

2021년 1학기

9

# 제어지능SW개발

openCV를 활용한 지능형시스템 개발

교수 도경민

컴퓨터전자공학과

- 8주차 자기평가 질문
- OpenCV 설치
- Image Reading & Writing
- Image Displaying
- Image Drawing
- Mouse Event 처리

- 문제를 처음 보고 아 어떻게 풀지?? 어떤 문법을 쓰지?? 고민을 했었는데 교수님의 수업을 듣고 나서 손으로 직접 써보면서 정리하고 문법을 생각하지말고 최대한 내가 생각하는 방식으로 해석을 하고나니 자연스럽게 프로그래밍이 되는거 같습니다. 손으로 써보면서 나의 생각을 정리하는 연습을 더 해야겠습니다.
- 저번 강의에서 헛갈리는 부분이 있었는데 이번 강의도 같이 다시 들으면서 해결된 것 같습니다. 감사합니다 !
- 문제를 나의 방법으로 만드는데에는 어느정도 숙지가 됐지만. 만든 나의 방법을 구체화 하는것이 아직 많이 미숙한 것 같습니다. 특히 8주차 과제 1번을 풀면서 많이 느꼈는데, 교수님이 해주시는 방법이 저와 매우 흡사했지만 교수님이 구체화 해주시는 부분부터 이해도가 떨어졌습니다. 더 많은 연습이 필요한 것 같습니다.

- 교수님 수업을 듣고 생각하고 원하는 결과 값을 만들어내는 것이 재미있어졌습니다. 감사합니다!

```
import numpy as np
```

- ```
a=np.arange(1,82).reshape(9,9)
b=np.arange(1,82).reshape(9,9)
```

- ```
a[1,1]=b[1,1]
a[1,1]=b[1,1]
for i in range(1,8):
    for j in range(1,8):
        a[i , j]=b[ j , i]
        a[j , j]=b[ i , j]
    print(a)
```

8-3번 문제를 이렇게 풀었는데 아무리 생각해도 이해가 안되는 출력이 나와서 질문드립니다

```
[[ 1 2 3 4 5 6 7 8 9]
 [10 65 20 29 38 47 56 65 18]
 [19 12 66 30 39 48 57 66 27]
 [28 13 22 67 40 49 58 67 36]
 [37 14 23 32 68 50 59 68 45]
 [46 15 24 33 42 69 60 69 54]
 [55 16 25 34 43 52 70 70 63]
 [64 17 26 35 44 53 62 62 72]
 [73 74 75 76 77 78 79 80 81]]
```

이러한 출력 나오는데 왜 a[1,1],a[2,2]~a[6,6]에 값이 a[1,7],a[2,7]~a[6,7]이 나오는지 이해가 안됩니다  
i와 j에 값이 1일 때 분명a[1,1]=b[1,1] 일꺼고 1,1이나2,2이나 다 동일한 결과값이 출력이 되어야한다고 생각했는데 나오지 않아 질문 드립니다.

(for i in range(1,8):{a[ i, i]=b[ i, i]} 마지막에 이부분을 추가해서 해결하긴 했지만 궁금해서 질문 드립니다.)

- 매주 난이도가 어려워지는거 같아 중간고사가 밀리게 더 부담스럽게 느껴집니다.
- 교수님께서 올려 주시는 예제 코드들을 읽은 후, 옆에 이 코드가 어떤 방식으로 동작하는지를 적어 보며 학습하여 더욱 체계적으로 코드를 작성할 수 있게 되었습니다.  
특히, 교수님의 프로그램 구현에 대한 필기를 참고하며 공부하는 것은 프로그램의 작동 방식을 혼자 구상하는 데 큰 도움을 주는 것 같습니다. 감사합니다.
- '배열은 앞에 있던 정보들 간의 관계를 따져가지고 배열로 나타나는 것'이 정확히 어떤 것인지 8주차 강의 듣고 이해하지 못했는데 이에 대해서 좀더 설명해 주실수 있을까요?  
한줄로 말씀하셔서 이해가 되지 않습니다.

중간고사는 실기인가요? 필기인가요?

- 코로나 상황으로 인해 중간고사가 10주차로 밀리게 되었습니다. 이에 따라 9주차 까지의 수업내용이 중간고사 시험범위가 되는 것이 맞는지 궁금합니다.

- 이번 주차 강의에 대한 질문은 없습니다.  
개인적인 질문을 하나 드리고 싶습니다. 지금까지의 강의를 통해 파이썬의 기본을 배웠다고 생각합니다 배우다 보니 파이썬의 흥미를 느껴 조금 더 깊게 배워보고 싶다는 생각이 들었습니다. 혹시 좀 더 심화된 내용을 배우고 싶다면 방법이 있을까 하여 교수님께 질문드립니다.
- 교수님의 방법대로 과제를 진행하니 순조롭게 됐습니다. 감사합니다.
- 배열을 알것 같으면서도 머릿속에서 완벽히 정리가 안되는 것 같습니다. 직접 노트에 필기하면서 정리해보니 조금씩 습득하고 있는것 같습니다.
- 시험공부를 하기위해 1주차부터 다시 예제를 풀어보았는데 연습장에 써서 하니까전보다 발전된 모습을 볼 수 있었습니다.




제어지능SW개발

# OpenCV



```
> python -m pip install opencv-python
```

```
PS C:\00.Caput\03.IDU\23.2021\02.제어지능SW\source\week9> python -m pip install opencv-python
```

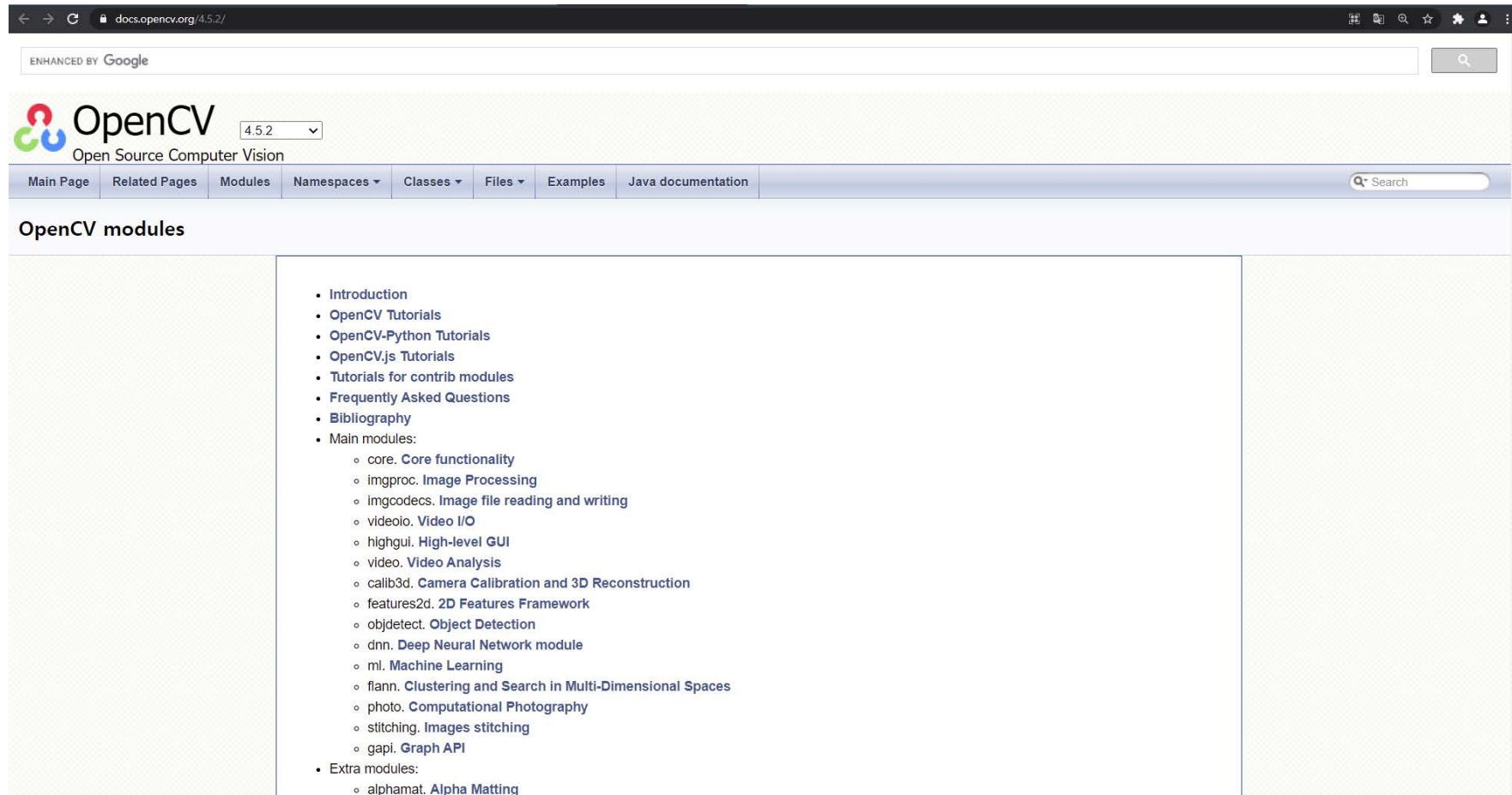
Collecting opencv-python  
Downloading opencv\_python-4.5.1.48-cp38-cp38-win\_amd64.whl (34.9 MB)  
 | 28.4 MB 26 kB/s eta 0:04:07

WARNING: You are using pip version 20.0.2; however, version 20.1 is available.  
You should consider upgrading via the  
'C:\Users\W????\AppData\Local\Programs\Python\Python37\python.exe -m pip install --  
upgrade pip' command.

```
> python -m pip install --upgrade pip
> python -m pip install opencv-python
```



<https://docs.opencv.org/>



# 영상 파일 reading & writing

10


- `retval=cv.imread(filename[, flags])`

**filename**      Name of file to be loaded.

**flags**              Flag that can take values of `cv::ImreadModes`

Windows bitmaps - \*.bmp, \*.dib  
 JPEG files - \*.jpeg, \*.jpg, \*.jpe  
 JPEG 2000 files - \*.jp2  
 Portable Network Graphics - \*.png  
 WebP - \*.webp  
 Portable image format - \*.pbm, \*.pgm, \*.ppm \*.pxm, \*.pnm  
 PFM files - \*.pfm  
 Sun rasters - \*.sr, \*.ras  
 TIFF files - \*.tiff, \*.tif  
 OpenEXR Image files - \*.exr

Enumerator <code>cv::ImreadModes</code>	
<code>cv.IMREAD_UNCHANGED</code>	원영상 형식으로 로드됨(EXIF orientation 무시됨)
<code>cv.IMREAD_GRAYSCALE</code>	single channel grayscale 이미지로 변환되어 로드됨
<code>cv.IMREAD_COLOR</code>	3 channel BGR color 이미지로 변환되어 로드됨
<code>cv.IMREAD_ANYDEPTH</code>	입력 영상이 해당하는 해당도를 가진 경우에는 16-bit/32-bit 이미지로, 그렇지 않으면, to 8-bit 이미지로 변환되어 로드됨
<code>cv.IMREAD_ANYCOLOR</code>	임의의 가능한 칼라 형식으로 변환되어 로드됨
<code>cv.IMREAD_LOAD_GDAL</code>	gdal driver를 사용하여 영상을 로드함
<code>cv.IMREAD_REDUCED_GRAYSCALE_2</code>	single channel grayscale 이미지로 변환되어 로드되며, 영상 크기가 1/2로 줄어듦
<code>cv.IMREAD_REDUCED_COLOR_2</code>	3 channel BGR color 이미지로 변환되어 로드되며, 영상 크기가 1/2로 줄어듦
<code>cv.IMREAD_REDUCED_GRAYSCALE_4</code>	single channel grayscale 이미지로 변환되어 로드되며, 영상 크기가 1/4로 줄어듦
<code>cv.IMREAD_REDUCED_COLOR_4</code>	3 channel BGR color 이미지로 변환되어 로드되며, 영상 크기가 1/4로 줄어듦
<code>cv.IMREAD_REDUCED_GRAYSCALE_8</code>	single channel grayscale 이미지로 변환되어 로드되며, 영상 크기가 1/8로 줄어듦
<code>cv.IMREAD_REDUCED_COLOR_8</code>	3 channel BGR color 이미지로 변환되어 로드되며, 영상 크기가 1/8로 줄어듦
<code>cv.IMREAD_IGNORE_ORIENTATION</code>	EXIF's orientation flag에 따라 회전하지 않음.

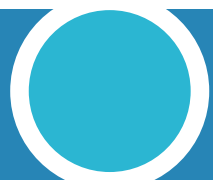
 1loadImage.py

- `retval=cv.imwrite(filename, img[, params])`

filename      Name of the file.

img            (Mat or vector of Mat) Image or Images to be saved.

params        Format-specific parameters encoded as pairs (paramId\_1, paramValue\_1, paramId\_2, paramValue\_2, ... .) see `cv::ImwriteFlags`



- `cv2.imshow(winname, mat)`
- `cv2.namedWindow(winname[, flags])`
- `cv2.destroyWindow(winname)`
- `cv2.destroyAllWindows()`
- `cv2.moveWindow(winname, x, y)`
- `cv2.resizeWindow(winname, width, height)`

- `cv2.line(img, pt1, pt2, color[, thickness[, lineType[, shift]]])`
- `cv2.rectangle(img, pt1, pt2, color[, thickness[, lineType[, shift]]])`
- `cv2.circle(img, center, radius, color[, thickness[, lineType[, shift]]])`
- `cv2.clipLine(imgRect, pt1, pt2)`
- `cv2.ellipse(img, center, axes, angle, startAngle, endAngle, color[, thickness[, lineType[, shift]]])`
- `cv2.ellipse(img, box, color[, thickness[, lineType]])`
- `cv2.fillConvexPoly(img, points, color[, lineType[, shift]])`
- `cv2.fillPoly(img, pts, color[, lineType[, shift[, offset]]])`



- `cv2.line(img, pt1, pt2, color[, thickness[, lineType[, shift]])`

<b>img</b>	Image.
<b>pt1</b>	First point of the line segment.
<b>pt2</b>	Second point of the line segment.
<b>color</b>	Line color.
<b>thickness</b>	Line thickness.
<b>lineType</b>	Type of the line. See <a href="#">LineTypes</a> .
<b>shift</b>	Number of fractional

enum <a href="#">cv::LineTypes</a>	
FILLED Python: cv.FILLED	
LINE_4 Python: cv.LINE_4	4-connected line
LINE_8 Python: cv.LINE_8	8-connected line
LINE_AA Python: cv.LINE_AA	antialiased line



- `cv2.ellipse(img, center, axes, angle, startAngle, endAngle, color[, thickness[, lineType[, shift]])`

<b>img</b>	Image.
<b>center</b>	Center of the ellipse.
<b>axes</b>	Half of the size of the ellipse main axes.
<b>angle</b>	Ellipse rotation angle in degrees.
<b>startAngle</b>	Starting angle of the elliptic arc in degrees.
<b>endAngle</b>	Ending angle of the elliptic arc in degrees.
<b>color</b>	Ellipse color.
<b>thickness</b>	Thickness of the ellipse arc outline, if positive. Otherwise, this indicates that a filled ellipse sector is to be drawn.
<b>lineType</b>	Type of the ellipse boundary. See <a href="#">LineTypes</a>
<b>shift</b>	Number of fractional bits in the coordinates of the center and values of axes.

- `cv2.fillConvexPoly(img, points, color[, lineType[, shift]])`

<b>img</b>	Image.
<b>points</b>	Polygon vertices.
<b>color</b>	Polygon color.
<b>lineType</b>	Type of the polygon boundaries. See <a href="#">LineTypes</a>
<b>shift</b>	Number of fractional bits in the vertex coordinates.

- `cv2.fillPoly(img, pts, color[, lineType[, shift[, offset]]])`

<b>img</b>	Image.
<b>pts</b>	Array of polygons where each polygon is represented as an array of points.
<b>color</b>	Polygon color.
<b>lineType</b>	Type of the polygon boundaries. See <a href="#">LineTypes</a>
<b>shift</b>	Number of fractional bits in the vertex coordinates.
<b>offset</b>	Optional offset of all points of the contours.



- `cv2.putText(img, text, org, fontFace, fontScale, color[, thickness[, lineType[, bottomLeftOrigin]]])`
  - lineType
    - 8 (or omitted) - 8-connected line.
    - 4 - 4-connected line.
    - CV\_AA - antialiased line.
  - fontFace
    - CV\_FONT\_HERSHEY\_SIMPLEX (normal size sans-serif font)
    - CV\_FONT\_HERSHEY\_PLAIN (small size sans-serif font)
    - CV\_FONT\_HERSHEY\_DUPLEX (normal size sans-serif font)
    - CV\_FONT\_HERSHEY\_COMPLEX (normal size serif font)
    - CV\_FONT\_HERSHEY\_TRIPLEX (normal size serif font)
    - CV\_FONT\_HERSHEY\_COMPLEX\_SMALL
    - CV\_FONT\_HERSHEY\_SCRIPT\_SIMPLEX (hand-writing style font)
    - CV\_FONT\_HERSHEY\_SCRIPT\_COMPLEX

 3makeImage.py



- cv2.clipLine(imgRect, pt1, pt2)

**imgSize**

Image size. The image rectangle is Rect(0, 0, imgSize.width, imgSize.height) .

**pt1**

First line point.

**pt2**

Second line point.



4clipline.py

- `cv2.waitKey([delay])` (return int)
- `cv.SetMouseCallback(windowName, onMouse, param=None)`
- `cv.CreateTrackbar(trackbarName, windowName, value, count, onChange)`
- `cv2.getTrackbarPos(trackbarname, winname)`
- `cv2.setTrackbarPos(trackbarname, winname, pos)`

- CvMouseCallback

cv.EVENT_MOUSEMOVE	마우스 움직임
cv.EVENT_LBUTTONDOWN	왼쪽 버튼 누름
cv.EVENT_RBUTTONDOWN	오른쪽 버튼 누름
cv.EVENT_MBUTTONDOWN	중앙 버튼 누름
cv.EVENT_LBUTTONUP	왼쪽 버튼 놓음
cv.EVENT_RBUTTONUP	오른쪽 버튼 놓음
cv.EVENT_MBUTTONUP	중앙 버튼 놓음
cv.EVENT_LBUTTONDBLCLK	왼쪽 버튼 더블 클릭
cv.EVENT_RBUTTONDBLCLK	오른쪽 버튼 더블 클릭
cv.EVENT_MBUTTONDBLCLK	중앙 버튼 더블 클릭
cv.EVENT_MOUSEWHEEL	앞(양수), 뒤 (음수) 스크롤
cv.EVENT_MOUSEHWHEEL	오른쪽(양수), 왼쪽(음수) 스크롤



6drawPicasso.py

- 1) 예쁜 두가지 색의 바둑 무늬 패턴을 그리시오.
- 2) 최소 5가지 이상의 색을 사용한 멋진 패턴을 만드시오,

- 1) 사진 위에 마우스를 클릭할 때마다 임의의 크기와 색상의 원이 그려지는 프로그램을 작성하시오.
- 2) 사진 위에 마우스 클릭한 위치에 점을 찍고 화소 옆에  $(x,y)$  형태로 위치를 표시하는 프로그램을 작성하시오.
- 3) 9주차에서 학습한 OpenCV 기능을 이용하여 자신만의 멋진 프로그램을 작성하시오.



## 연습 9-3

- Circles.jpg 를 멋지게 색칠해 주세요