

5주차 - 화소 점처리 예제

- 로고 삽입
- 워터마크 삽입
- QR 코드 삽입

```
In [1]: ### Packages
import cv2
import numpy as np
from matplotlib import pyplot as plt
import os
```

```
In [2]: ### 출력 영상 크기
plt.rcParams["figure.figsize"] = (16,9)
### 한글 표시
plt.rcParams['font.family'] = "Gulim" # 'AppleGothic' in mac
```

▶ 로고 삽입

- 대상 영상에서 로고가 삽입될 위치의 화소값을 로고 이미지가 가진 화소값으로 변경

```
In [3]: ### 영상 읽기
img1 = cv2.imread(r'D:WimageWimg1.jpg')
img_logo = cv2.imread(r'D:WimageWbda_logo.png')
```

```
In [4]: ### 영상 삽입 - 좌측 상단
img_out = img1.copy()
h, w, c = img_logo.shape
img_out[0:h, 0:w, :] = img_logo
```

```
In [5]: ### 영상 출력
img_rgb = cv2.cvtColor(img_out, cv2.COLOR_BGR2RGB)
plt.imshow(img_rgb)
plt.axis('off')
plt.show()
```



▣ 예제

- 우측 하단에 로고 삽입

▶ 워터마크 삽입

- 워터마크 이미지 생성 후 영상 혼합(image blending)

```
In [6]: ### 영상 읽기
img1 = cv2.imread(r'D:\Wimage\Wimg1.jpg')
img_logo = cv2.imread(r'D:\Wimage\Wbda_logo.png')

In [7]: ### 이진화 - grayscale
img_logo_gray = cv2.cvtColor(img_logo, cv2.COLOR_BGR2GRAY)
_, img_logo_th = cv2.threshold(img_logo_gray, 200, 255, cv2.THRESH_BINARY_INV)

In [8]: ### 뽕샘 - 영상의 밝기 감소
img_logo_th_2 = cv2.add(img_logo_th, -240)
img_logo_th_2 = cv2.cvtColor(img_logo_th_2, cv2.COLOR_GRAY2BGR)

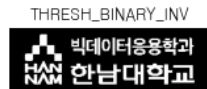
In [9]: ### 출력 영상
img_out = img1.copy()

### 대상 영역 추출
h, w, c = img_logo.shape
img_out_sub = img_out[0:h, 0:w, :]

In [10]: ### 덧셈 - 영상의 밝기 증가
img_out_wm = cv2.add(img_out_sub, img_logo_th_2)

In [11]: ### 영상 출력
titles = ["Logo", "THRESH_BINARY_INV", "뽕샘", "영역 추출", "덧셈"]
images = [img_logo, img_logo_th, img_logo_th_2, img_out_sub, img_out_wm]
for i in range(len(images)):
    img_rgb = cv2.cvtColor(images[i], cv2.COLOR_BGR2RGB)
    plt.subplot(1, 5, i+1)
```

```
plt.imshow(img_rgb)
plt.title(titles[i])
plt.axis('off')
plt.show()
```



In [12]:

```
### 영상 삼입 - 좌측 상단
img_out[0:h, 0:w, :] = img_out_wm
```

```
### 영상 출력
img_rgb = cv2.cvtColor(img_out, cv2.COLOR_BGR2RGB)
plt.imshow(img_rgb)
plt.axis('off')
plt.show()
```



▶ QR 코드

- QR 코드(영어: QR code, Quick Response code)는 컴퓨터가 만든 흑백 격자무늬 패턴 코드로, 정보를 나타내는 매트릭스 형식의 이차원 코드
- 한국, 일본, 중국, 영국, 미국 등에서 많이 사용
- 텐소 웨이브의 등록 상표 'Quick Response'에서 유래
- 숫자 외에 문자의 데이터를 저장
 - 숫자 최대 7,089 자
 - 영문자와 한글은 최대 4,296 자
 - 한자 1,817 자
- <https://www.qrcode.com/ko/>
 - 버전은 1~40으로 구성
 - 버전1(21×21cell)로 시작하여 가로/세로 각각 4cell씩 늘어,버전40(177×177cell)로 설정



■ QR 코드 생성

- Google API
 - https://developers.google.com/chart/infographics/docs/qr_codes
 - <https://chart.googleapis.com/chart?cht=qr&chs=200x200&chl=http://bigdata.hannam.ac.kr/>



- Python
 - Package: qrcode
 - <https://pypi.org/project/qrcode/>
 - <https://github.com/lincolnloop/python-qrcode>

```
In [13]: ### Package 설치
#!pip install qrcode
```

```
In [14]: ### Package
import qrcode
import os
```

```
In [15]: ### 내용
msg = 'http://bigdata.hannam.ac.kr/'

### QR image 생성
img_QR = qrcode.make(msg)
```

```
In [16]: ### QR 코드 출력
img_QR
```

Out[16]:



```
In [17]: ### png 파일로 저장
img_QR.save('QR_BDA.png')
```

```
In [18]: ##### Get the current working directory - 파일 저장 위치 확인
os.getcwd()
```

Out[18]: 'C:\\Users\\WW05\\Desktop\\W\\jupyter\\빅 데이터 응용_영상 처리_2023'

```
In [19]: ### QR 코드 삽입
img1 = cv2.imread(r'D:\\image\\img1.jpg')
img_qr_code = cv2.imread('QR_BDA.png')
```

```
In [20]: ### 크기 조정
img_qr_code_resize = cv2.resize(img_qr_code,
                                dsize=(0, 0),
                                fx=0.2, fy=0.2,
                                interpolation=cv2.INTER_AREA)
```

```
In [21]: ### 영상 삽입 - 좌측 상단
img_out = img1.copy()
h, w, c = img_qr_code_resize.shape
img_out[10:h+10, 10:w+10, :] = img_qr_code_resize
```

```
In [22]: ### 영상 출력
img_rgb = cv2.cvtColor(img_out, cv2.COLOR_BGR2RGB)
plt.imshow(img_rgb)
plt.axis('off')
plt.show()
```



```
In [23]: ### 영상 저장  
out_file = r'D:\Wimage\img1_qr.png'  
cv2.imwrite(out_file, img_out)
```

```
Out[23]: True
```