OpenCV-Python 기초 사용법

영상의 속성과 픽셀 값 참조

OpenCV 는 영상 데이터를 numpy.ndarray 로 표현

```
import cv2

img1 = cv2.imread('cat.bmp', cv2.IMREAD_GRAYSCALE)
img2 = cv2.imread('cat.bmp', cv2.IMREAD_COLOR)
```

numpy.ndarray

- ndim: 차원 수. len(img.shape)과 같음.
- shape: 각 차원의 크기. (h, w) 또는 (h, w, 3)
  - (h, w, 3): 컬러 영상
  - (h, w): 그레이스케일 영상
- size: 전체 원소 개수
- dtype: 원소의 데이터 타입. 영상 데이터는 uint8.

영상의 속성과 픽셀 값 참조

OpenCV 영상 데이터 자료형과 NumPy 자료형

OpenCV 자료형 (1채널)	NumPy 자료형	구분
cv2.CV_8U	numpy.uint8	8비트 부호없는 정수
cv2.CV_8S	numpy.int8	8비트 부호있는 정수
cv2.CV_16U	numpy.uint16	16비트 부호없는 정수
cv2.CV_16S	numpy.int16	16비트 부호있는 정수
cv2.CV_32S	numpy.int32	32비트 부호있는 정수
cv2.CV_32F	numpy.float32	32비트 부동소수형
cv2.CV_64F	numpy.float64	64비트 부동소수형
cv2.CV_16F	numpy.float16	16비트 부동소수형

- 그레이스케일 영상 : cv2.CV_8UC1 → numpy.uint8, shape = (h, w)
- 컬러 영상 : cv2.CV_8UC3 → numpy.uint8, shape = (h, w, 3)

영상의 속성과 픽셀 값 참조

영상의 속성 참조 예제

```python
import sys
import cv2


# 영상 불러오기
img1 = cv2.imread('cat.bmp', cv2.IMREAD_GRAYSCALE)
img2 = cv2.imread('cat.bmp', cv2.IMREAD_COLOR)

if img1 is None or img2 is None:
    print('Image load failed!')
    sys.exit()

# 영상의 속성 참조
print('type(img1):', type(img1))
print('img1.shape:', img1.shape)
print('img2.shape:', img2.shape)
print('img1.dtype:', img1.dtype)

# 영상의 크기 참조
h, w = img2.shape[:2]
print('img2 size: {} x {}'.format( *args: w, h))

```

영상의 속성과 픽셀 값 참조

영상의 속성 참조 예제

```
23  if len(img1.shape) == 2:
24      print('img1 is a grayscale image')
25  elif len(img1.shape) == 3:
26      print('img1 is a truecolor image')
27
28  cv2.imshow( winname: 'img1', img1)
29  cv2.imshow( winname: 'img2', img2)
30  cv2.waitKey()
31
32  # 영상의 픽셀 값 참조
33  '''
34  for y in range(h):
35      for x in range(w):
36          img1[y, x] = 255
37          img2[y, x] = (0, 0, 255)
38  '''
39  img1[:,:] = 255
40  img2[:,:] = (0, 0, 255)
41
42  cv2.imshow( winname: 'img1', img1)
43  cv2.imshow( winname: 'img2', img2)
44  cv2.waitKey()
45
46  cv2.destroyAllWindows()
```

영상의 생성, 복사, 부분 영상 추출

지정한 크기로 새 영상 생성하기

```
numpy.empty(shape, dtype=float, .....) --> arr
numpy.zeros(shape, dtype=float, .....) --> arr
numpy.ones(shape, dtype=None, .....) --> arr
numpy.full(shape, fill_value, dtype=None, .....) --> arr
```

- shape: 각 차원의 크기. (h, w) 또는 (h, w, 3)
- dtype: 원소의 데이터 타입. 일반적인 영상이면 numpy.uint8 지정
- arr: 생성된 영상 (numpy.ndarray)
- 참고사항
  - numpy.empty() 함수는 임의의 값으로 초기화된 배열을 생성
  - numpy.zeros() 함수는 0 으로 초기화된 배열을 생성
  - numpy.ones() 함수는 1 로 초기화된 배열을 생성
  - numpy.full() 함수는 fill_value 로 초기화된 배열을 생성

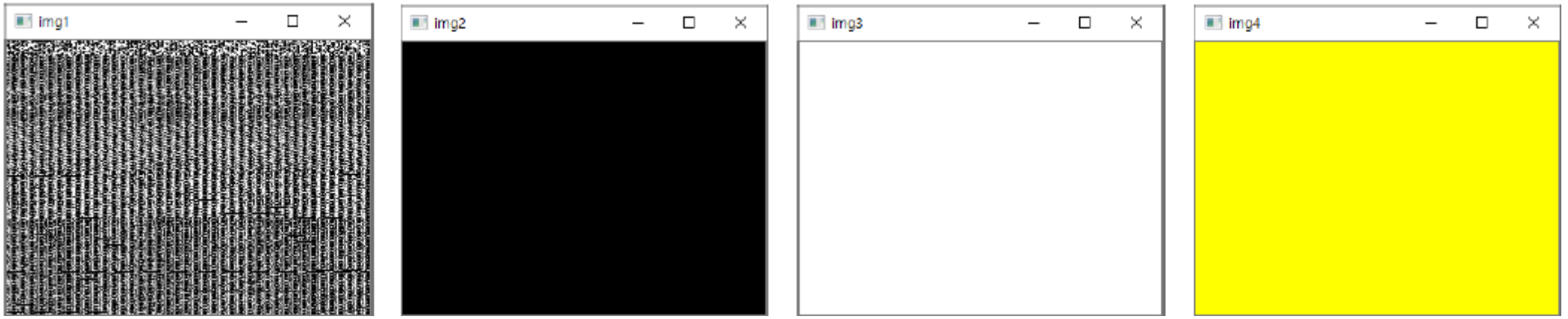
영상의 생성, 복사, 부분 영상 추출

영상의 생성 예제 코드

```python
import numpy as np
import cv2



# 새 영상 생성하기
img1 = np.empty((240, 320), dtype=np.uint8)       # grayscale image
img2 = np.zeros((240, 320, 3), dtype=np.uint8)    # color image
img3 = np.ones((240, 320), dtype=np.uint8) * 255  # dark gray
img4 = np.full((240, 320, 3), (0, 255, 255), dtype=np.uint8)  # yellow

cv2.imshow('img1', img1)
cv2.imshow('img2', img2)
cv2.imshow('img3', img3)
cv2.imshow('img4', img4)
cv2.waitKey()
cv2.destroyAllWindows()
```

영상의 생성, 복사, 부분 영상 추출

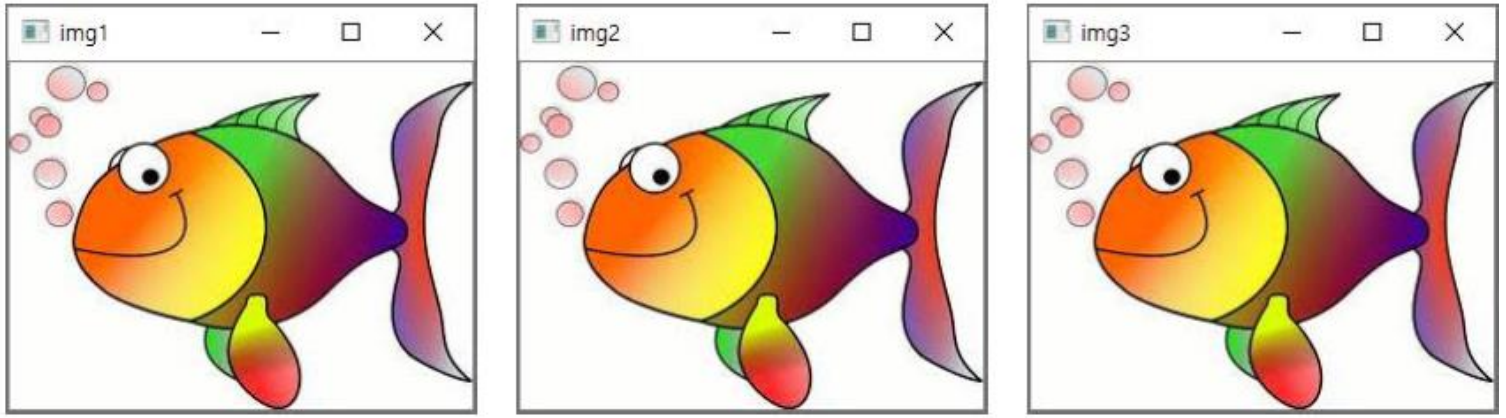
영상의 생성 예제 코드



영상의 생성, 복사, 부분 영상 추출

영상의 참조 및 복사 예제 코드

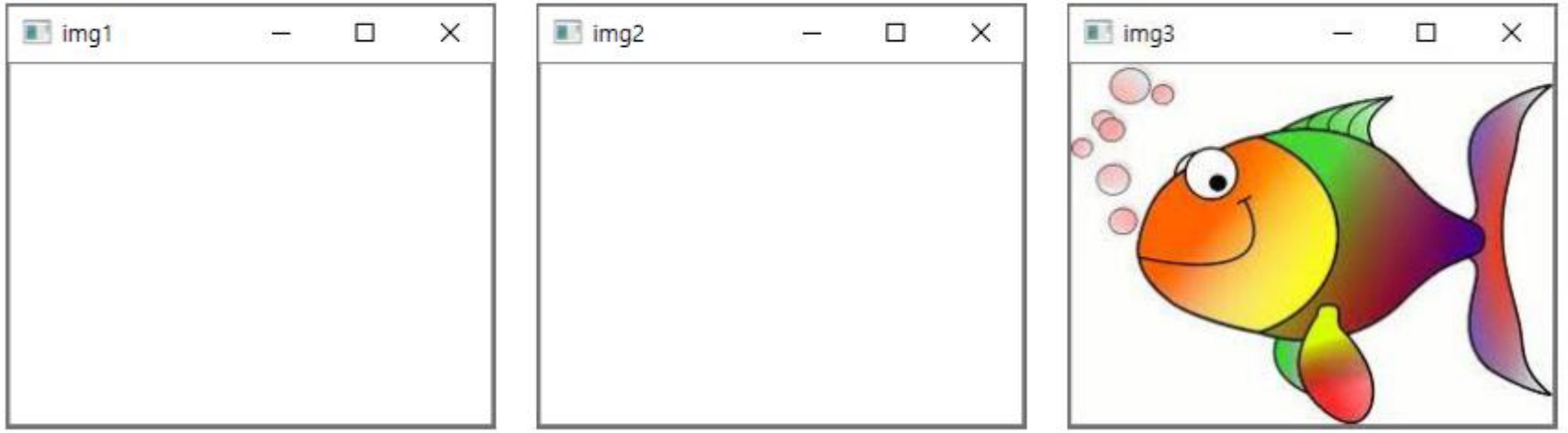
```
17
18  # 영상 복사
19  img1 = cv2.imread('HappyFish.jpg')
20
21  img2 = img1
22  img3 = img1.copy()
23
24  #img1.fill(255)
25
26  cv2.imshow('img1', img1)
27  cv2.imshow('img2', img2)
28  cv2.imshow('img3', img3)
29  cv2.waitKey()
30  cv2.destroyAllWindows()
31
```



영상의 생성, 복사, 부분 영상 추출

영상의 참조 및 복사 예제 코드

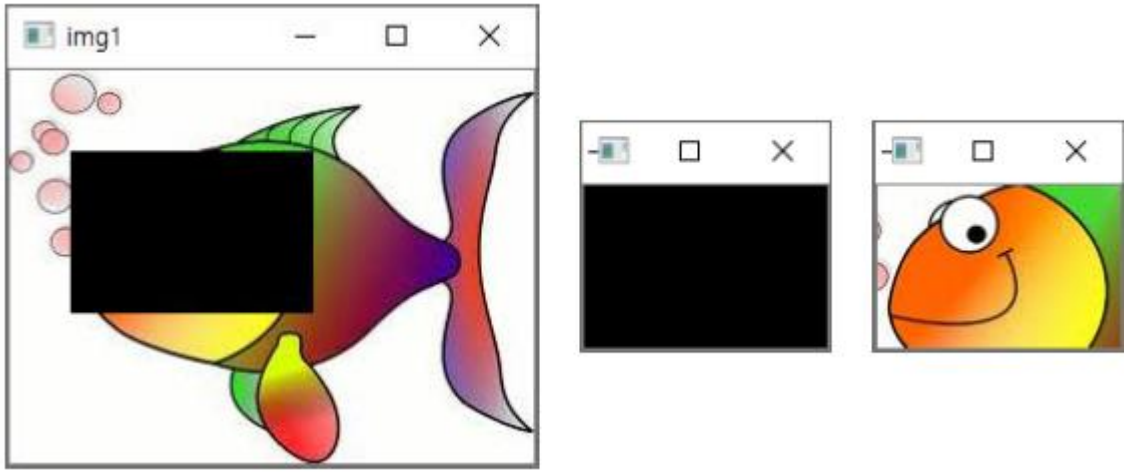
```
img1 = cv2.imread('HappyFish.jpg')

img2 = img1
img3 = img1.copy()

img1.fill(255)
```



영상의 생성, 복사, 부분 영상 추출

부분 영상 추출

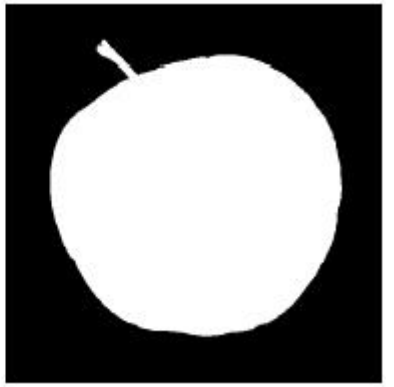
```
32  # 부분 영상 추출
33  img1 = cv2.imread('HappyFish.jpg')
34
35  img2 = img1[40:120, 30:150]  # numpy.ndarray의 슬라이싱
36  img3 = img1[40:120, 30:150].copy()
37
38  img2.fill(0)
39
40  cv2.imshow('img1', img1)
41  cv2.imshow('img2', img2)
42  cv2.imshow('img3', img3)
43  cv2.waitKey()
44  cv2.destroyAllWindows()
```



마스크 연산과 ROI

ROI

- Region of Interest, 관심 영역
- 영상에서 특정 연산을 수행하고자 하는 임의의 부분 영역



마스크 연산

- OpenCV 는 일부 함수에 대해 ROI 연산을 지원하며 , 이때 마스크 영상 을 인자로 함께 전달해야 함
  - (e.g.) cv2.copyTo(), cv2.calcHist(), cv2.bitwise_or(), cv2.matchTemplate(),
- 마스크 영상은 cv2.CV_8UC1 타입 (그레이스케일 영상)
- 마스크 영상의 픽셀 값이 0 이 아닌 위치에서만 연산이 수행됨

  → 보통 마스크 영상으로는 0 또는 255 로 구성된 이진 영상 (binary 을 사용

마스크 연산과 ROI

마스크 연산을 지원하는 픽셀 값 복사 함수

```
cv2.copyTo(src, mask, dst=None ) --> dst
```

- src: 입력 영상
- mask: 마스크 영상. cv2.CV_8U. (numpy.uint8)

  0 이 아닌 픽셀에 대해서만 복사 연산을 수행
- dst: 출력 영상 . 만약 src 와 크기 및 타입이 같은 dst를 입력으로 지정하면

  dst를 새로 생성하지 않고 연산을 수행

  그렇지않으면 dst를 새로 생성하여 연산을 수행한 후 반환함

마스크 연산과 ROI

마스크 연산 예제

```
src = cv2.imread('airplane.bmp', cv2.IMREAD_COLOR)
mask = cv2.imread('mask_plane.bmp', cv2.IMREAD_GRAYSCALE)
dst = cv2.imread('field.bmp', cv2.IMREAD_COLOR)

cv2.copyTo(src, mask, dst)
```

src, mask, dst는 모두 크기가 같아야 함.
src와 dst는 같은 타입이어야 하고, mask는 그레이스케일 타입의 이진 영상

- NumPy 의 불리언 인덱싱 (Boolean 을 이용한 마스크 연산

```
dst[mask > 0] = src[mask > 0]
```

마스크 연산과 ROI

마스크 연산 예제

```python
import sys
import cv2

# 마스크 영상을 이용한 영상 합성
src = cv2.imread('airplane.bmp', cv2.IMREAD_COLOR)
mask = cv2.imread('mask_plane.bmp', cv2.IMREAD_GRAYSCALE)
dst = cv2.imread('field.bmp', cv2.IMREAD_COLOR)

if src is None or mask is None or dst is None:
    print('Image load failed!')
    sys.exit()

cv2.copyTo(src, mask, dst)
# dst[mask > 0] = src[mask > 0]

cv2.imshow( winname: 'src', src)
cv2.imshow( winname: 'dst', dst)
cv2.imshow( winname: 'mask', mask)
cv2.waitKey()
cv2.destroyAllWindows()
```

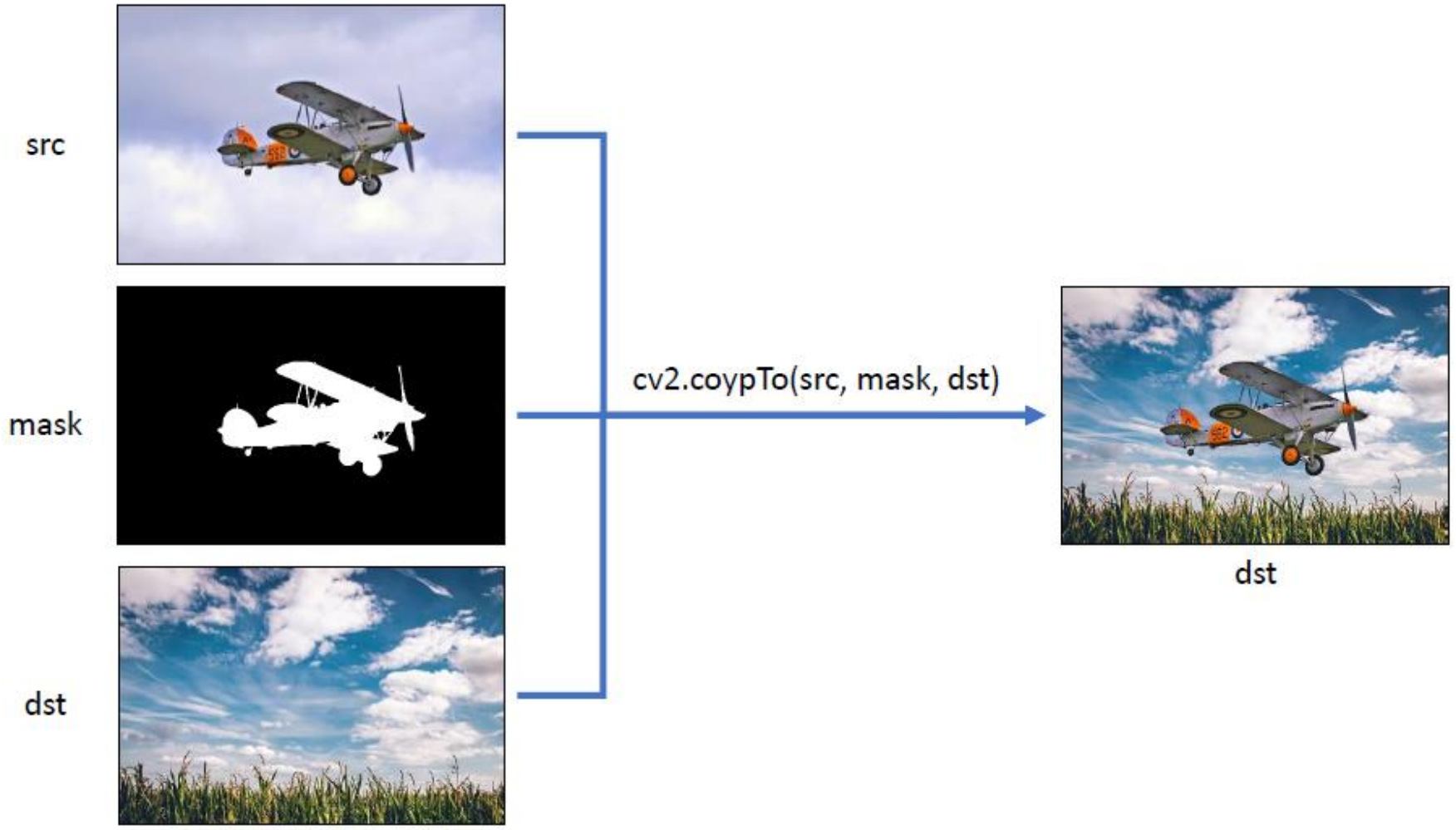
마스크 연산과 ROI

- **마스크 연산 예제**

```
22  # 알파 채널을 마스크 영상으로 이용
23  src = cv2.imread('cat.bmp', cv2.IMREAD_COLOR)
24  logo = cv2.imread('opencv-logo-white.png', cv2.IMREAD_UNCHANGED)x
25
26  if src is None or logo is None:
27      print('Image load failed!')
28      sys.exit()
29
30  mask = logo[:, :, 3]    # mask는 알파 채널로 만든 마스크 영상
31  logo = logo[:, :, :-1]  # logo는 b, g, r 3채널로 구성된 컬러 영상
32  h, w = mask.shape[:2]
33  crop = src[10:10+h, 10:10+w]  # logo, mask와 같은 크기의 부분 영상 추출
34
35  cv2.copyTo(logo, mask, crop)
36  #crop[mask > 0] = logo[mask > 0]
37
38  cv2.imshow( winname: 'src', src)
39  cv2.imshow( winname: 'logo', logo)
40  cv2.imshow( winname: 'mask', mask)
41  cv2.waitKey()
42  cv2.destroyAllWindows()
```

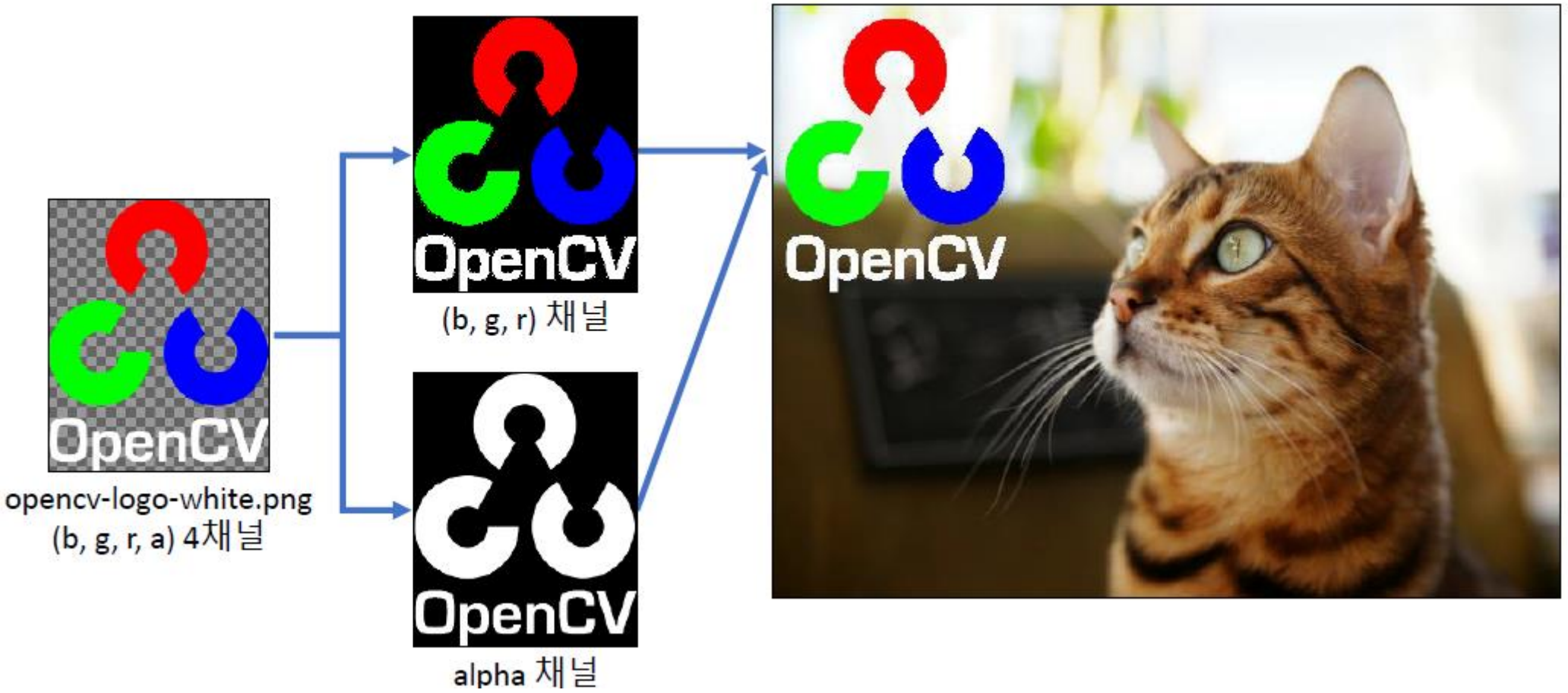
마스크 연산과 ROI

마스크 연산 예제



마스크 연산과 ROI

마스크 연산 예제



OpenCV 그리기 함수

OpenCV 그리기 함수

- OpenCV 는 영상에 선, 도형, 문자열을 출력하는 그리기 함수를 제공
- 선 그리기 : 직선, 화살표, 마커 등
- 도형 그리기 : 사각형, 원, 타원, 다각형 등
- 문자열 출력

그리기 함수 사용 시 주의할 점

- 그리기 알고리즘을 이용하여 영상의 픽셀 값 자체를 변경

  → 원본 영상이 필요하면 복사본을 만들어서 그리기 & 출력
- 그레이스케일 영상에는 컬러로 그리기 안 됨

  → cv2.cvtColor() 함수로 BGR 컬러 영상으로 변환한 후 그리기 함수 호출

OpenCV 그리기 함수

직선 그리기

```
cv2.line(img, pt1, pt2, color, thickness=None, lineType=None, shift=None) --> img
```

- img: 그림을 그릴 영상
- pt1, pt2: 직선의 시작점과 끝점 . (x, y) 튜플
- color: 선 색상 또는 밝기 . (B, G, R) 튜플 또는 정수값
- thickness: 선 두께 . 기본값은 1.
- lineType 선 타입 . cv2.LINE_4, cv2.LINE_8, cv2.LINE_AA 중 선택. 기본값은 cv2.LINE_8
- shift: 그리기 좌표 값의 축소 비율 . 기본값은 0.

OpenCV 그리기 함수

사각형 그리기

cv2.rectangle(img, pt1, pt2, color, thickness=None, lineType=None, shift=None) --> img
cv2.rectangle(img, rec, color, thickness=None, lineType=None, shift=None) --> img

- img: 그림을 그릴 영상
- pt1, pt2: 직선의 시작점과 끝점 . (x, y) 튜플
- color: 선 색상 또는 밝기 . (B, G, R) 튜플 또는 정수값
- thickness: 선 두께 . 기본값은 1.
- lineType 선 타입 . cv2.LINE_4, cv2.LINE_8, cv2.LINE_AA 중 선택. 기본값은 cv2.LINE_8
- shift: 그리기 좌표 값의 축소 비율 . 기본값은 0.

OpenCV 그리기 함수

원 그리기

cv2.circle(img, center, radius, color, thickness=None, lineType=None, shift=None) --> img

- img: 그림을 그릴 영상
- center: 원의 중심 좌표. (x, y) 튜플
- radius: 원의 반지름
- color: 선 색상 또는 밝기. (B, G, R) 튜플 또는 정수값
- thickness: 선 두께. 기본값은 1. 음수 (-1) 를 지정하면 내부를 채움
- lineType: 선 타입. cv2.LINE_4, cv2.LINE_8, cv2.LINE_AA 중 선택.
- 기본값은 cv2.LINE_8
- shift: 그리기 좌표 값의 축소 비율. 기본값은 0.

OpenCV 그리기 함수

다각형 그리기

cv2.polylines(img, pts, isClosed, color, thickness=None, lineType=None, shift=None) --> img

- img: 그림을 그릴 영상
- pts: 다각형 외곽 점들의 좌표 배열. numpy.ndarray 의 리스트
  - (e.g.) np.array([[10, 10], [50, 50], [10, 50]], dtype=np.int32)
- isClosed: 폐곡선 여부. True 또는 False 지정
- color: 선 색상 또는 밝기. (B, G, R) 튜플 또는 정수값
- thickness: 선 두께. 기본값은 1. 음수(-1) 를 지정하면 내부를 채움
- lineType: 선 타입. cv2.LINE_4, cv2.LINE_8, cv2.LINE_AA 중 선택 .
- 기본값은 cv2.LINE_8
- shift: 그리기 좌표 값의 축소 비율 . 기본값은 0.

OpenCV 그리기 함수

문자열 출력

cv2.putText(img, text, org, fontFace, fontScale, color, thickness=None, lineType=None, bottomLeftOrigin=None) --> img

- img: 그림을 그릴 영상
- text: 출력할 문자열
- org: 영상에서 문자열을 출력할 위치의 좌측 하단 좌표. (x, y) 튜플
- fontFace: 폰트 종류. cv2.FONT_HERSHEY_ 로 시작하는 상수 중 선택
- fontScale: 폰트 크기 확대/축소 비율
- color: 선 색상 또는 밝기. (B, G, R) 튜플 또는 정수값
- thickness: 선 두께. 기본값은 1. 음수(-1) 를 지정하면 내부를 채움
- lineType: 선 타입 . cv2.LINE_4, cv2.LINE_8, cv2.LINE_AA 중 선택 .
- bottomLeftOrigin: True 이면 영상의 좌측 하단을 원점으로 간주 . 기본값은 False.

OpenCV 그리기 함수

- **cv2.putText() 에서 지원하는 fontFace 상수와 실제 출력 모양**

OpenCV 그리기 함수

다양한 그리기 함수 실행 예제

```
1   import numpy as np
2   import cv2
3
4   img = np.full((400, 400, 3), 255, np.uint8)
5
6   cv2.line(img, (50, 50), (200, 50), (0, 0, 255), 5)
7   cv2.line(img, (50, 60), (150, 160), (0, 0, 128))
8
9   cv2.rectangle(img, (50, 200, 150, 100), (0, 255, 0), 2)
10  cv2.rectangle(img, (70, 220), (180, 280), (0, 128, 0), -1)
11
12  cv2.circle(img, (300, 100), 30, (255, 255, 0), -1, cv2.LINE_AA)
13  cv2.circle(img, (300, 100), 60, (255, 0, 0), 3, cv2.LINE_AA)
14
```

OpenCV 그리기 함수

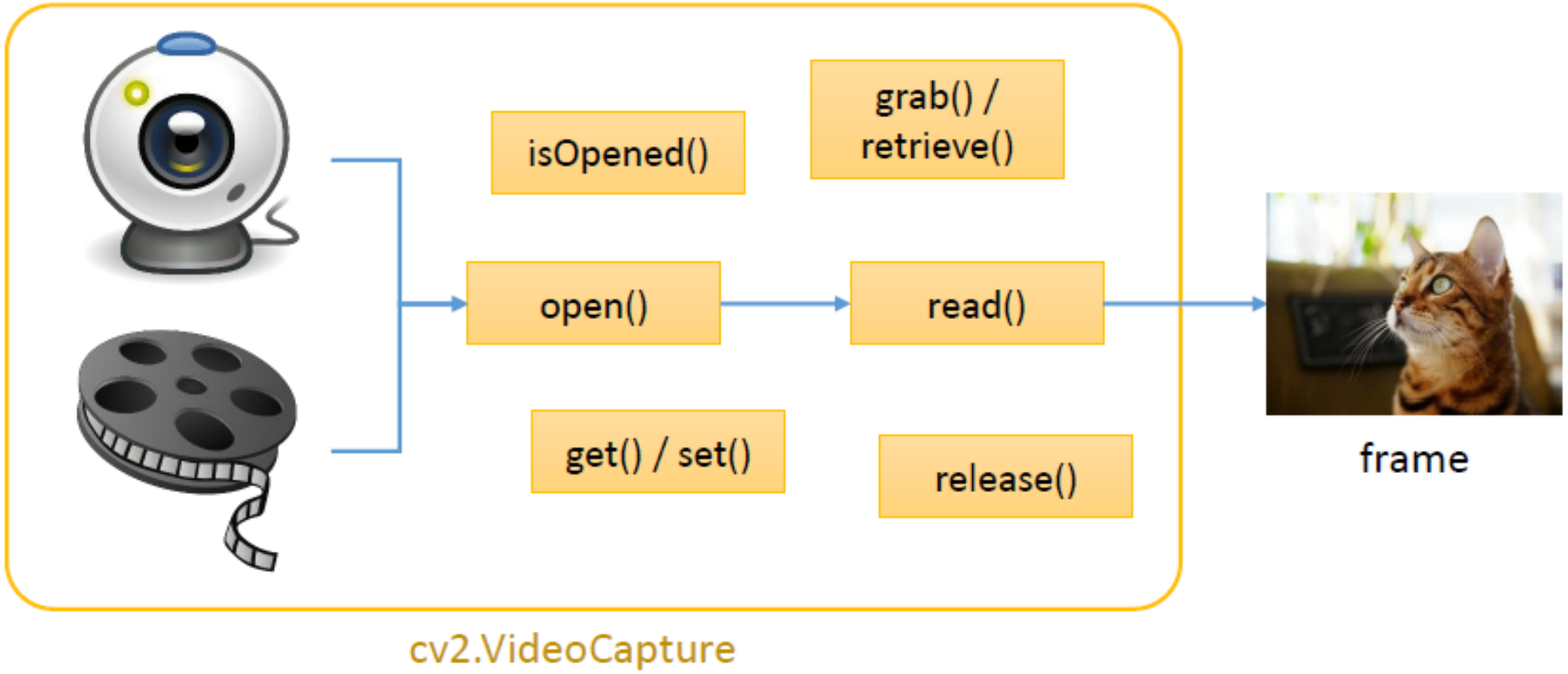
다양한 그리기 함수 실행 예제

```
15      pts = np.array([[250, 200], [300, 200], [350, 300], [250, 300]])
16      cv2.polylines(img, [pts], True, (255, 0, 255), 2)
17
18      text = 'Hello? OpenCV ' + cv2.__version__
19      cv2.putText(img, text, (50, 350), cv2.FONT_HERSHEY_SIMPLEX, 0.8,
20                  (0, 0, 255), 1, cv2.LINE_AA)
21
22      cv2.imshow("img", img)
23      cv2.waitKey()
24      cv2.destroyAllWindows()
```

카메라와 동영상 처리하기 1

cv2.VideoCapture 클래스

- OpenCV 에서는 카메라와 동영상으로부터 프레임(frame)을 받아오는 작업을 cv2.VideoCapture 클래스 하나로 처리함



카메라와 동영상 처리하기 1

카메라 열기

cv2.VideoCapture(index, apiPreference=None) --> retval

- index: camera_id + domain_offset_id
- 시스템 기본 카메라를 기본 방법으로 열려면 index 에 0 을 전달
- apiPreference: 선호하는 카메라 처리 방법을 지정
- retval: cv2.VideoCapture 객체

cv2.VideoCapture.open(index, apiPreference=None) --> retval

- retval: 성공하면 True, 실패하면 False.

카메라와 동영상 처리하기 1

동영상, 정지 영상 시퀀스, 비디오 스트림 열기

cv2.VideoCapture(filename, apiPreference=None) --> retval

- filename: 비디오 파일 이름 , 정지 영상 시퀀스 , 비디오 스트림 URL 등
  - e.g) 'video.avi', 'img_%02d.jpg', ' host:port script?params|auth
- apiPreference : 선호하는 동영상 처리 방법을 지정
- retval: cv2.VideoCapture 객체

cv2.VideoCapture.open(filename, apiPreference=None) --> retval

- retval: 성공하면 True, 실패하면 False.

카메라와 동영상 처리하기 1

비디오 캡쳐가 준비되었는지 확인

```
cv2.VideoCapture.isOpened() --> retval
```

- retval: cv2.VideoCapture 객체

비디오 캡쳐가 준비되었는지 확인

```
cv2.VideoCapture.read(image=None) --> retval, image
```

- retval: 성공하면 True, 실패하면 False.
- image: 현재 프레임 (numpy.ndarray)

카메라와 동영상 처리하기 1

카메라, 비디오 장치 속성 값 참조

cv2.VideoCapture.get(propId) --> retval

- propId : 속성 상수. OpenCV 문서 참조

CAP_PROP_FRAME_WIDTH	프레임 가로 크기
CAP_PROP_FRAME_HEIGHT	프레임 세로 크기
CAP_PROP_FPS	초당 프레임 수
CAP_PROP_FRAME_COUNT	비디오 파일의 총 프레임 수
CAP_PROP_POS_MSEC	밀리초 단위로 현재 위치
CAP_PROP_POS_FRAMES	현재 프레임 번호
CAP_PROP_EXPOSURE	노출

- retval : 성공하면 해당 속성 값, 실패하면 0.

카메라와 동영상 처리하기 1

카메라, 비디오 장치 속성 값 참조

cv2.VideoCapture.set(propId, value) --> retval

- propId : 속성 상수
- value: 속성 값
- retval: 성공하면 True, 실패하면 False.

카메라와 동영상 처리하기 1

카메라 처리 예제

```
import sys
import cv2

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()

    inversed = ~frame

    cv2.imshow('frame', frame)
    cv2.imshow('inversed', inversed)

    if cv2.waitKey(10) == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

기본 카메라 장치 열기.

카메라로부터 프레임을 정상적으로 받아오면 ret에는 True, frame에는 해당 프레임이 저장됨.

현재 프레임 반전

일정 시간(e.g. 10ms) 기다린 후 다음 프레임 처리. 만약 ESC 키를 누르면 while 루프 종료.

사용한 자원 해제

카메라와 동영상 처리하기 1

카메라 처리 예제

```
import sys
import cv2

cap = cv2.VideoCapture(0)

while True:
    ret, frame = cap.read()

    inversed = ~frame

    cv2.imshow('frame', frame)
    cv2.imshow('inversed', inversed)

    if cv2.waitKey(10) == 27:
        break

cap.release()
cv2.destroyAllWindows()
```

```
if not cap.isOpened():
    print("Camera open failed!")
    exit()
```

```
if not ret:
    break
```

카메라와 동영상 처리하기 1

- **동영상 처리 예제**

```python
import sys
import cv2


# 비디오 파일 열기
cap = cv2.VideoCapture('video1.mp4')

if not cap.isOpened():
    print("Video open failed!")
    sys.exit()

# 비디오 프레임 크기, 전체 프레임수, FPS 등 출력
print('Frame width:', int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)))
print('Frame height:', int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))
print('Frame count:', int(cap.get(cv2.CAP_PROP_FRAME_COUNT)))

fps = cap.get(cv2.CAP_PROP_FPS)
print('FPS:', fps)
```

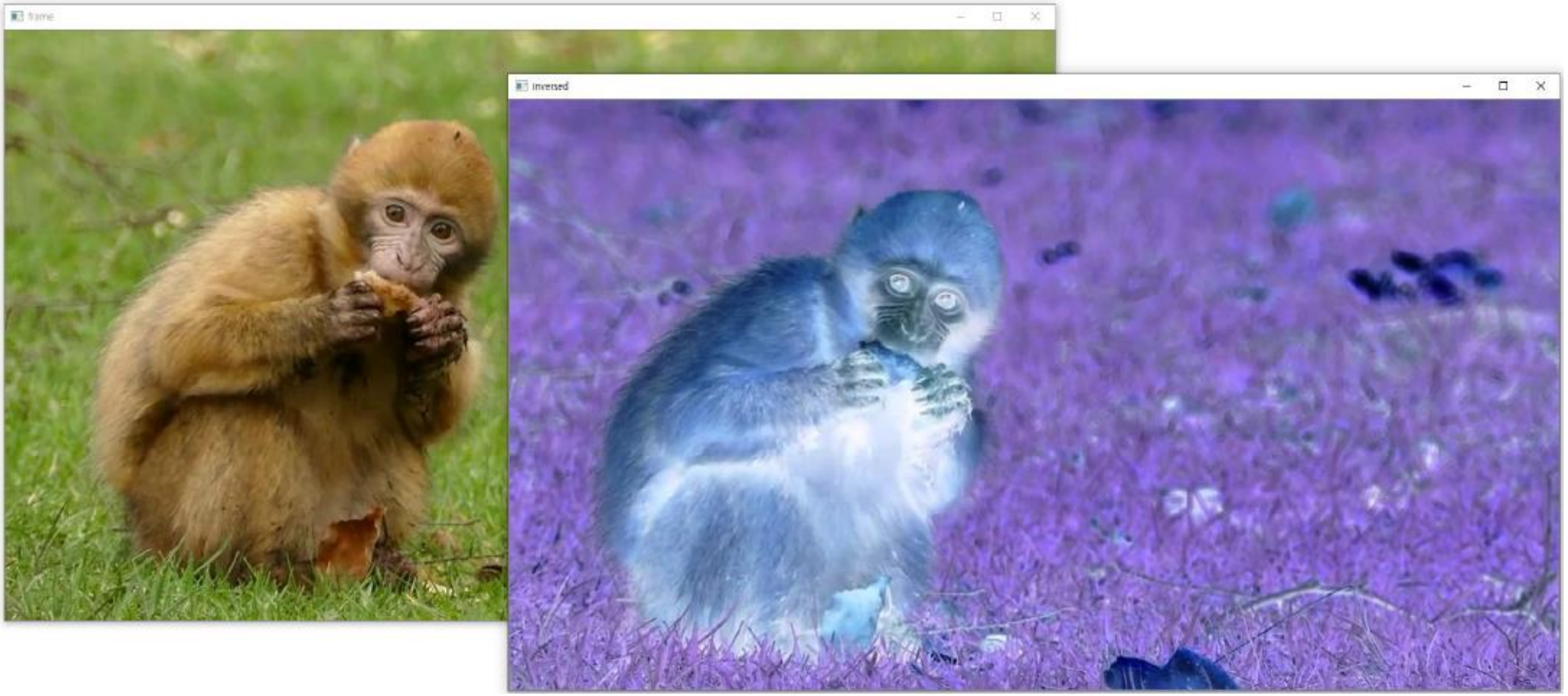
카메라와 동영상 처리하기 1

- **동영상 처리 예제**

```
19
20  delay = round(1000 / fps)
21
22  # 비디오 매 프레임 처리
23  while True:
24      ret, frame = cap.read()
25
26      if not ret:
27          break
28
29      inversed = ~frame  # 반전
30
31      cv2.imshow('frame', frame)
32      cv2.imshow('inversed', inversed)
33
34      if cv2.waitKey(delay) == 27:
35          break
36
37  cap.release()
38  cv2.destroyAllWindows()
```

카메라와 동영상 처리하기 1

동영상 처리 예제 실행 결과



카메라와 동영상 처리하기 2

cv2.VideoWriter 클래스

- OpenCV 에서는 cv2.VideoWriter 클래스를 이용하여 일련의 프레임을 동영상 파일로 저장할 수 있음
- 일련의 프레임은 모두 크기와 데이터 타입이 같아야 함

Fourcc (4 문자 코드, four character code)

- 동영상 파일의 코덱, 압축 방식, 색상, 픽셀 포맷 등을 정의하는 정수 값

cv2.VideoWriter_fourcc(*'DIVX')	DIVX MPEG-4 코덱
cv2.VideoWriter_fourcc(*'XVID')	XVID MPEG-4 코덱
cv2.VideoWriter_fourcc(*'FMP4')	FFMPEG MPEG-4 코덱
cv2.VideoWriter_fourcc(*'X264')	H.264/AVC 코덱
cv2.VideoWriter_fourcc(*'MJPG')	Motion-JPEG 코덱

카메라와 동영상 처리하기 2

저장을 위한 동영상 파일 열기

cv2.VideoWriter(filename, fourcc, fps, frameSize, isColor=None) --> retval

- filename: 비디오 파일 이름 (e.g. 'video.mp4)
- fourcc: fourcc (e.g. cv2.VideoWriter_fourcc(*'DIVX'))
- fps: 초당 프레임 수 (e.g. 30)
- frameSize 프레임 크기. (width, height) 튜플
- isColor 컬러 영상이면 True, 그렇지않으면 False.
- retval: cv2.VideoWriter 객체

cv2.VideoWriter.open(filename, fourcc, fps, frameSize, isColor=None) --> retval

- retval: 성공하면 True, 실패하면 False.

카메라와 동영상 처리하기 2

비디오 파일이 준비되었는지 확인

cv2.VideoWriter.isOpened() --> retval

- retval: 성공하면 True, 실패하면 False.

프레임 저장하기

cv2.VideoWriter.write(image) --> None

- image: 저장할 프레임 (numpy.ndarray)

카메라와 동영상 처리하기 2

웹카메라 입력을 동영상으로 저장하기

```
1   import sys
2   import cv2
3
4
5   cap = cv2.VideoCapture(0)
6
7   if not cap.isOpened():
8       print("Camera open failed!")
9       sys.exit()
10
11  w = round(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
12  h = round(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
13  fps = cap.get(cv2.CAP_PROP_FPS)
14
```

카메라와 동영상 처리하기 2

웹카메라 입력을 동영상으로 저장하기

```
15      fourcc = cv2.VideoWriter_fourcc(*'DIVX') # *'DIVX' == 'D', 'I', 'V', 'X'
16      delay = round(1000 / fps)
17
18      out = cv2.VideoWriter('output.avi', fourcc, fps, (w, h))
19
20      if not out.isOpened():
21          print('File open failed!')
22          cap.release()
23          sys.exit()
24
```

카메라와 동영상 처리하기 2

- **웹카메라 입력을 동영상으로 저장하기**

```
25  while True:
26      ret, frame = cap.read()
27
28      if not ret:
29          break
30
31      inversed = ~frame
32
33      out.write(inversed)
34
35      cv2.imshow('frame', frame)
36      cv2.imshow('inversed', inversed)
37
38      if cv2.waitKey(delay) == 27:
39          break
40
41  cap.release()
42  out.release()
43  cv2.destroyAllWindows()
```

키보드 이벤트 처리하기

키보드 입력 대기 함수

cv2.waitKey(delay=None) --> retval

- delay: 밀리초 단위 대기 시간. delay → 0 이면 무한히 기다림. 기본값은 0.
- retval: 눌린 키 값 (ASCII code). 키가 눌리지 않으면 1.
- 참고 사항
  - cv2.waitKey() 함수는 OpenCV 창이 하나라도 있을 때 동작함
  - 특정 키 입력을 확인하려면 ord() 함수를 이용

```
while True:
    if cv2.waitKey() == ord('q'):
        break
```

  - 주요 특수키 코드 : 27(ESC), 13(ENTER), 9(TAB)

키보드 이벤트 처리하기

키보드 특수키 입력 처리하기

- Windows 운영체제에서 방향키, 함수키 등의 특수키 입력은 cv2.waitKeyEx() 함수 사용

표 4-7 주요 특수 키에 해당하는 waitKeyEx() 함수 반환값

특수 키	waitKeyEx() 반환값	특수 키	waitKeyEx() 반환값
Insert	0x2d0000	F1	0x700000
Delete	0x2e0000	F2	0x710000
Home	0x240000	F3	0x720000
End	0x230000	F4	0x730000
Page Up	0x210000	F5	0x740000
Page Down	0x220000	F6	0x750000
←	0x250000	F7	0x760000
↑	0x260000	F8	0x770000
→	0x270000	F9	0x780000
↓	0x280000	F10	0x790000
		F11	0x7a0000
		F12	0x7b0000

키보드 이벤트 처리하기

- **키보드에서 i ' 또는 'I' 키를 누르면 영상을 반전**

```python
import sys
import numpy as np
import cv2


img = cv2.imread('cat.bmp', cv2.IMREAD_GRAYSCALE)

if img is None:
    print('Image load failed!')
    sys.exit()

cv2.namedWindow('image')
cv2.imshow('image', img)

```

키보드 이벤트 처리하기

- **키보드에서 i ' 또는 'I' 키를 누르면 영상을 반전**

```
15  while True:
16      keycode = cv2.waitKey()
17      if keycode == ord('i') or keycode == ord('I'):
18          img = ~img
19          cv2.imshow('image', img)
20      elif keycode == 27:
21          break
22
23  cv2.destroyAllWindows()
```

마우스 이벤트 처리하기

마우스 이벤트 콜백함수 등록 함수

cv2.setMouseCallback(windowName, onMouse, param=None) --> None

- windowName: 마우스 이벤트 처리를 수행할 창 이름
- onMouse: 마우스 이벤트 처리를 위한 콜백 함수 이름 .

  마우스 이벤트 콜백 함수는 다음 형식을 따라야 함

  onMouse(event, x, y, flags, param) --> None

- param: 콜백 함수에 전달할 데이터

마우스 이벤트 처리하기

마우스 이벤트 처리 함수(콜백 함수) 형식

onMouse(event, x, y, flags, param) --> None

- event: 마우스 이벤트 종류. cv2. 로 시작하는 상수
- x: 마우스 이벤트가 발생한 x 좌표
- y: 마우스 이벤트가 발생한 y 좌표
- flags: 마우스 이벤트 발생 시 상태. cv2.EVENT_ 로 시작하는 상수
- param: cv2.setMouseCallback() 함수에서 설정한 데이터

마우스 이벤트 처리하기

마우스 이벤트 처리 함수의 event 인자

MouseEventTypes 열거형 상수	값	설명
cv2.EVENT_MOUSEMOVE	0	마우스가 창 위에서 움직이는 경우
cv2.EVENT_LBUTTONDOWN	1	마우스 왼쪽 버튼이 눌려지는 경우
cv2.EVENT_RBUTTONDOWN	2	마우스 오른쪽 버튼이 눌려지는 경우
cv2.EVENT_MBUTTONDOWN	3	마우스 가운데 버튼이 눌려지는 경우
cv2.EVENT_LBUTTONUP	4	마우스 왼쪽 버튼이 떼어지는 경우
cv2.EVENT_RBUTTONUP	5	마우스 오른쪽 버튼이 떼어지는 경우
cv2.EVENT_MBUTTONUP	6	마우스 가운데 버튼이 떼어지는 경우
cv2.EVENT_LBUTTONDBLCLK	7	마우스 왼쪽 버튼을 더블클릭하는 경우
cv2.EVENT_RBUTTONDBLCLK	8	마우스 오른쪽 버튼을 더블클릭하는 경우
cv2.EVENT_MBUTTONDBLCLK	9	마우스 가운데 버튼을 더블클릭하는 경우
cv2.EVENT_MOUSEWHEEL	10	마우스 휠을 앞뒤로 돌리는 경우
cv2.EVENT_MOUSEHWHEEL	11	마우스 휠을 좌우로 움직이는 경우

마우스 이벤트 처리하기

마우스 이벤트 처리 함수의 flags 인자

MouseEventFlags열거형 상수	값	설명
cv2.EVENT_FLAG_LBUTTON	1	마우스 왼쪽 버튼이 눌려져 있음
cv2.EVENT_FLAG_RBUTTON	2	마우스 오른쪽 버튼이 눌려져 있음
cv2.EVENT_FLAG_MBUTTON	4	마우스 가운데 버튼이 눌려져 있음
cv2.EVENT_FLAG_CTRLKEY	8	CTRL 키가 눌려져 있음
cv2.EVENT_FLAG_SHIFTKEY	16	SHIFT 키가 눌려져 있음
cv2.EVENT_FLAG_ALTKEY	32	ALT 키가 눌려져 있음

마우스 이벤트 처리하기

마우스를 이용한 그리기 예제

```python
import sys
import numpy as np
import cv2



oldx = oldy = -1

def on_mouse(event, x, y, flags, param):
    global oldx, oldy

    if event == cv2.EVENT_LBUTTONDOWN:
        oldx, oldy = x, y
        print('EVENT_LBUTTONDOWN: %d, %d' % (x, y))

    elif event == cv2.EVENT_LBUTTONUP:
        print('EVENT_LBUTTONUP: %d, %d' % (x, y))

```

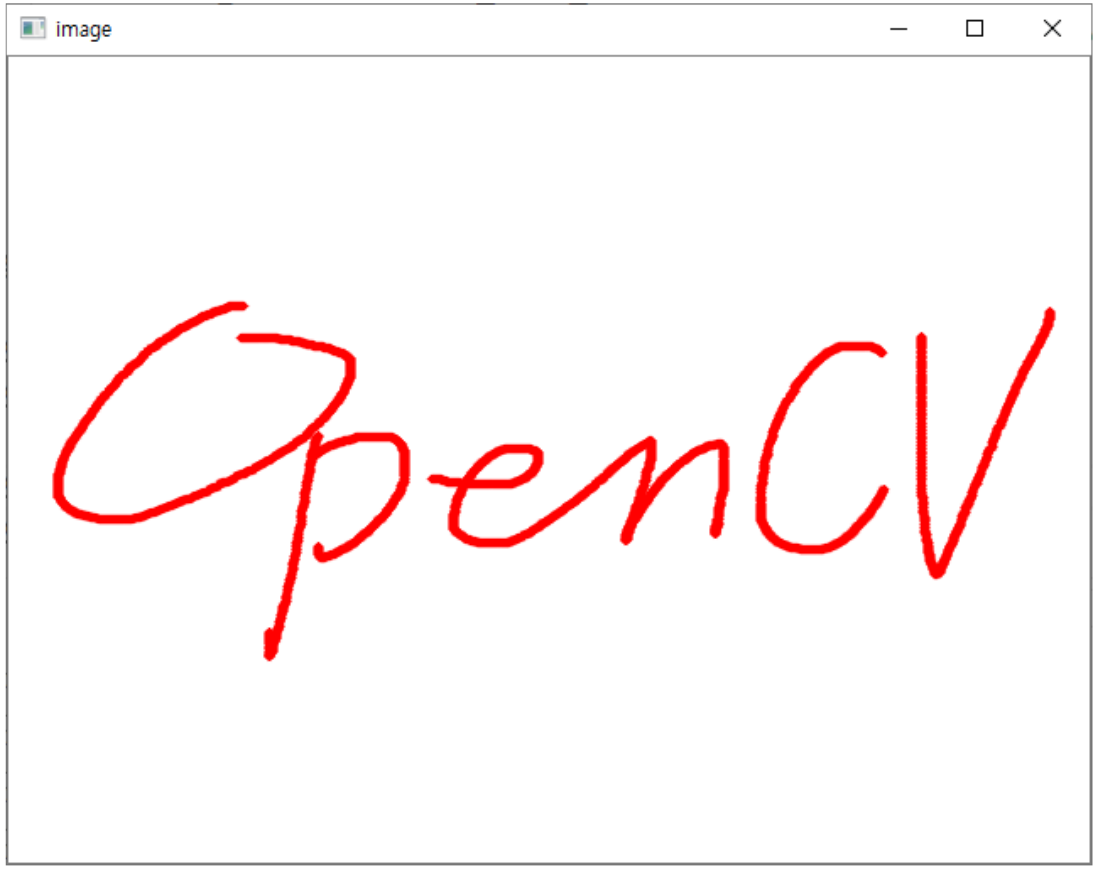
마우스 이벤트 처리하기

마우스를 이용한 그리기 예제

```
18      elif event == cv2.EVENT_MOUSEMOVE:
19          if flags & cv2.EVENT_FLAG_LBUTTON:
20              cv2.line(img, (oldx, oldy), (x, y), (0, 0, 255), 4, cv2.LINE_AA)
21              cv2.imshow('image', img)
22              oldx, oldy = x, y
23
24
25  img = np.ones((480, 640, 3), dtype=np.uint8) * 255
26
27  cv2.namedWindow('image')
28  cv2.setMouseCallback('image', on_mouse, img)
29
30  cv2.imshow('image', img)
31  cv2.waitKey()
32
33  cv2.destroyAllWindows()
```

마우스 이벤트 처리하기

마우스를 이용한 그리기 예제



트랙바 사용하기

트랙바(Trackbar)란?

- 프로그램 동작 중 사용자가 지정한 범위 안의 값을 선택할 수 있는 컨트롤
- OpenCV 에서 제공하는 유일한 ?) 그래픽 사용자 인터페이스

트랙바 사용하기

트랙바 생성 함수

cv2.createTrackbar(trackbarName, windowName, value, count, onChange) --> None

- trackbarName: 트랙바 이름
- windowName: 트랙바를 생성할 창 이름 .
- value: 트랙바 위치 초기값
- count: 트랙바 최댓값 . 최솟값은 항상 0.
- onChange: 트랙바 위치가 변경될 때마다 호출할 콜백 함수 이름

  트랙바 이벤트 콜백 함수는 다음 형식을 따름

  onChange(pos) --> None

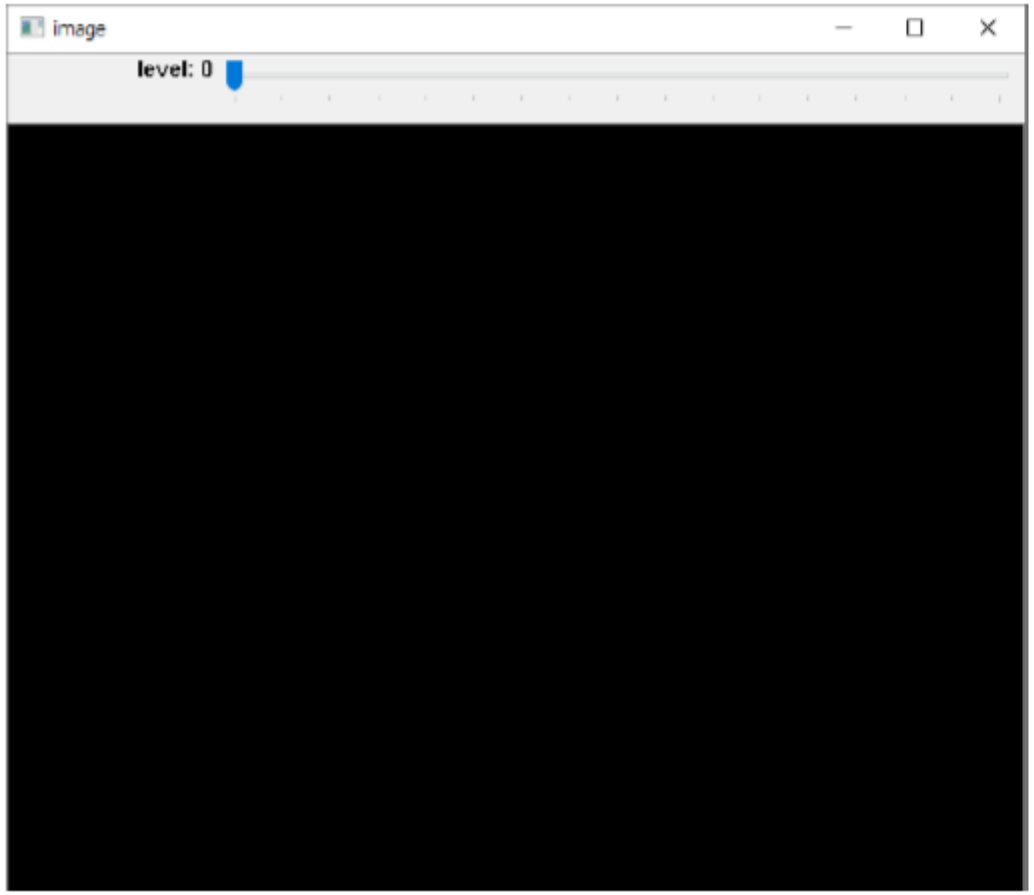
트랙바 사용하기

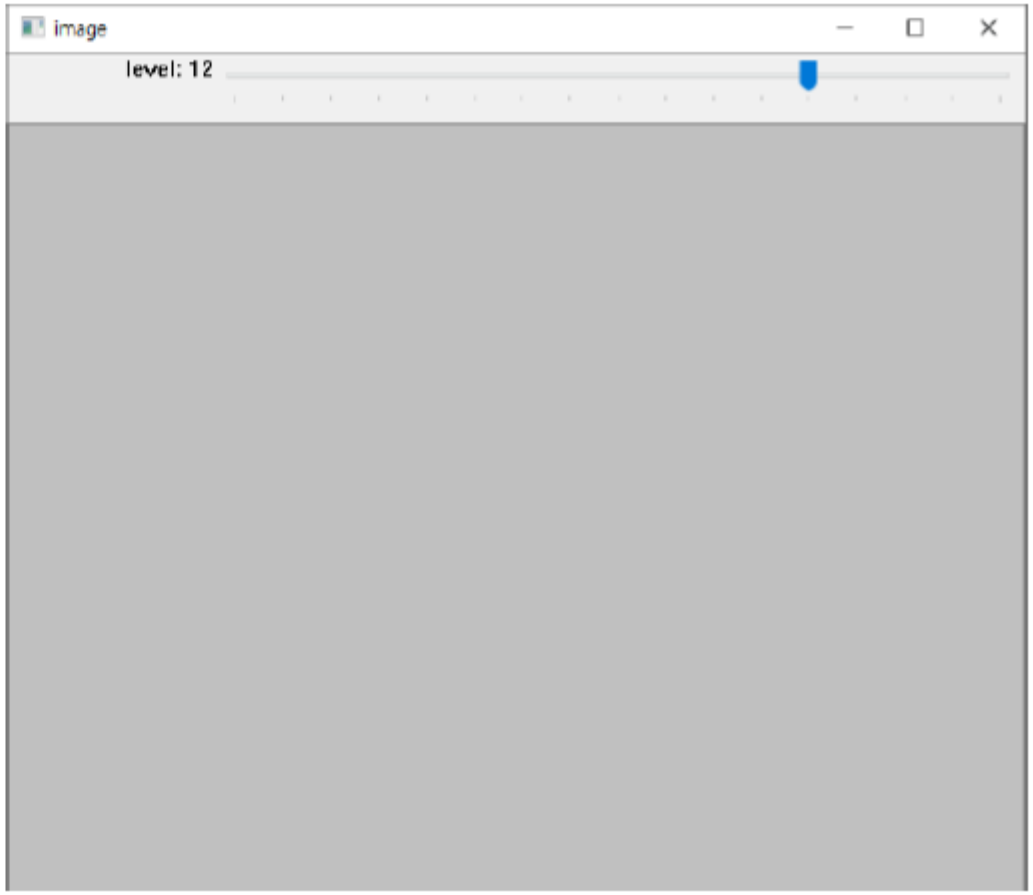
트랙바를 이용한 그레이스케일 레벨 표현

```python
import numpy as np
import cv2

def on_level_change(pos):
    value = pos * 16
    if value >= 255:
        value = 255

    img[:] = value
    cv2.imshow('image', img)

img = np.zeros((480, 640), np.uint8)
cv2.namedWindow('image')
cv2.createTrackbar('level', 'image', 0, 16, on_level_change)

cv2.imshow('image', img)
cv2.waitKey()
cv2.destroyAllWindows()
```

트랙바 사용하기

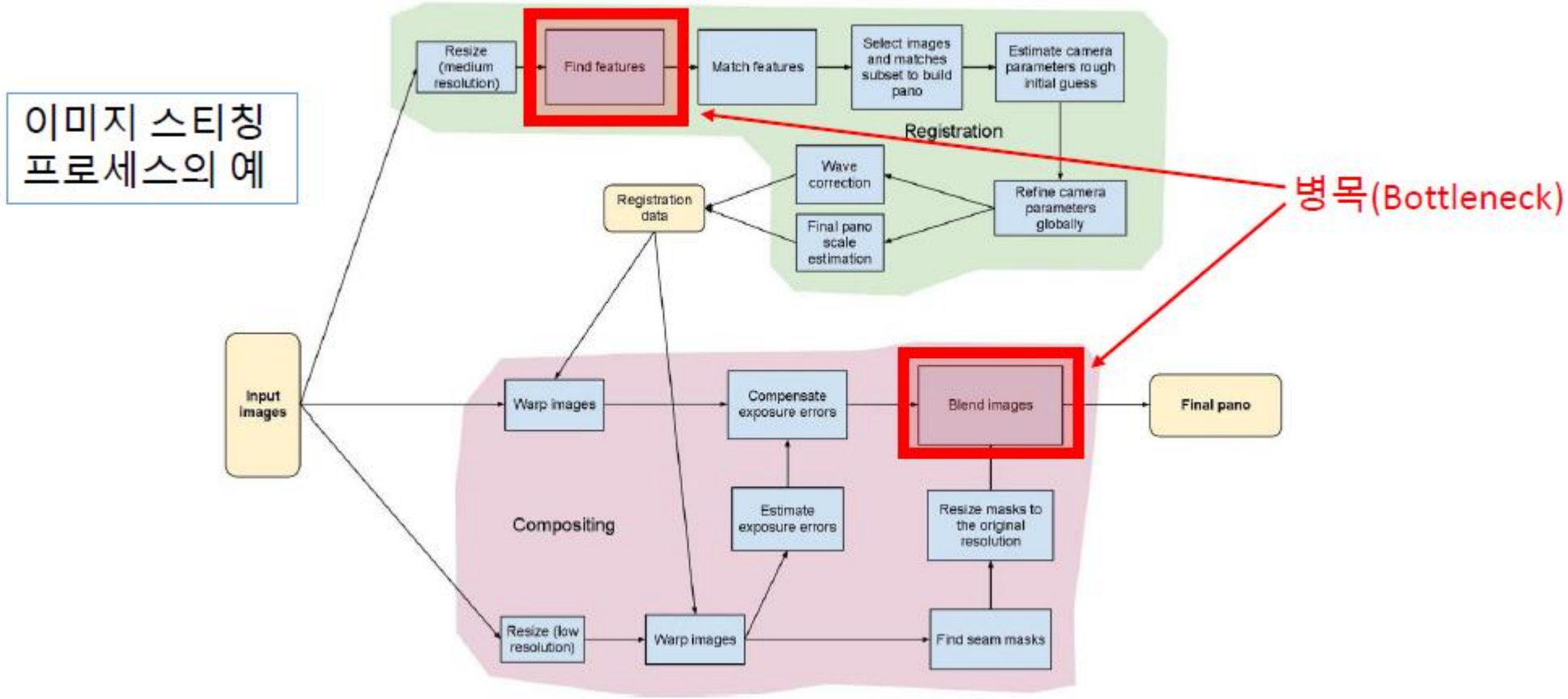
트랙바를 이용한 그레이스케일 레벨 표현





연산 시간 측정 방법

- 컴퓨터 비전은 대용량 데이터를 다루고 , 일련의 과정을 통해 최종 결과를 얻으므로 매 단계에서 연산 시간을 측정하여 관리할 필요가 있음



연산 시간 측정 방법

OpenCV 에서는 TickMeter 클래스를 이용하여 연산 시간을 측정

cv2.TickMeter () --> tm

- tm: cv2.TickMeter 객체
- tm.start: 시간 측정 시작
- tm.stop: 시간 측정 끝
- tm.reset: 시간 측정 초기화
- tm.getTimeSec: 측정 시간을 초 단위로 반환
- tm.getTimeMilli: 측정 시간을 밀리 초 단위로 반환
- tm.getTimeMicro: 측정 시간을 마이크로 초 단위로 반환

연산 시간 측정 방법

- **특정 연산의 시간 측정 예제**

```python
import sys
import time
import numpy as np
import cv2

img = cv2.imread('hongkong.jpg')

tm = cv2.TickMeter()

tm.reset()
tm.start()
t1 = time.time()

edge = cv2.Canny(img, 50, 150)

tm.stop()
print('time:', (time.time() - t1) * 1000)
print('Elapsed time: {}ms.'.format(tm.getTimeMilli()))
```

[프로젝트] 동영상 전환 이펙트

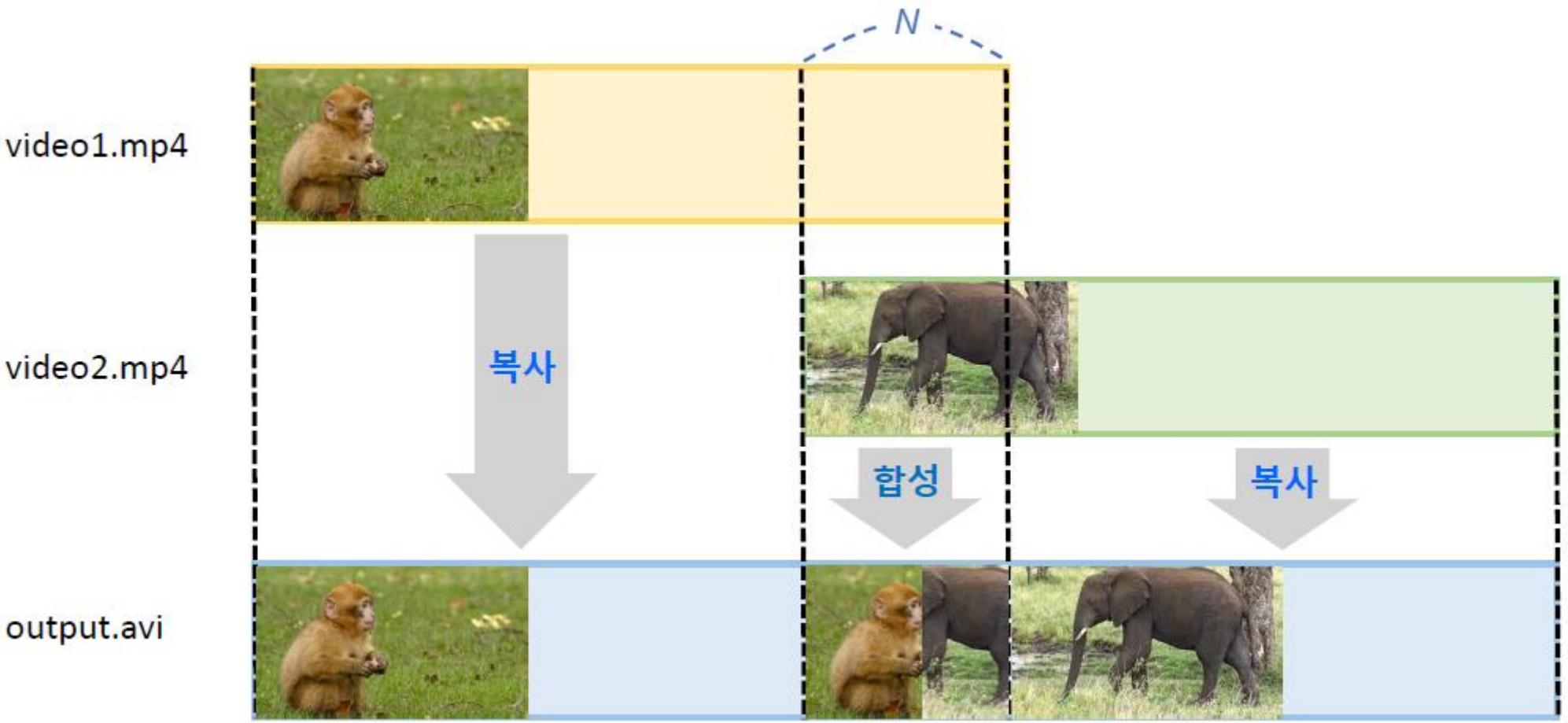
동영상 전환 이펙트

- 두 동영상 클립 사이에 추가되는 애니메이션 효과
- 페이드 인 (fade in), 페이드 아웃 (fade out), 디졸브 (dissolve), 밀기, 확대 등

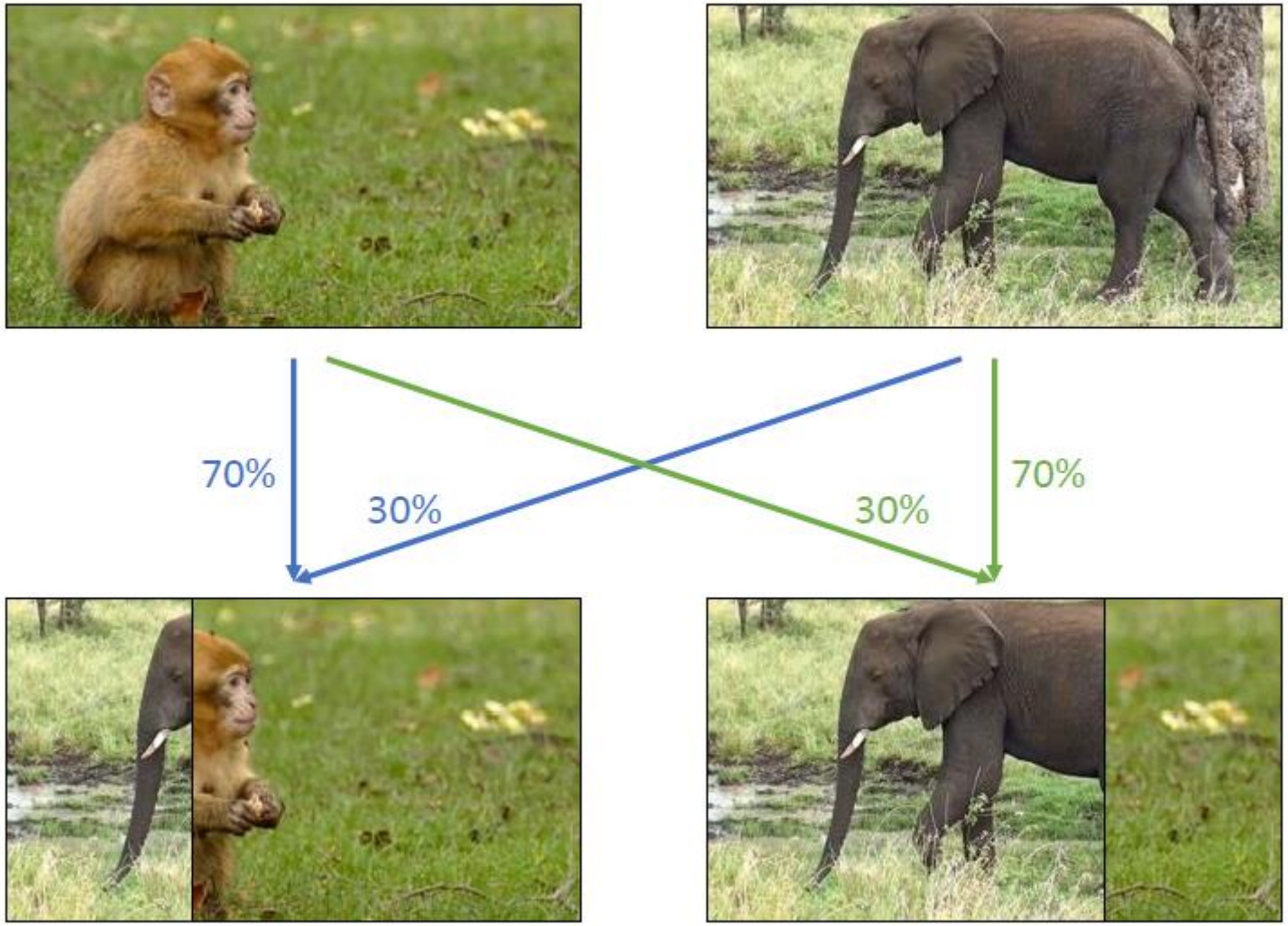
구현 할 기능

- 두 개의 동영상 동시 열기
- 첫 번째 동영상의 마지막 N 개 프레임과 두 번째 동영상의 처음 N 개 프레임을 합성
- 합성된 영상을 동영상으로 저장하기

[프로젝트] 동영상 전환 이펙트



[프로젝트] 동영상 전환 이펙트



[프로젝트] 동영상 전환 이펙트

```python
import sys
import numpy as np
import cv2



# 두 개의 동영상을 열어서 cap1, cap2로 지정
cap1 = cv2.VideoCapture('video1.mp4')
cap2 = cv2.VideoCapture('video2.mp4')

if not cap1.isOpened() or not cap2.isOpened():
    print('video open failed!')
    sys.exit()

# 두 동영상의 크기, FPS는 같다고 가정함
frame_cnt1 = round(cap1.get(cv2.CAP_PROP_FRAME_COUNT))
frame_cnt2 = round(cap2.get(cv2.CAP_PROP_FRAME_COUNT))
fps = cap1.get(cv2.CAP_PROP_FPS)
effect_frames = int(fps * 2)

```

[프로젝트] 동영상 전환 이펙트

```
20  print('frame_cnt1:', frame_cnt1)
21  print('frame_cnt2:', frame_cnt2)
22  print('FPS:', fps)
23
24  delay = int(1000 / fps)
25
26  w = round(cap1.get(cv2.CAP_PROP_FRAME_WIDTH))
27  h = round(cap1.get(cv2.CAP_PROP_FRAME_HEIGHT))
28  fourcc = cv2.VideoWriter_fourcc(*'DIVX')
29
30  # 출력 동영상 객체 생성
31  out = cv2.VideoWriter('output.avi', fourcc, fps, (w, h))
32
```

[프로젝트] 동영상 전환 이펙트

```
33  # 1번 동영상 복사
34  for i in range(frame_cnt1 - effect_frames):
35      ret1, frame1 = cap1.read()
36
37      if not ret1:
38          print('frame read error!')
39          sys.exit()
40
41      out.write(frame1)
42      print('.', end='')
43
44      cv2.imshow('output', frame1)
45      cv2.waitKey(delay)
```

[프로젝트] 동영상 전환 이펙트

```
47      # 1번 동영상 뒷부분과 2번 동영상 앞부분을 합성
48      for i in range(effect_frames):
49          ret1, frame1 = cap1.read()
50          ret2, frame2 = cap2.read()
51
52          if not ret1 or not ret2:
53              print('frame read error!')
54              sys.exit()
55
56          dx = int(w / effect_frames) * i
57
58          frame = np.zeros((h, w, 3), dtype=np.uint8)
59          frame[:, 0:dx, :] = frame2[:, 0:dx, :]
60          frame[:, dx:w, :] = frame1[:, dx:w, :]
61
62          #alpha = i / effect_frames
63          #frame = cv2.addWeighted(frame1, 1 - alpha, frame2, alpha, 0)
64
65          out.write(frame)
66          print('.', end='')
67
68          cv2.imshow('output', frame)
69          cv2.waitKey(delay)
70
```

[프로젝트] 동영상 전환 이펙트

```
71      # 2번 동영상을 복사
72      for i in range(effect_frames, frame_cnt2):
73          ret2, frame2 = cap2.read()
74
75          if not ret2:
76              print('frame read error!')
77              sys.exit()
78
79          out.write(frame2)
80          print('.', end='')
81
82          cv2.imshow('output', frame2)
83          cv2.waitKey(delay)
84
85      print('\noutput.avi file is successfully generated!')
86
87      cap1.release()
88      cap2.release()
89      out.release()
90      cv2.destroyAllWindows()
```