



Day65; 20221207

날짜	@2022년 12월 7일
유형	@2022년 12월 7일
태그	

GitHub - u8yes/AI

You can't perform that action at this time. You signed in with another tab or window. You signed out in another tab or window. Reload to refresh your session. Reload to refresh your session.

<https://github.com/u8yes/ai>

u8yes/AI



1 Contributor 0 Issues 0 Stars 0 Forks

화소(pixel) 처리

영상 화소 밝기 변환

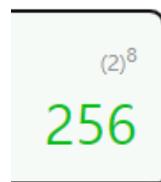
❖ 그레이 스케일 영상

- ✓ 단일 채널의 영상 즉 일반적으로 이해하는 컬러가 아닌 영상을 흑백 영상이라고 하는데 흑백 영상이라는 것은 검은색과 흰색으로 구성된 영상을 의미
- ✓ 디지털 영상처리에서 보통 단일 채널의 영상을 그레이 스케일(gray-scale) 영상이라고 하기도 하고 명암도 영상이라 함
- ✓ 픽셀 값의 그레이 스케일 표현



- ✓ 하나의 픽셀 값은 0~255의 값을 가지는데 0은 검은색을 255는 흰색을 의미
- ✓ 0~255 사이의 값들은 진한 회색에서 연한 회색까지를 나타냄
- ✓ 픽셀 값이 회색의 비율 정도로 표현되고 0~255의 값을 가지는 픽셀들이 모여서 구성된 것이 영상이기 때문에 그레이 스케일 영상이라고도 함

1바이트(=8비트)로 픽셀을 표현할 수 있다. (0 ~ 255, 당연히 0도 포함)



모든 데이터는 정수로 표현된다. = 동일한 자료형이다.

흑백은 이미지 데이터를 2차원 배열(행렬)로 담아서 표현한다.

컬러는 이미지 확대로 인하여 3차원(면행렬)으로 표현된다.

그레이 스케일(gray-scale)

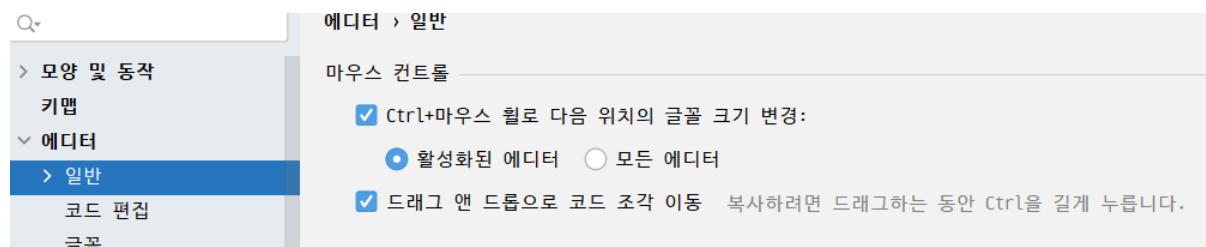
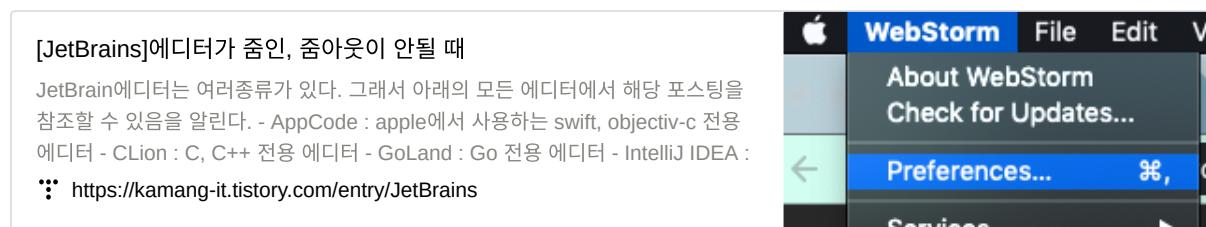
픽셀 값이 회색의 비율 정도로 표현되고 0~255의 값을 가지는 픽셀들이 모여서 구성된 것이 영상이기 때문에 그레이 스케일 영상이라고도 함

scale ⚖ ★★ +

- 명사 (특히 다른 것과 비교해서 본) 규모[범위] (\rightarrow full-scale, large-scale, small-scale)
- 명사 (측정용) 등급 (\rightarrow Richter scale, sliding scale, timescale)
- 동사 격식 (아주 높고 가파른 곳을) 오르다
- 동사 비늘을 치다[벗기다]



파이참 화면 확대, 축소(Ctrl + Mouse Wheel)



chap04_ 01_image_access.py

```
01_image_access.py x
import numpy as np, cv2, time
```

1) 파이썬을 이용해 접근한 방법

```
def pixel_access1(image): # 파일을 이용해 접근한 방법
    image1 = np.zeros(image.shape[:2], image.dtype) # 행과 열의 값이라서 [:2]
    for i in range(image.shape[0]): # 행의 갯수만큼 반복
        for j in range(image.shape[1]): # 열의 갯수만큼 반복
            pixel = image[i, j]
            image1[i, j] = 255 - pixel # 리스트 형태로 다 반전해줌.
    return image1
```

2) *openCV*를 이용한 방법

```
def pixel_access2(image): # openCV를 이용한 방법
    image2 = np.zeros(image.shape[:2], image.dtype)
    for i in range(image.shape[0]):
        for j in range(image.shape[1]):
            pixel = image.item(i, j) # cv2가 제공해주는 기능
            image2.itemset((i,j), 255 - pixel)
    return image2
```

```
def pixel_access3(image):
    lut = [255 - i for i in range(256)] # lookup 테이블 생성 - LookUpTable
    lut = np.array(lut,np.uint8)
    image3 = lut[image]
    return image3
```

```
def pixel_access4(image):
    image4 = cv2.subtract(255, image)
    return image4
```

subtract

미국·영국 [səb'trækt] 영국식 [səb'trækɪt]

(동사)

(수·양을) 빼다 (=take away), (\leftrightarrow add)

6 subtracted from 9 is 3

9에서 6을 빼면 3이다.

영어사전 다른 뜻 1

```
def pixel_access5(image):
    image5 = 255 - image
    return image5
```

```
image = cv2.imread("images/bright.jpg", cv2.IMREAD_GRAYSCALE) # 2차원 회색
if image is None:
    raise Exception("영상 파일 읽기 오류 발생")
```

개발자에게 가장 중요한 것은 가장 빠른 퍼포먼스(수행)에 있다는 것.

performance

미국식[per | fɔ:r'məns] ↗ 영국식[pə | fɔ:məns] ↗

- 1 공연, 연주회
 - 2 (개인의) 연기[연주]
 - 3 실적, 성과
- [영어사전 결과 더보기](#)

```
# 수행 시간 체크
def time_check(func, msg):
    start_time = time.perf_counter() # performance 현재 시간 가져오기
    ret_img = func(image)
    elapsed = (time.perf_counter() - start_time) * 1000
    # 그동안의 시간 비교해보기 # 밀리세컨드 단위 # *1000을 하면 1초로
    print(msg, "수행시간 : %.2f ms" % elapsed) # msg - 전달받은 메세지 출력
    return ret_img
```

```
image1 = time_check(pixel_access1, "[방법1] 직접 접근 방식")
image2 = time_check(pixel_access2, "[방법2] item() 함수 방식")
image3 = time_check(pixel_access3, "[방법3] 루프테이블 접근 방식")
image4 = time_check(pixel_access4, "[방법4] OpenCV 방식")
image5 = time_check(pixel_access5, "[방법5] ndarray 연산 방식")
```

```
C:\ProgramData\Anaconda3\envs\tf_cpu\python.exe |  
[방법1] 직접 접근 방식 수행시간 : 335.12 ms  
[방법2] item() 함수 방식 수행시간 : 41.34 ms  
[방법3] 룩업테이블 접근 방식 수행시간 : 1.88 ms  
[방법4] OpenCV 방식 수행시간 : 0.16 ms  
[방법5] ndarray 연산 방식 수행시간 : 0.15 ms
```

종료 코드 0(으)로 완료된 프로세스

for문 안에 for문은 지양해야한다.

라이브러리를 많이 알아놔야 된다.(달리 구현이 중요)

욕심이 나서 내가 구현하다보면 코드가 길어지기 때문에 버리고 다시 구현하게 되는 상황이 오게 된다.

실무에서 실력을 쌓여가는 방법

고참이 불평을 언급할 때는 수행시간을 보여주면 됨.

```
# 결과 영상 보기  
cv2.imshow("image - original", image)  
cv2.imshow("image1 - directly access", image1)  
cv2.imshow("image2 - item()/itemset", image2)  
cv2.imshow("image3 - LUT", image3)  
cv2.imshow("image4 - OpenCV", image4)  
cv2.imshow("image5 - ndarray", image5)  
cv2.waitKey(0)
```



처음만 원본이며, 이미지1~5까지 수행속도만 본 것이라서 다 동일한 이미지다.

[영어사전]

수행 (遂行)

(이행) fulfill, performance, fulfillment, perform, carry out

영어사전 다른 뜻 2

02_pixel_value.py

```

import numpy as np, cv2

image = cv2.imread("images/pixel.jpg", cv2.IMREAD_GRAYSCALE)
if image is None:
    raise Exception("영상 파일 읽기 오류 발생")

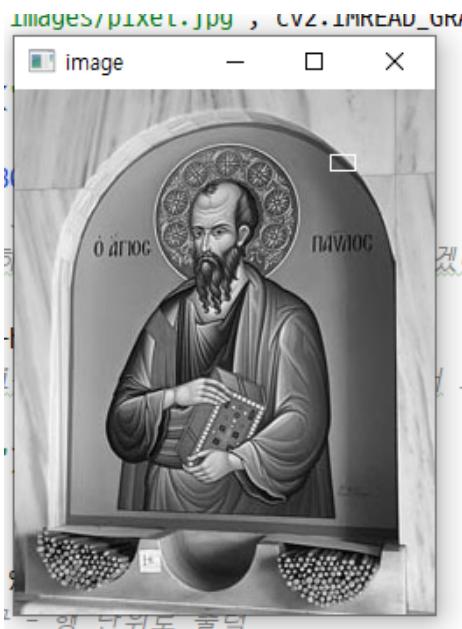
(x, y), (w, h) = (180, 37), (15, 10)
# x는 180위치에 y는 37위치에 시작포인트로 잡아줌.
# x는 15px만큼 접근해가겠다. y는 10px만큼 접근해가겠다는 것.

roi_img = image[y:y+h, x:x+w]
# [주의] 관심영역 roi는 행은 y, 열은 x이다. 바꿔어서 표현.

print("[roi_img] = ")
for row in roi_img:
    for p in row:
        print("%4d" % p, end=' ')
    print() # 줄바꿈 - 행 단위로 출력

cv2.rectangle(image, (x, y, w, h), 255, 1) # 사각형
cv2.imshow("image", image)
cv2.waitKey(0)

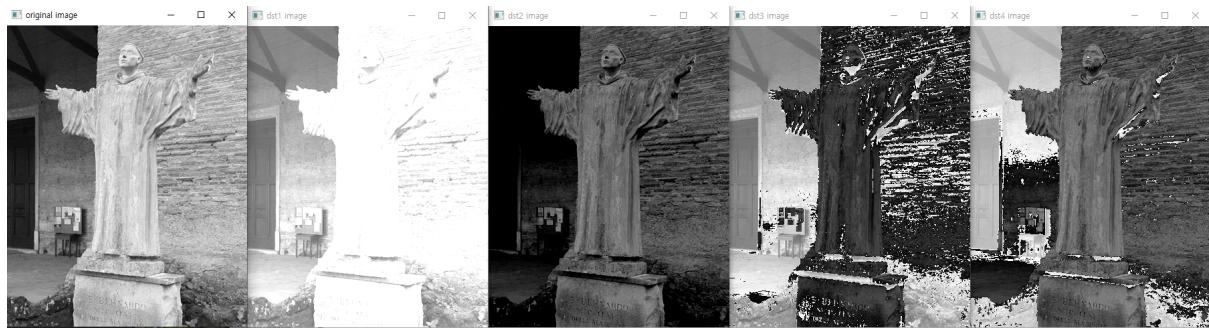
```



```
C:\ProgramData\Anaconda3\envs\tf_cpu\python.exe D:/heaven_dev/\n[roi_img] =\n    56  51  59  66  84 104 154 206 220 208 203 207 205 204 204\n    75  57  53  53  72  71 100 152 195 214 212 201 209 207 205\n    88  76  65  53  51  60  73  96 143 200 219 200 206 204 202\n    91  92  80  63  53  59  59  61  89 144 195 222 205 200 205\n    89  94  90  82  63  54  51  56  65  92 149 203 223 209 196\n    89  91  90  89  84  64  54  55  51  56  94 140 208 223 203\n    91  86  84  85  97  86  72  59  50  53  66  81 148 211 216\n    92  86  85  88  92  95  88  70  55  53  59  64  89 155 211\n    88  85  86  90  87  87  89  86  72  56  50  53  59  88 175\n    87  85  86  88  87  84  86  90  86  70  53  44  51  56 111
```

03_bright_dark.py

```
import numpy as np, cv2\n\nimage = cv2.imread("images/bright.jpg", cv2.IMREAD_GRAYSCALE)\nif image is None:\n    raise Exception("영상 파일 읽기 오류 발생")\n\n# 1) OpenCV 함수 이용\ndst1 = cv2.add(image, 100) # 영상 밝게(+) saturation 방식\ndst2 = cv2.subtract(image, 100) # 영상 어둡게(-)\n\n# 2) numpy array 이용\ndst3 = image + 100 # 영상 밝게 modulo 방식\ndst4 = image - 100 # 영상 어둡게\n\ncv2.imshow("original image", image)\ncv2.imshow("dst1 image", dst1)\ncv2.imshow("dst2 image", dst2)\ncv2.imshow("dst3 image", dst3)\ncv2.imshow("dst4 image", dst4)\ncv2.waitKey(0)
```



dst1 과 dst3, dst2와 dst4의 차이를 보면 되는데 1,2가 더 명확하게 나온다.

synthesis

미국·영국 [sɪnθəsɪs] 영국식 [sɪnθɛsɪs]

- 1 종합, 통합
- 2 (생물체의 체내에서 일어나는) 합성
- 3 (인위적으로 화학 물질을 만들어 내는) 합성

[▶▶▶ 다른 정의 찾기](#)

04_image_synthesis.py

```

import numpy as np, cv2

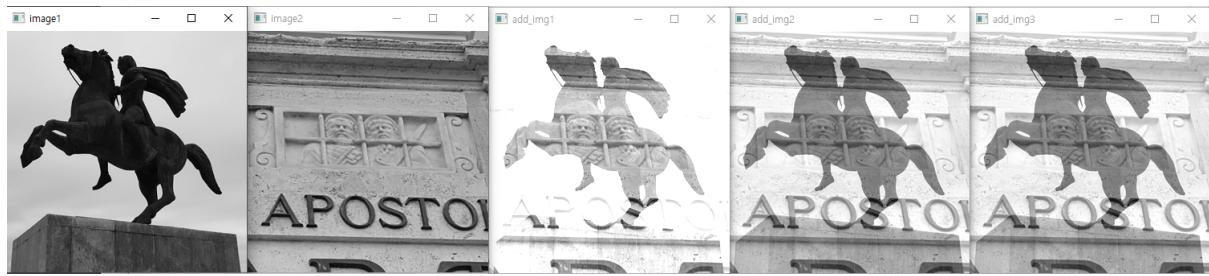
image1 = cv2.imread("images/add1.jpg", cv2.IMREAD_GRAYSCALE) # 알렉산더왕
image2 = cv2.imread("images/add2.jpg", cv2.IMREAD_GRAYSCALE)
if image1 is None or image2 is None:
    raise Exception("영상 파일 읽기 오류 발생")

# 영상 합성
alpha, beta = 0.6, 0.7 # 곱셈 비율
add_img1 = cv2.add(image1, image2)
add_img2 = cv2.add(image1*alpha, image2*beta)
add_img2 = np.clip(add_img2, 0, 255).astype("uint8")
# saturation 처리를 numpy 방식으로 처리 # 최소값 : 0, 최대값 : 255으로 설정.
add_img3 = cv2.addWeighted(image1, alpha, image2, beta, 0)
# addWeighted는 가중치를 더함 # 자체적으로 합성을 수행하는 함수

titles = ['image1','image2','add_img1', 'add_img2', 'add_img3']
for t in titles:
    cv2.imshow(t, eval(t))

cv2.waitKey(0)

```



add_img2(numpy방식), add_img3(cv2)는 같은 결과

멘토링(14:30)

웹 투게더(룰류 아이티) 13년 째 재직중, 윤지환 멘토.

회사소개 회사연혁 주요고객 및 파트너 수상·인증 제품정보 오시는길

오시는길

(주)웹투게더
서울시 구로구 디지털로 30길 31
코오롱빌란트타워 2차 610호

TEL 02.853.0630 FAX 02.853.0634

2호선 구로디지털단지역 2번 출구
7호선 날구조역 2번 출구

5536, 5615, 5616
6004
150-0001, 구로09

TMS
Transportation Management System

Web2gether의 운송관리 시스템은 Container 운송, Bulk 운송에 특화된 웹기반 시스템으로, 복합운송 포워더 관리 시스템(FMS),
창고관리 시스템(WMS)와 유기적으로 연동되며, Global 운송사, 국가간, Agent 등 비지니스 형태에 따라 선택적으로 도입이 가능합니다.

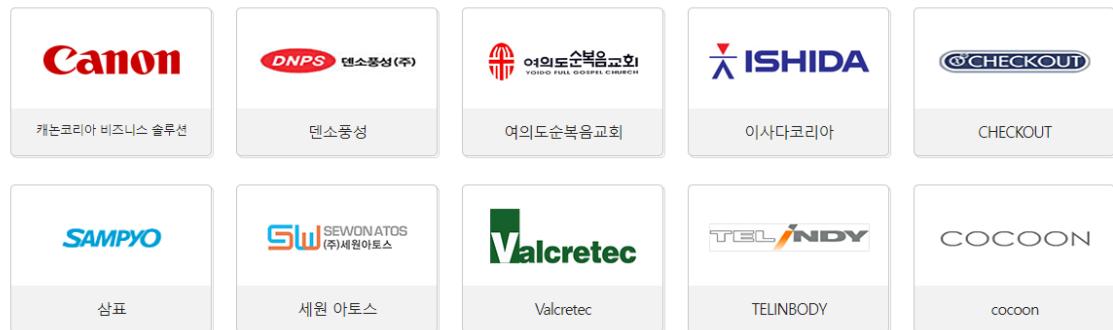
react 개발자도 따로 뽑음.

프론트엔드는 무조건 react를 할 줄 알아야 하고, 자바 개발자도 react를 해야함(프론트 엔드와 협업).

바코드 장비

웹투게더(주)가 국내 공급 라이센스를 받아 프로젝트를 위하여 제공하는 핸디 터미널, RF Scanner, Barcode Scanner, 산업용 PDA 제품들은 일본의 DENSO 제품으로 웹투게더가 10년이상 국내외 유수 기업들의 물류 현장에서 장비의 안정성과 정확

http://www.web2gether.co.kr/barcode_info.html



여의도 순복음 바코드도 제공함.

R 등 AI관련된 머신러닝에서는 수학해야함.

요즘에는 SQL 안 쓰는 추세로 가고 있는 중.

하지만 지금은 SQL 많이 사용함.

PMO자격증

사단법인 피엠전문가협회

Copyright © 2016~2023 사단법인 피엠전문가협회. All rights reserved.

https://www.icpm.or.kr/?act=board&bbs_code=PMO_test&bbs_mode=view&bbs_seq=206



잡플래닛 - 아직하려는 사람

NO.1 기업 정보 플랫폼, 잡플래닛 - 지금 채용 지원하고 취업 축하금 200만원 받자!

대기업, 중견기업, 중소기업, 스타트업까지. 기업 전문 뉴스 컴퍼니타임스, 기업리뷰, 면접후기, 연봉, 채용, 기업 정보를 한 번에!

Jobplanet

▶ <https://www.jobplanet.co.kr/job>

리액트 - 자바스크립트

리액트 (자바스크립트 라이브러리) - 위키백과, 우리 모두의 백과사전

리액트는 싱글 페이지 애플리케이션이나 모바일 애플리케이션 개발에 사용될 수 있다. 대규모 또는 복잡한 리액트 애플리케이션 개발에는 보통 라우팅, API 통신 등의 기능이 요구되는데 리액트에는 기본적으로 제공되지 않기 때문에 추가 라이브러리를 W [https://ko.wikipedia.org/wiki/%EB%A6%AC%EC%95%A1%ED%8A%B8_\(%EC%9E%90%EB%B0%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8_%EB%9D%BC%EC%9D%B4%EB%B8%8C%EB%9F%AC%EB%A6%AC\)](https://ko.wikipedia.org/wiki/%EB%A6%AC%EC%95%A1%ED%8A%B8_(%EC%9E%90%EB%B0%94%EC%8A%A4%ED%81%AC%EB%A6%BD%ED%8A%B8_%EB%9D%BC%EC%9D%B4%EB%B8%8C%EB%9F%AC%EB%A6%AC))



SI업체 중에 보도방 조심해야함. 혼자 나가는지 물어봐야 함(신입인데 혼자 내보내면 무조건 걸려야 함), 팀 단위로 가야함.

- 노트북 사오라고 하는 데가 있음. (예: 1년 뒤에 노트북 비용을 줌)

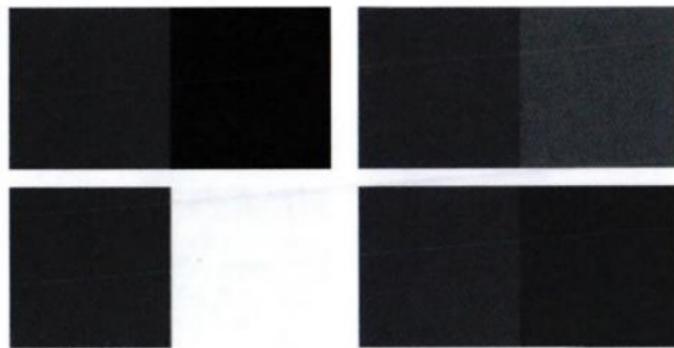
그런 데가 조금 위험.

- 채용 공고가 자주 올라오는 데는 위험함. (**사람이 너무 자주 바뀜**, 혹은 **확장이거나**)
- 연혁이 짧고 너무 적은 인원인 회사(매출 높은데는 월급을 잘 받음, 다만 모든 일을 다하고 엄청 많이 시킴)
- 카카오는 내실을 못 다지고 성장만 해서 일이 터진 경우
- 그렇다고 연혁이 길어도 자기 문화가 강해서 텃새가 심할 수 있음.
- 너무 형식적인 서류면접은 탈락 가능성이 높음. (A회사, B회사, C회사에 동시에 넣을만한 서류는 비교적 떨어지는 편)

끝

❖ 명암 대비

- ✓ 명암 대비(contrast)는 상이한 두 가지 색(밝기)이 경계에서 서로 영향을 미쳐 그 차이가 강조되어 나타나는 현상
- ✓ 다음 그림은 명암 대비의 예를 보인 것으로 각 그림에서 왼쪽 부분은 픽셀 값이 100으로 동일하지만 오른쪽 부분의 다른 밝기로 인해 경계 부분에서 약간 다른 색으로 인식됨



05_contrast

```
4_image_synthesis.py x 05_contrast.py x
import numpy as np, cv2

image = cv2.imread("images/contrast.jpg", cv2.IMREAD_GRAYSCALE)
if image is None:
    raise Exception("영상 파일 읽기 오류 발생")

noimage = np.zeros(image.shape[:2], image.dtype)
# image와 동일한 data type으로 설정 # 더미 영상으로 만들어줌
# noimage에는 0만 담아져있음.

avg = cv2.mean(image)[0] / 2.0 # 평균을 구한 후에 2를 나눔, 그래서 반이 됨.

dst1 = cv2.scaleAdd(image, 0.2, noimage) + 20 # 영상 대비 감소
# 20을 더해줘서 아예 어두워짐 방지해줌.
dst2 = cv2.scaleAdd(image, 2.0, noimage) # 영상대비 증가
dst3 = cv2.addWeighted(image, 0.5, noimage, 0, avg)
# 전체 더 낮게 # 명암 대비 반 감소
dst4 = cv2.addWeighted(image, 2.0, noimage, 0, -avg)
# 전체 2배로 # 명암 대비 증가
```

```
# 영상 띄우기
cv2.imshow("image", image)
cv2.imshow("dst1 - decrease contrast", dst1)
cv2.imshow("dst2 - increase contrast", dst2)
cv2.imshow("dst3 - decrease contrast using average", dst3)
cv2.imshow("dst4 - increase contrast using average", dst4)
cv2.imwrite("dst.jpg", dst)
cv2.waitKey(0)
```



범주형, 연속형으로 정해져있는 시각화가 아닌 결과가 어떤 것에 focus를 맞춰야 한다.

따라서 범주형으로도 연속형으로도 시각화를 사용할 수 있다.

예: hist는 빈도로 가면 범주형, 밀도로 가면 연속형.

히스토그램은 연속형에도 쓰이지만 아래에서는 정확히 빈도를 찾게 되면서 범주형이 됨.

히스토그램

❖ Histogram

- ✓ Histogram(histogram)은 관측값의 개수를 겹치지 않는 다양한 계급으로 표시하는 것
- ✓ 단일 채널 8비트(uint8) 행렬의 Histogram은 부호 없는 8비트 행렬이기에 나타날 수 있는 원소 값의 범위는 0~255이며 한 계급(bin)이 1이 되게 설정하는데 계급 간격은 경우에 따라 조절
- ✓ 직접 계산



마치 파생변수인 것처럼 이해를 넓혀가는 것이 중요하다.

06_calc_histogram_opencv.py

```

import numpy as np, cv2

def calc_hist(image, hsize, ranges=[0, 256]): # 행렬 원소의 1차원 히스토그램 계산
    hist = np.zeros((hsize, 1), np.float32) # 히스토그램 누적 행렬
    gap = ranges[1] / hsize # 계급 간격

    for i in range(image.shape[0]): # 행의 갯수만큼 반복
        for j in range(image.shape[1]): # 열의 갯수만큼 반복
            idx = int(image.item(i, j) / gap)
            hist[idx] += 1

    return hist

image = cv2.imread("images/pixel.jpg", cv2.IMREAD_GRAYSCALE)
if image is None:
    raise Exception("영상 파일 읽기 오류 발생")

```

```

hsize, ranges = [32], [0, 256]
gap = ranges[1] / hsize[0]
ranges_gap = np.arange(0, ranges[1]+1, gap) # arange는 range와 동일한 기능이다.
hist1 = calc_hist(image, hsize[0], ranges)
hist2 = cv2.calcHist([image], [0], None, hsize, ranges) # 범주형
hist3, bins = np.histogram(image, ranges_gap) # 가장 수행시간 효율이 좋은 numpy

print("User 함수: \n", hist1.flatten())
print("OpenCV 함수: \n", hist2.flatten())
print("numpy 함수: \n", hist3) # numpy는 알아서 벡터의 형태로 만들어서 return해준다.

cv2.imshow("image", image)
cv2.waitKey(0)

```



User 함수:

```
[ 97. 247. 563. 1001. 1401. 1575. 1724. 1951. 2853. 3939. 3250. 2549.  
2467. 2507. 2402. 2418. 2727. 3203. 3410. 3161. 2985. 2590. 3384. 4312.  
4764. 3489. 2802. 2238. 1127. 628. 199. 37.]
```

OpenCV 함수:

```
[ 97. 247. 563. 1001. 1401. 1575. 1724. 1951. 2853. 3939. 3250. 2549.  
2467. 2507. 2402. 2418. 2727. 3203. 3410. 3161. 2985. 2590. 3384. 4312.  
4764. 3489. 2802. 2238. 1127. 628. 199. 37.]
```

numpy 함수:

```
[ 97 247 563 1001 1401 1575 1724 1951 2853 3939 3250 2549 2467 2507  
2402 2418 2727 3203 3410 3161 2985 2590 3384 4312 4764 3489 2802 2238  
1127 628 199 37]
```

07_draw_histogram.py

```

histo.py  ss_colorasst.py  ss_calc_histogram_OpenCV.py  cv2_drawing_hist.py
import numpy as np, cv2

def draw_histo(hist, shape=(200, 256)):
    hist_img = np.full( shape, 255, np.uint8)
    cv2.normalize(hist, hist, 0, shape[0], cv2.NORM_MINMAX)
    # normalize - 표준화를 수행, NORM_MINMAX - 정규화를 수행
    gap = hist_img.shape[1]/hist.shape[0]

    for i, h in enumerate(hist):
        x = int(round(i * gap))
        w = int(round(gap))
        roi = (x, 0, w, int(h))
        cv2.rectangle(hist_img, roi, 150, -1)
        cv2.rectangle(hist_img, roi, 0, 1)

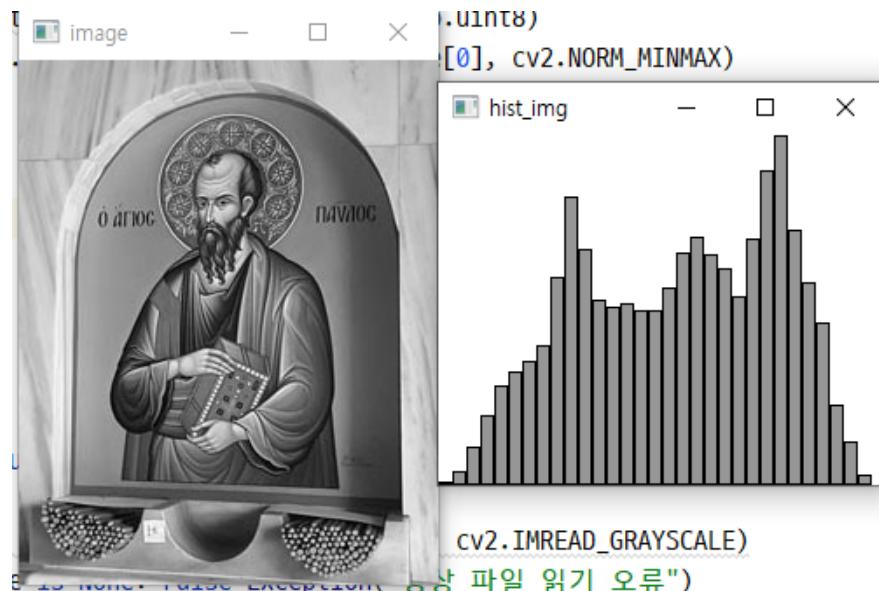
    return cv2.flip(hist_img, 0)
# x축을 기준으로 거꾸로 뒤집어서 반환해줌. 이미지는 0,0이 왼쪽 상단이라서

image = cv2.imread("images/pixel.jpg", cv2.IMREAD_GRAYSCALE)
if image is None: raise Exception("영상 파일 읽기 오류")

hist = cv2.calcHist([image], [0], None, [32], [0, 256])
hist_img = draw_histo(hist)

cv2.imshow("image", image)
cv2.imshow("hist_img", hist_img)
cv2.waitKey(0)

```



8개씩 32등분 돼있음. 이미지의 1등분은 성분이라고 표현할 수 있다.

08_hue_histogram.py

hue saturation value의 약어. 색상(H), 채도(S), 명도(V)로 색을 지정하는 방법을 이용한 디자인 용도에 적합한 프로그램 .

컴퓨터인터넷IT용어대사전 terms.naver.com

HSV (지식백과)

```
import numpy as np, cv2

def make_palette(rows):
    # 리스트 생성 방식
    hue = [round(i * 180 / rows) for i in range(rows)] # hue 값 리스트 계산
    hsv = [[(h, 255, 255)] for h in hue] # (hue, 255,255) 화소값 계산
    hsv = np.array(hsv, np.uint8) # numpy 행렬의 uint8형 변환
    return cv2.cvtColor(hsv, cv2.COLOR_HSV2BGR) # HSV 컬러 -> BGR 컬러

def draw_histo_hue(hist, shape=(200, 256, 3)):
    hsv_palate = make_palette(hist.shape[0])
    hist_img = np.full(shape, 255, np.uint8)
    cv2.normalize(hist, hist, 0, shape[0], cv2.NORM_MINMAX)

    gap = hist_img.shape[1] / hist.shape[0]
    for i, h in enumerate(hist):
        x, w = int(round(i * gap)), int(round(gap))
        color = tuple(map(int, hsv_palate[i][0]))
        cv2.rectangle(hist_img, (x, 0, w, int(h)), color, cv2.FILLED)

    return cv2.flip(hist_img, 0)

image = cv2.imread("images/hue_hist.jpg", cv2.IMREAD_COLOR) # 영상 읽기
if image is None: raise Exception("영상 파일 읽기 오류")

hue_img = cv2.cvtColor(image, cv2.COLOR_BGR2HSV) # BGR 컬러 -> HSV 컬러
hue_hist = cv2.calcHist([hue_img], [0], None, [18], [0,180]) # Hue 채널 히스토그램 계산
hue_hist_img = draw_histo_hue(hue_hist, (200, 360, 3)) # 히스토그램 그래프

cv2.imshow("image", image)
cv2.imshow("hue_hist_img", hue_hist_img)
cv2.waitKey(0)
```



히스토그램

❖ Histogram Stretching

- ✓ 명암도 영상에서 영상이 보기에 선명하고 깨끗해 보이려면 어두운 부분에서 밝은 부분 까지 고루 분포되어 있어야 하는데 그렇지 않고 주로 특정 밝기 부분만 있는 영상은 그림에서 보는 바와 같이 전체적으로 선명하지 않으며 또렷하지도 않음
- ✓ 이런 영상을 히스토그램에서 보면 그림의 오른쪽과 같이 한쪽으로 치우쳐서 히스토그램 분포가 좁은 그래프들로 나타남
- ✓ 히스토그램의 분포가 좁아서 영상의 대비가 좋지 않은 영상들의 화질을 개선할 수 있는 알고리즘이 히스토그램 스트레칭(histogram stretching)

□ 밝은 부분이 많이 분포하는 영상



□ 어두운 부분이 많이 분포하는 영상



보강 수업

Model은 DB와의 연동

MVC 패턴에서 절대 서비스를 빼고 컨트롤러에서 바로 모델로 연결하지 말자.

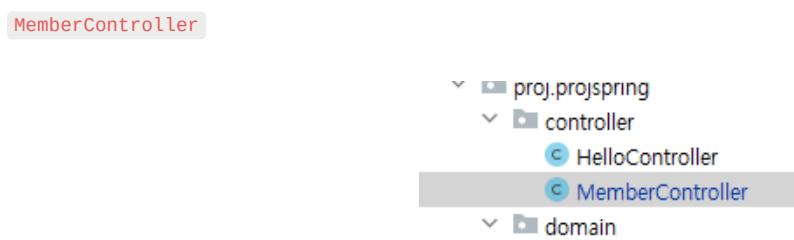
그리면 전체가 변경됨.

Business Logic - Service, Model

members/createMemberForm.html

```
<!DOCTYPE html>
<html xmlns:th="http://www.thymeleaf.org">
<head>
    <meta charset="UTF-8">
    <title>Title</title>
</head>
<body>

    <!-- 데이터 분석 크롤링 때 가장 중요한 것은 class 이름이었음-->
    <div class="container" > <!-- container 적재, 템재-->
        <form action="/members/new" method="post">
            <div class="form-group">
                <label for="name">이름</label>
                <input type="text" id="name" name="name" placeholder="이름을 입력하세요">
            </div>
            <button type="submit">등록</button>
        </form>
    </div>
</body>
</html>
```



```
└ u8yes
@Controller
public class MemberController {

    2 usages
    @Autowired
    private MemberService service;

    └ u8yes
    @GetMapping("/")
    public String member(Model Model){
        return "member";
    }

    └ u8yes
    @GetMapping(value="/members/new")
    public String createForm(){
        return "members/createMemberForm";
    }
}
```

```
└ u8yes
    @PostMapping(value="/members/new")
    public String create(MemberForm form){
        Member member = new Member();

        member.setName(form.getName());

        Long id = service.join(member);
        System.out.println("=> 아이디 : " + id);

        return "redirect:/";
    }

    └ u8yes
    @GetMapping(value="/members")
    public String list(Model model){
        service.findAll();

        return null;
    }
}
```

MemberService

```
package proj.projspring.service;

import proj.projspring.domain.Member;

import java.util.List;

3 usages 1 implementation ▾ u8yes
public interface MemberService {

    1 usage 1 implementation ▾ u8yes
    public Long join(Member member);
    1 usage 1 implementation ▾ u8yes
    public List<Member> findAll();
}
```

MemberServiceImpl

MemberService.java

```
import proj.projspring.repository.MemberRepository;

import java.util.List;

@u8yes
@Service
public class MemberServiceImpl implements MemberService{

    1 usage
    @Autowired
    private MemberRepository repository;
    1 usage  + u8yes
    @Override
    public Long join(Member member) {

        repository.save(member);
        return member.getId();
    }

    1 usage  + u8yes
    @Override
    public List<Member> findAll() {
        return null;
    }
}
```

MemberForm

MemberForm

```
MemberService.java ^ iMemberServiceImpl.java
package proj.projspring.domain;

import lombok.Getter;
import lombok.Setter;

2 usages  + u8yes
@Getter
@Setter
public class MemberForm {
    private Long id;
    private String name;
}
```

The screenshot shows a file structure and a code editor. At the top, there's a tree view of files:

- memberform
- repository (selected)
- MemberRepository (highlighted)
- MemoryMemberRepository
- service

Below the tree view is a code editor window with two tabs:

- emoryMemberRepository.java
- MemberRepository.java (selected)

The code in the editor is:

```
package proj.projspring.repository;

import proj.projspring.domain.Member;

import java.util.List;

public interface MemberRepository {
    Member save(Member member);

    List<Member> findAll();
}
```

```
MemoryMemberRepository.java × idea64.exe.vmoptions × application.properties × Member
package proj.projspring.repository;

import org.springframework.stereotype.Repository;
import proj.projspring.domain.Member;

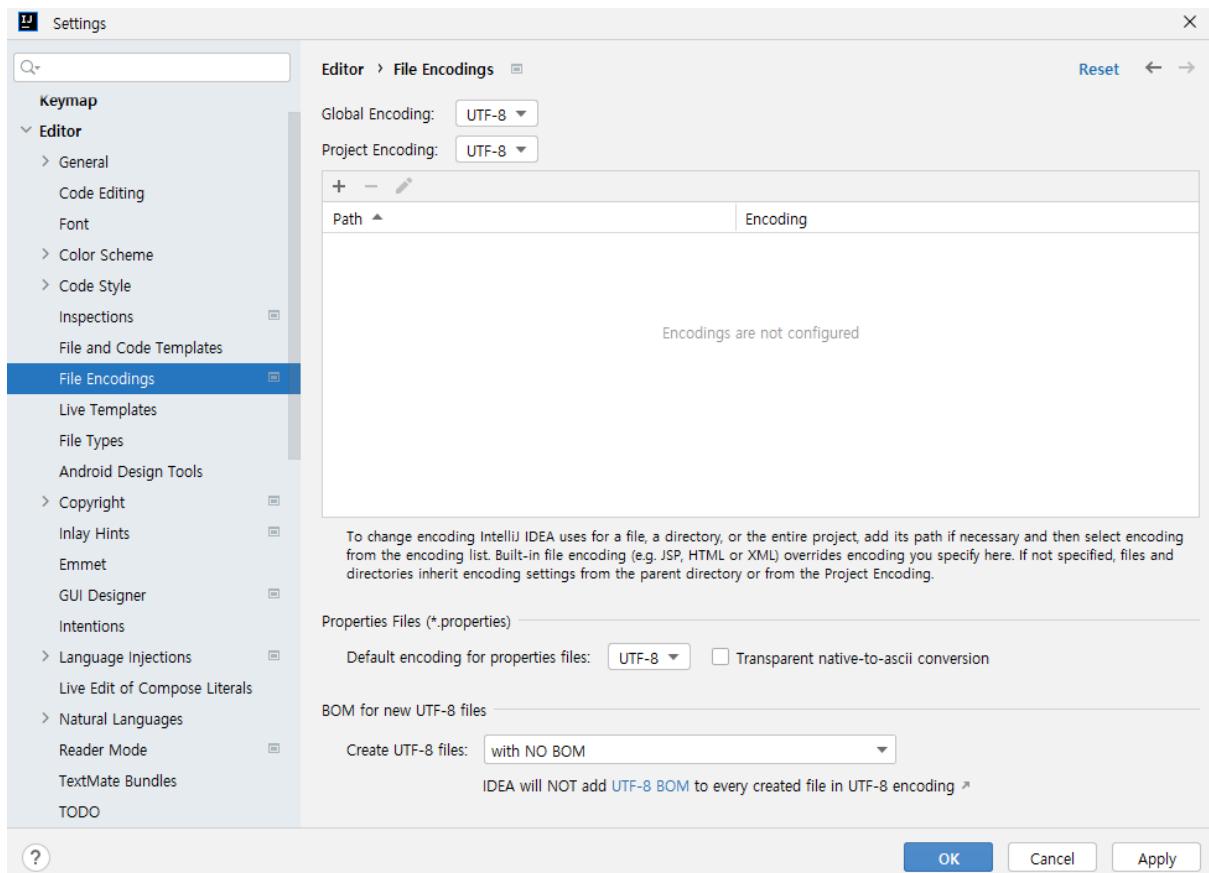
import java.util.HashMap;
import java.util.List;
import java.util.Map;

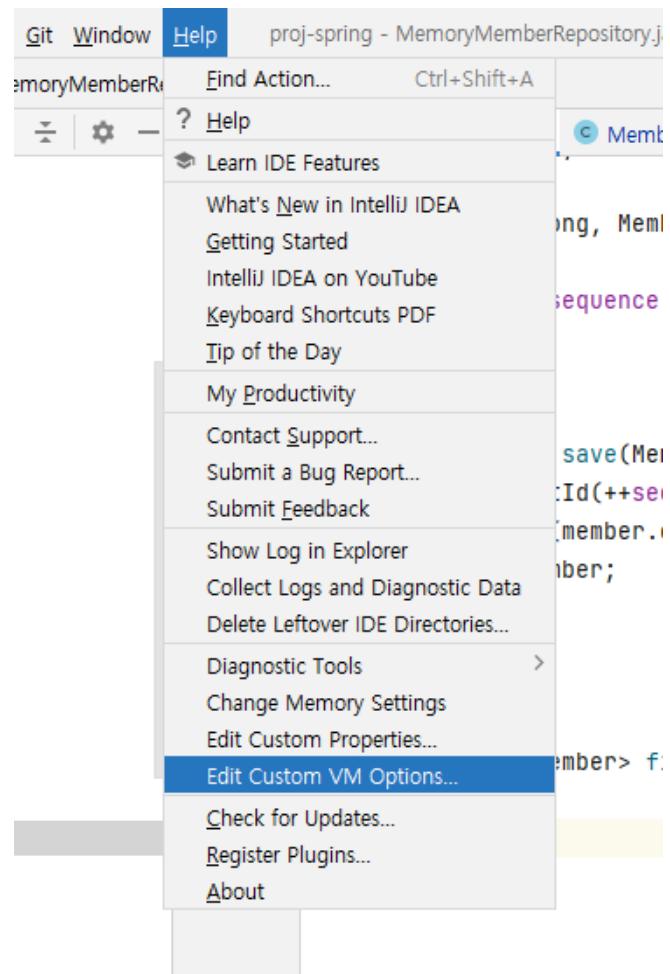
2 usages ± u8yes
@Repository
public class MemoryMemberRepository implements MemberRepository{
    1 usage
    private Map<Long, Member> store = new HashMap<>();
    1 usage
    private long sequence = 0L;

    2 usages ± u8yes
    @Override
    public Member save(Member member) {
        member.setId(++sequence);
        store.put(member.getId(), member); // put - 임시로 저장해줌.
        return member;
    }

    ± u8yes
    @Override
    public List<Member> findAll() { return null; }
}
```

한글 깨질경우





The screenshot shows the 'VM Options' dialog box in IntelliJ IDEA. It displays three entries: '-Xmx2048m', '-Dfile.encoding=UTF-8', and '-Dconsole.encoding=UTF-8'. The third entry is currently selected.

-Xmx2048m
-Dfile.encoding=UTF-8
-Dconsole.encoding=UTF-8

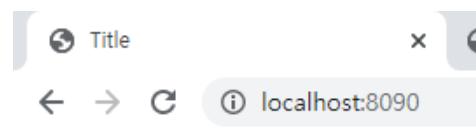
오라클 DB 정보를 알려줄 때

application.properties

The screenshot shows the project structure of a Spring Application named 'ProjSpringApplication'. The 'resources' folder contains 'static' and 'templates' subfolders. 'static' has 'hello-static.html' and 'index.html'. 'templates' has 'members' subfolder containing 'createMemberForm.html', 'hello.html', 'hello-template.html', and 'member.html'. An 'application.properties' file is also present in the resources folder. A 'test' folder is shown at the bottom.

```
server.port = 8090
spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver
spring.datasource.url=jdbc:oracle:thin:@localhost:1521/xe
spring.datasource.username=scott
spring.datasource.password=tiger
```

```
spring.datasource.driver-class-name=oracle.jdbc.driver.OracleDriver
spring.datasource.url=jdbc:oracle:thin:@localhost:1521/xe
spring.datasource.username=scott
spring.datasource.password=tiger
```



SH is very rich

[회원 기능](#)

[회원 가입](#) [회원 목록](#)

