \cdot center. \, y \ -eta & lpha & eta \cdot center. \, x + (1-lpha) \cdot center. \, y \end{array}
ight]$$

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



Affine Transform

retval=cv.getPerspectiveTransform(src, dst[, solveMethod])

$$egin{bmatrix} t_i x_i' \ t_i y_i' \ t_i \end{bmatrix} = exttt{map_matrix} \cdot egin{bmatrix} x_i \ y_i \ 1 \end{bmatrix}$$

$$dst(i) = (x_i', y_i'), src(i) = (x_i, y_i), i = 0, 1, 2, 3$$



Perspective Transformation

retval=cv.getPerspectiveTransform(src, dst[, solveMethod])

dst=cv.warpPerspective(src, M, dsize[, dst[, flags[, borderMode[, borderValue]]]])

$$exttt{dst}(x,y) = exttt{src}\left(rac{M_{11}x + M_{12}y + M_{13}}{M_{31}x + M_{32}y + M_{33}}, rac{M_{21}x + M_{22}y + M_{23}}{M_{31}x + M_{32}y + M_{33}}
ight)$$

retval, dst = cv.threshold(src, thresh, maxval, type[, dst])

threshold type	처리
cv.THRESH_BINARY	threshold 값보다 크면 maxval, 아니면 0
cv.THRESH_BINARY_INV	threshold 값보다 크면 0, 아니면 maxval
cv.THRESH_TRUNC	threshold 값보다 크면 threshold값, 아니면 원래 화소값
cv.THRESH_TOZERO	threshold 값보다 크면 원래 화소값, 아니면 0
cv.THRESH_TOZERO_INV	threshold 값보다 크면 0, 아니면 원래 화소값

스마트폰 등을 이용하여 촬영하여 4장 이상의 256x256 크기의 컬러 사진을 준비한다.

- 1) 1번 영상을 이용하여 마우스로 선택한 사각형을 다음 클릭 위치로 복사하는 프로 그램을 작성한다.
- 2) 2번 영상을 이용하여 마우스로 선택한 부분의 영상을 저장하는 프로그램을 작성한다.
- 3) 3번 영상을 이용하여 transformation 함수 3개 이상 테스트하는 프로그램을 작성한다.
- 4) 1번~3번 3개 영상의 각 컬러를 혼합하여 만든 9개의 영상을 출력하시오.