|  |  |
| --- | --- |
| 교육 제목 | **딥러닝 전이학습** |
| 교육 일시 | 2021년 12월 13일 월요일 |
| 교육 장소 | 영우글로벌러닝 2층 |
| **교육 내용** | |
| 오전 | 1. 이미지 읽기    1. 우선 openCV모듈을 import합니다.:       * >>> import cv2    2. cv2.imread() 함수를 이용하여 이미지 파일을 읽습니다. 이미지 파일의 경로는 절대/상대경로가 가능합니다.       * >>> img = cv2.imread('lena.jpg', cv2.IMREAD\_COLOR)    3. cv2.imread(fileName, flag)    4. 이미지 파일을 flag값에 따라서 읽어들입니다.    5. Parameters:       * fileName (str) – 이미지파일의 경로       * flag (int) – 이미지 파일을 읽을 때의 Option.       * Returns:       * image객체 행렬       * Return type:       * numpy.ndarray    6. 이미지 읽기의 flag는 3가지가 있습니다. 2. cv2.IMREAD\_COLOR : 이미지 파일을 Color로 읽어들입니다. 투명한 부분은 무시되며, Default값입니다. 3. cv2.IMREAD\_GRAYSCALE : 이미지를 Grayscale로 읽어 들입니다. 실제 이미지 처리시 중간단계로 많이 사용합니다. 4. cv2.IMREAD\_UNCHANGED : 이미지파일을 alpha channel까지 포함하여 읽어 들입니다. 5. Note: 3개의 flag대신에 1, 0, -1을 사용해도 됩니다. 6. img값은 numpy의 ndarray type입니다. numpy는 python에서 수학적 처리를 위한 모듈로 openCV에서도 많이 사용됩니다. img가 어떤 형태의 행렬인지 확인을 해보려면 아래와 같이 입력합니다.    1. >>> img.shape    2. (206, 207, 3)    3. 이미지는 3차원 행렬로 return이 됩니다. 206은 행(Y축), 207은 열(X축), 3은 행과 열이 만나는 지점의 값이 몇개의 원소로 이루어져 있는지를 나타납니다. 위 값의 의미는 이미지의 사이즈는 207 X 206이라는 의미입니다.    4. 그렇다면 3은 어떤 의미일까요. 바로 색을 표현하는 BGR값입니다. 일반적으로 RGB로 많이 나타내는데, openCV는 B(lue), G(reen), R(ed)로 표현을 합니다. 7. 이미지 보기    1. cv2.imshow() 함수는 이미지를 사이즈에 맞게 보여줍니다.       * >>> c22.imshow('image', img)       * >>> cv2.waitKey(0)       * >>> cv2.destroyAllWindows()    2. cv2.imshow(title, image)    3. 읽어들인 이미지 파일을 윈도우창에 보여줍니다.   Parameters:   * 1. title (str) – 윈도우 창의 Title   2. image (numpy.ndarray) – cv2.imread() 의 return값   3. cv2.waitKey() 는 keyboard입력을 대기하는 함수로 0이면 key입력까지 무한대기이며 특정 시간동안 대기하려면 milisecond값을 넣어주면 됩니다.   4. cv2.destroyAllWindows() 는 화면에 나타난 윈도우를 종료합니다. 일반적으로 위 3개는 같이 사용됩니다.   5. Sample Code      + import cv2      + fname = 'lena.jpg'      + original = cv2.imread(fname, cv2.IMREAD\_COLOR)      + gray = cv2.imread(fname, cv2.IMREAD\_GRAYSCALE)      + unchange = cv2.imread(fname, cv2.IMREAD\_UNCHANGED)      + cv2.imshow('Original', original)      + cv2.imshow('Gray', gray)      + cv2.imshow('Unchange', unchange)      + cv2.waitKey(0)      + cv2.destroyAllWindows() |
| 오후 | 1. import cv2    * img = cv2.imread('lena.jpg', cv2.IMREAD\_GRAYSCALE)    * cv2.imshow('image',img)    * k = cv2.waitKey(0)    * if k == 27: # esc key    * cv2.destroyAllWindow()    * elif k = ord('s'): # 's' key    * cv2.imwrite('lenagray.png',img)    * cv2.destroyAllWindow() 2. ## gray scale 영상으로 출력 3. imgGray = cv2.imread('fig/puppy.bmp', cv2.IMREAD\_GRAYSCALE) 4. # plt.imshow(imgGray, cmap='gray') 5. # plt.axis('off') 6. # plt.show() 7. ## 두 개의 영상을 모두 출력    1. imgBGR = cv2.imread('fig/puppy.bmp', cv2.IMREAD\_COLOR)    2. imgRGB = cv2.cvtColor(imgBGR, cv2.COLOR\_BGR2RGB)    3. img = cv2.imread('lena.jpg', cv2.IMREAD\_COLOR)       1. b, g, r = cv2.split(img) # img파일을 b,g,r로 분리       2. img2 = cv2.merge([r,g,b]) # b, r을 바꿔서 Merge    4. # fig, axes = plt.subplots(1, 2)    5. # axes[0].imshow(imgRGB), axes[0].axis('off')    6. # axes[1].imshow(imgBGR), axes[1].axis('off')    7. plt.figure(figsize = (16, 4))    8. plt.subplot(141), plt.imshow(imgBGR), plt.axis('off')    9. plt.subplot(142), plt.imshow(imgRGB), plt.axis('off')    10. plt.subplot(143), plt.imshow(imgGray), plt.axis('off') #psedo color cmap    11. plt.subplot(144), plt.imshow(imgGray, cmap = 'gray'), plt.axis('off') #psudo color cmap    12. plt.show() |