

빅데이터 분석이지만 "OpenCV" 강의

강사: 김정호

kimsmap@outlook.com

1 웹캠 이미지 받아오기

VideoCapture 사용하기

- OpenCV에서 카메라를 이용해서
영상을 받아오기 위해서는
VideoCapture를 사용

```
1 import cv2
2
3 # 카메라 세팅하기
4 CAMERA_ID = 0
5
6 cam = cv2.VideoCapture(CAMERA_ID)
7 if cam.isOpened() == False:
8     print('카메라를 열 수 없습니다.')
9     exit()
10
11 # 윈도우 설정하기
12 cv2.namedWindow('image', cv2.WINDOW_NORMAL)
13
14 while True:
15     ret, frame = cam.read()
16     if not ret:
17         print('프레임을 얻을 수 없습니다.')
18         break
19
20     cv2.imshow('image', frame)
21
22     key = cv2.waitKey(5) & 0xFF
23
24     if key == 27:
25         break
26
27 # 카메라 해제하기
28 cam.release()
29 cv2.destroyAllWindows()
```

2 웹캠 이미지 비디오 파일에

VideoWriter 사용하기

- OpenCV에서 동영상을 저장하기 위해서는 VideoWriter를 사용
- 파일명, 코덱, FPS, 해상도 순으로 videoWriter에 입력해 줌 (FPS = frame per second)
- 끝나면 release()를 해줘야 함

```
width = int(cam.get(cv2.CAP_PROP_FRAME_WIDTH))
height = int(cam.get(cv2.CAP_PROP_FRAME_HEIGHT))

vod = cv2.VideoWriter('output2.mp4', cv2.VideoWriter_fourcc(*'mp4v'), 30, (width,height))
vod = cv2.VideoWriter('output2.mp4', cv2.VideoWriter_fourcc(*'MJPG'), 30, (width,height))
```

```
1 import cv2
2
3 # 카메라 세팅하기
4 CAMERA_ID = 0
5
6 cam = cv2.VideoCapture(CAMERA_ID)
7 if cam.isOpened() == False:
8     print('카메라를 열 수 없습니다.')
9     exit()
10
11 # 비디오 쓰기 설정하기
12 vod = cv2.VideoWriter('output.avi', cv2.VideoWriter_fourcc(*'XVID'), 30, (960, 540))
13
14 # 윈도우 설정하기
15 cv2.namedWindow('image', cv2.WINDOW_NORMAL)
16
17 while True:
18     # 카메라 영상 읽기
19     ret, frame = cam.read()
20     if not ret:
21         print('프레임을 얻을 수 없습니다.')
22         break
23
24     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
25     cv2.imshow('image', gray)
26
27     # 비디오 파일에 프레임 저장할 때는 BGR 형식으로 저장해야 함 (GRAY 형식은 저장 불가)
28     gray_bgr = cv2.cvtColor(gray, cv2.COLOR_GRAY2BGR)
29     vod.write(gray_bgr)
30
31     key = cv2.waitKey(5) & 0xFF
32
33     if key == 27:
34         break
35
36 cam.release() # 카메라 해제하기
37 vod.release() # 비디오 파일 해제하기
38 cv2.destroyAllWindows()
```


3 영상에 글자 쓰기

- putText를 이용해서 글자쓰기 가능
- 폰트와 색상을 원하는 대로 선택할 수 있음

```
1 import cv2
2 import numpy as np
3
4 image = np.full(shape=(600, 800, 3), fill_value=255, dtype=np.uint8)
5
6 font_italic = "FONT_ITALIC"
7 font_hershey_plain = "FONT_HERSHEY_PLAIN"
8 font_hershey_complex = "FONT_HERSHEY_COMPLEX"
9 font_hershey_complex_small = "FONT_HERSHEY_COMPLEX_SMALL"
10 font_hershey_duplex = "FONT_HERSHEY_DUPLEX"
11 font_hershey_script_complex = "FONT_HERSHEY_SCRIPT_COMPLEX"
12 font_hershey_script_simplex = "FONT_HERSHEY_SCRIPT_SIMPLEX"
13 font_hershey_simplex = "FONT_HERSHEY_SIMPLEX"
14 font_hershey_triplex = "FONT_HERSHEY_TRIPLEX"
15
16 cv2.putText(image, font_italic, (50,50), cv2.FONT_ITALIC, 1, (255,0,0), 2)
17 cv2.putText(image, font_hershey_plain, (50,100), cv2.FONT_HERSHEY_PLAIN, 2, (0,255,0), 2)
18 cv2.putText(image, font_hershey_complex, (50,150), cv2.FONT_HERSHEY_COMPLEX, 1, (0,0,255), 2)
19 cv2.putText(image, font_hershey_complex_small, (50,200), cv2.FONT_HERSHEY_COMPLEX_SMALL, 1, (255,255,255), 2)
20 cv2.putText(image, font_hershey_duplex, (50,250), cv2.FONT_HERSHEY_DUPLEX, 1, (255,255,0), 2)
21 cv2.putText(image, font_hershey_script_complex, (50,300), cv2.FONT_HERSHEY_SCRIPT_COMPLEX, 1, (255,0,255), 2)
22 cv2.putText(image, font_hershey_script_simplex, (50,350), cv2.FONT_HERSHEY_SCRIPT_SIMPLEX, 1, (0,255,255), 2)
23 cv2.putText(image, font_hershey_simplex, (50,400), cv2.FONT_HERSHEY_SIMPLEX, 1, (51,102,0), 2)
24 cv2.putText(image, font_hershey_triplex, (50,450), cv2.FONT_HERSHEY_TRIPLEX, 1, (0,153,255), 2)
25
26
27 cv2.imshow("SAMPLE IMAGE", image)
28 cv2.waitKey()
29 cv2.destroyAllWindows()
```

FONT_ITALIC

FONT_HERSHEY_PLAIN

FONT_HERSHEY_COMPLEX

FONT_HERSHEY_DUPLEX

FONT_HERSHEY_SCRIPT_COMPLEX

FONT_HERSHEY_SCRIPT_SIMPLEX

FONT_HERSHEY_SIMPLEX

FONT_HERSHEY_TRIPLEX

4 트랙바 사용하기

- 숫자로 된 파라미터를 트랙바를 이용해서 조절할 수 있음
- Nothing이라고 함수를 쓰고 있지만 실제로는 별도 함수 지정 없이 사용할 수 있는 경우도 많음



```
1 import cv2
2 import numpy as np
3
4 def nothing():
5     pass
6
```

```
7 cv2.namedWindow('RGB track bar')
8 cv2.createTrackbar('red color', 'RGB track bar', 0, 255, nothing)
9 cv2.createTrackbar('green color', 'RGB track bar', 0, 255, nothing)
10 cv2.createTrackbar('blue color', 'RGB track bar', 0, 255, nothing)
11
12 cv2.setTrackbarPos('red color', 'RGB track bar', 125)
13 cv2.setTrackbarPos('green color', 'RGB track bar', 125)
14 cv2.setTrackbarPos('blue color', 'RGB track bar', 125)
15
16 img = np.zeros((512,512,3), np.uint8)
17
18 while(1):
19     redVal = cv2.getTrackbarPos('red color', 'RGB track bar')
20     greenVal = cv2.getTrackbarPos('green color', 'RGB track bar')
21     blueVal = cv2.getTrackbarPos('blue color', 'RGB track bar')
22
23     print(redVal)
24
25     cv2.rectangle(img, (0,0), (512,512), (blueVal, greenVal, redVal), -1)
26     cv2.imshow('RGB track bar', img)
27
28     if cv2.waitKey(30) & 0xFF == 27:
29         break
```


5 수행평가

- 1. 카메라로 영상을 입력받습니다.
- 2. 트랙바를 2개 생성합니다.
 - mode : 0, 1의 값을 가지고 초기값은 0
 - history : 0~500의 값을 가지고 초기값은 500
- 3. mode는 다음과 같은 설정을 가짐
 - 0: MOG2 방법, 1: KNN 방법
- 4. history는 MOG2와 KNN에 모두 적용되는 파라미터
 - 해당 파라미터라 알고리즘에 적용되도록 함 (setHistory() 메소드 이용)
- 5. ‘o’와 ‘p’키를 이용해서 히스토리를 늘리거나 줄이도록 함
 - o키는 10씩 작게, p키는 10씩 커지도록 설정
 - 10보다 작아지지 않게, 500보다 커지지 않게 설정
 - 값이 변경되면 트랙바도 같이 움직이도록 함 (setTrackbarPos 메소드 이용)
- 6. cv2.putText 함수를 이용해서 history 설정값을 이미지에 출력함 → 동영상에도 저장됨
- 7. ‘output.mp4’ 파일명을 가진 동영상으로 저장함

```
1 bgMethod1 = cv2.createBackgroundSubtractorMOG2()  
2 bgMethod2 = cv2.createBackgroundSubtractorKNN()  
3  
4 x = 100  
5 bgMethod1.setHistory(x)  
6 bgMethod1.setVarThreshold(x)
```