

# 영상처리

## CHAPTER 02 영상 및 비디오 입출력

2019. 3

한밭대학교 정보통신공학과

이충호

# 01. 영상 입출력과 디스플레이

- ▶ Input/Output, Displaying, Saving of image files
  - ▶ [] options, → return values of functions
- ▶ Table 2. Input output and display functions for images

Function	
cv2.imread(filename[,flags])→retval	Read an image
cv2.imwrite(filename,img[,params])→retval	Write an image
cv2.namedWindow(winname, [,flags])	Create a window
cv2.imshow(winname,mat)	Display an image
cv2.waitKey( [delay])→retval	Wait a keyboard input
cv2.destroyWindow (winname)	Destroy an window
cv2.destroyAllWindows()	Destroy all windows

# 01. 영상 입출력과 디스플레이

▶ Try 0201.py~ 0206.py.

0201.py :영상 파일 읽기 및 파일 표시

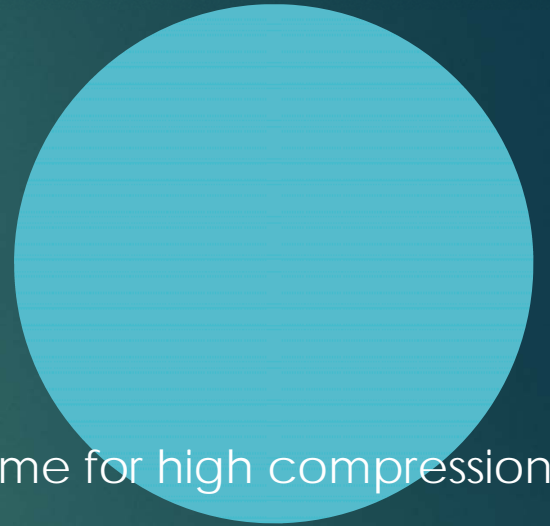
512x512 size, 3 channels, BGR(Blue Green Red)

waitKey(): delay = 0, wait a keyboard input infinitely.

0202.py: 영상 파일 저장

compression rate 9, compressing range [0, 9], takes longer time for high compression, default value is 3

[cv2.IMWRITE\_JPEG\_QUALITY,90] 90% quality, quality range [0,100], default 95



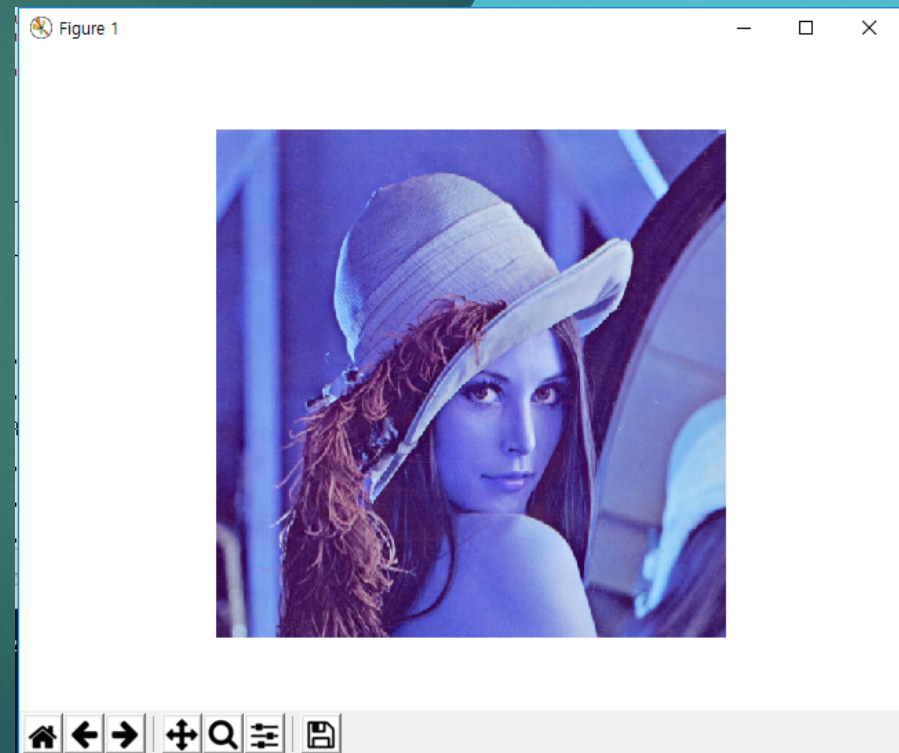
# 01. 영상 입출력과 디스플레이

## ▶ 0202.py

`cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB) # cvt means convert`

```
import cv2
from matplotlib import pyplot as plt

imageFile = './data/lena.jpg'
imgBGR = cv2.imread(imageFile)
plt.axis('off')
plt.imshow(imgBGR)
plt.show()
```





# 01. 영상 입출력과 디스플레이

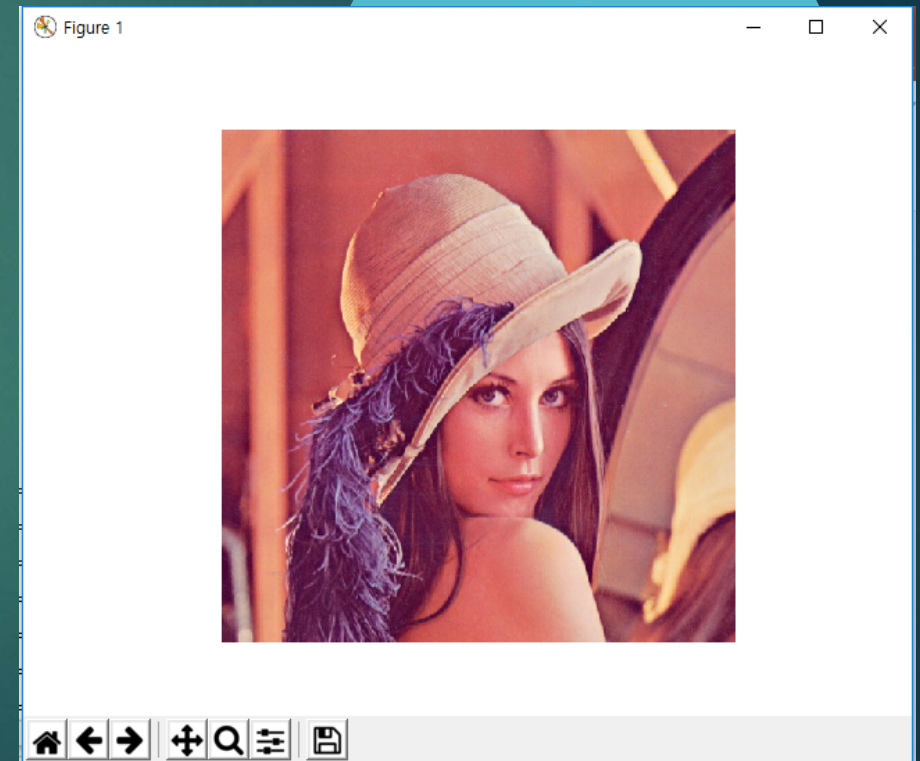
## ▶ 0203.py 컬러 영상 표시

`cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB)` # cvt means convert

```
import cv2
from matplotlib import pyplot as plt
```

```
imageFile = './data/lena.jpg'
imgBGR = cv2.imread(imageFile)
plt.axis('off')
#plt.imshow(imgBGR)
#plt.show()
```

```
imgRGB = cv2.cvtColor(imgBGR, cv2.COLOR_BGR2RGB)
plt.imshow(imgRGB)
plt.show()
```

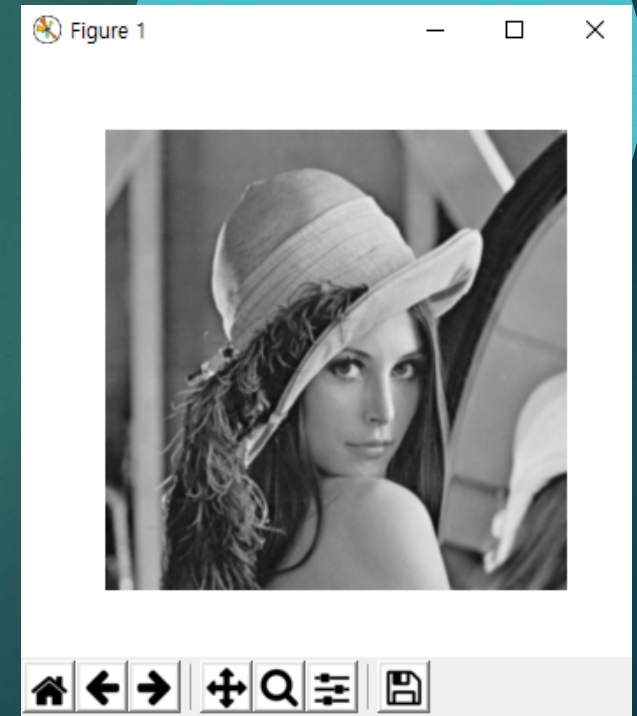


# 01. 영상 입출력과 디스플레이

- ▶ 0204.py 그레이스케일 영상표시

```
plt.imshow(imgGray, cmap='gray', interpolation='biscubic')
```

# biscubic: 인접한 16개의 화소값과 거리에 따른 가중치로 보간



# 01. 영상 입출력과 디스플레이

- ▶ 0205.py 여백조정 및 영상저장
- ▶ `plt.figure(figsize=(6,6))`

`plt.figure(figsize=(6,6))` # figure size is 6 inches x 6 inches

`plt.subplots_adjust(left=0, right=1, bottom=0, top=1)` # 출력범위 left<right bottom < top

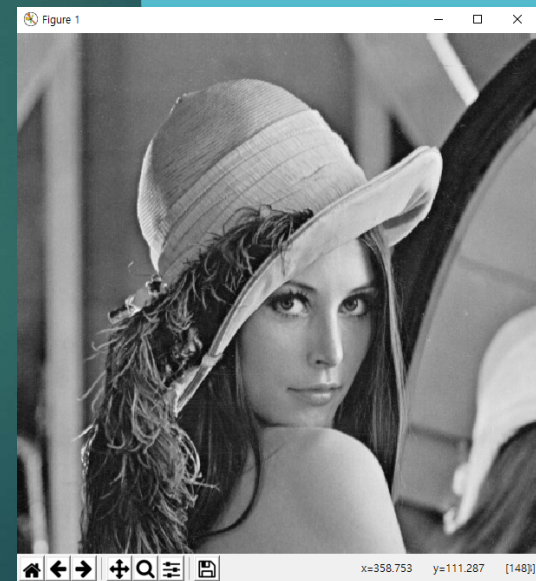


그림 2.4 Matplotlib 여백제거



# 01. 영상 입출력과 디스플레이

## ▶ 0206.py 서브플롯에 영상표시

Read four images

Convert images BRG to RGB

fig, ax = plt.subplots(2, 2, figsize=(10,10), sharey=True) # 2x2 subplot, 10x10 size

ax[0][0] # top left

ax[0][1] # top right

ax[1][0] # bottom left

ax[1][1] # bottom right

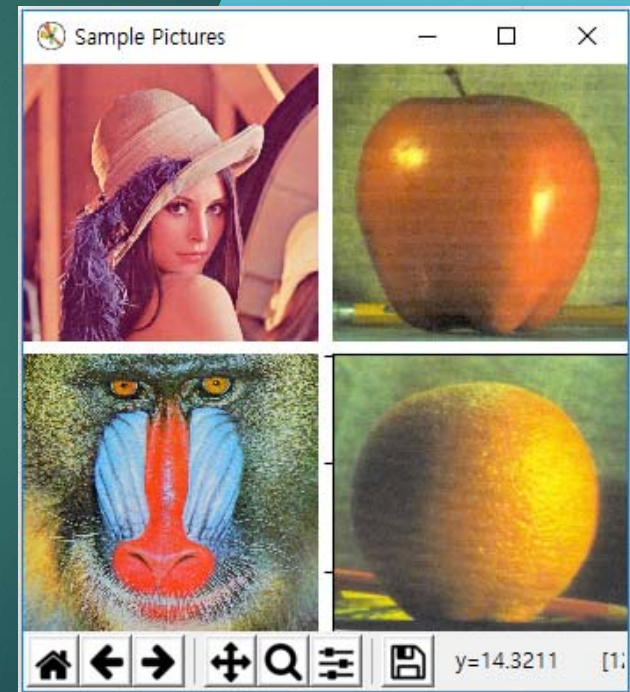


그림 2.5 Matplotlib을 이용한 다중 영상 표시



## 02. 비디오 프레임 캡처와 화면 표시

- ▶ Analog to digital
- ▶ Frame
- ▶ A video is a sequence of frames
- ▶ VideoCapture(), while, Videocapture.read(), esc key

표2.2 비디오프레임 캡처와 화면 표시 함수

Function	Explanation
cv2.VideoCapture(), cv2.VideoCapture(filename), cv2.VideoCapture(device) → <VideoCapture object>	Capture a video and create an object
cv2.VideoCapture.read([image]) → retval, image	Read a frame
cv2.VideoCapture.grab() → retval	Grab a frame
cv2.VideoCapture.retrieve([image[, channel]]) → retval, image	Retrieve a frame
cv2.VideoCapture.release()	Release a frame
cv2.VideoCapture.get(propid) → retval	Get features of a video
cv2.VideoCapture.set(propid, value) → retval	Set features of a video

## 02. 비디오 프레임 캡처

표2.3 Property ID 주요상수

10

Property_id	explanation
cv2.CAP_PROP_POS_MSEC	Milliseconds position
cv2.CAP_PROP_POS_FRAMES	Frame number to capture
cv2.CAP_PROP_FRAME_WIDTH	Frame width
cv2.CAP_PROP_FRAME_HEIGHT	Frame height
cv2.CAP_PROP_FPS	Frame per second
cv2.CAP_PROP_FOURCC	Four characters of Codec
cv2.CAP_PROP_FRAME_COUNT	Total number of frames of a video
cv2.CAP_PROP_CONVERT_RGB	Convert to RGB
cv2.CAP_PROP_FORMAT	Format of video
cv2.CAP_PROP_BRIGHTNESS	Brightness of camera
cv2.CAP_PROP_CONTRAST	Contrast of camera
cv2.CAP_PROP_SATURATION	Saturation of camera
cv2.CAP_PROP_HUE	Hue of camera
cv2.CAP_PROP_GAIN	Gain of camera
cv2.CAP_PROP_EXPOSURE	Exposure of camera

## 02. 비디오 프레임 캡처와 화면 표시

11

```
#0207.py
import cv2
Cap = cv2.VideoCapture(0) # 0번 카메라
#cap = cv2.VideoCapture('./data/vtest.avi') # 저장된 video 이용 시
Cap.set(cv2.CAP_PROP_FRAME_WIDTH, 320) #카메라 이용 시
Cap.set(cv2.CAP_PROP_FRAME_HEIGHT, 240) #카메라 이용 시
Frame_size = (int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)),
              int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))
Print('frame_size = ', frame_size)
While True:
    retval, frame = cap.read() # 프레임 캡처
    if not retval:
        break
    cv2.imshow('frame',frame)
    key = cv2.waitKey(25) # esc key 누르면
    if key == 27: # Esc
        break
if cap.isOpened():
    cap.release()
cv2.destroyAllWindows()
```



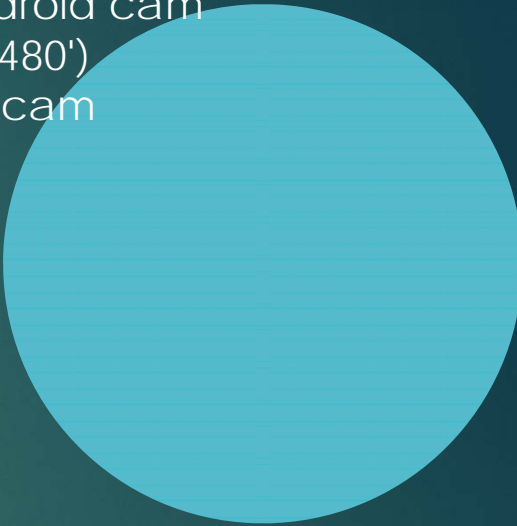


## 02. 비디오 프레임 캡처와 화면 표시

12

```
# 0208.py
import cv2
#cap = cv2.VideoCapture('http://192.168.0.21:4747/mjpegfeed') # droid cam
cap = cv2.VideoCapture('http://192.168.0.21:4747/mjpegfeed?640x480')
cap = cv2.VideoCapture('http://192.168.0.21:4747/video') # IP Webcam
frame_size = (int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)),
              int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))
print('frame_size =', frame_size)

while True:
    retval, frame = cap.read() # 프레임 캡처
    if not retval:
        break
    cv2.imshow('frame', frame)
    key = cv2.waitKey(25)
    if key == 27: # Esc
        break
if cap.isOpened():
    cap.release()
cv2.destroyAllWindows()
```



## 02. 비디오 프레임 캡처와 화면 표시

13

```
# 0209.py
'''
    pip install youtube_dl
    pip install pafy
'''

import cv2, pafy
url = 'https://www.youtube.com/watch?v=u_Q7Dkl7Alk'
video = pafy.new(url)
print('title = ', video.title)
print('video.rating = ', video.rating)
print('video.duration = ', video.duration)
best = video.getbest(preftype='webm') # 'mp4','3gp'
print('best.resolution', best.resolution)
cap=cv2.VideoCapture(best.url)
while(True):
    ....
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    edges = cv2.Canny(gray,100,200) #Canny edge detection
    cv2.imshow('edges',edges)
cv2.destroyAllWindows()
```



## 03. 비디오 파일 녹화

14

```
# 0210.py
import cv2
cap = cv2.VideoCapture(0) # 0번 카메라
frame_size = (int(cap.get(cv2.CAP_PROP_FRAME_WIDTH)),
              int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT)))
print('frame_size =', frame_size)
#fourcc = cv2.VideoWriter_fourcc(*'DIVX') # ('D', 'I', 'V', 'X')
fourcc = cv2.VideoWriter_fourcc(*'XVID')
out1 = cv2.VideoWriter('./data/record0.mp4',fourcc, 20.0, frame_size) #save a grey video
out2 = cv2.VideoWriter('./data/record1.mp4',fourcc, 20.0, frame_size,isColor=False) # save clr vdo
while True:
    retval, frame = cap.read()
    if not retval:
        break
    out1.write(frame)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    out2.write(gray)
    cv2.imshow('frame',frame)
    cv2.imshow('gray',gray)
    .....
```



# 04. matplotlib 비디오 디스플레이

15

```
# 0211.py
import cv2
import matplotlib.pyplot as plt
#1
def handle_key_press(event):
    if event.key == 'escape':
        cap.release()
        plt.close()
def handle_close(evt):
    print('Close figure!')
    cap.release()
#2 프로그램 시작
cap = cv2.VideoCapture(0) # 0번 카메라
plt.ion() # 대화모드 설정
fig = plt.figure(figsize=(10, 6)) # fig.set_size_inches(10, 6)
plt.axis('off')
#ax = fig.gca()
#ax.set_axis_off()
fig.canvas.set_window_title('Video Capture')
fig.canvas.mpl_connect('key_press_event', handle_key_press)
fig.canvas.mpl_connect('close_event', handle_close)
```

```
retval, frame = cap.read() # 첫 프레임 캡처
im = plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
```

```
#3
while True:
    retval, frame = cap.read() # 프레임 캡처
    if not retval:
        break
    # plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
    im.set_array(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
    fig.canvas.draw()
    # fig.canvas.draw_idle()
    fig.canvas.flush_events() # plt.pause(0.001)
    if cap.isOpened():
        cap.release()
```

# 04. matplotlib 비디오 디스플레이

16

```
# 0212.py
import cv2
import matplotlib.pyplot as plt
import matplotlib.animation as animation
# 프로그램 시작
cap = cv2.VideoCapture(0)
fig = plt.figure(figsize=(10, 6)) # fig.set_size_inches(10, 6)
fig.canvas.set_window_title('Video Capture')
plt.axis('off')
def init():
    global im
    retval, frame = cap.read() # 첫 프레임 캡처
    im = plt.imshow(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
def updateFrame(k):
    retval, frame = cap.read()
    if retval:
        im.set_array(cv2.cvtColor(frame, cv2.COLOR_BGR2RGB))
ani = animation.FuncAnimation(fig, updateFrame, init_func=init, interval=50)
plt.show()
if cap.isOpened():
    cap.release()
```



# 0213.py

.....

class Video:

def \_\_init\_\_(self, device=0):

self.cap = cv2.VideoCapture(device)

self.retval, self.frame = self.cap.read()

self.im = plt.imshow(cv2.cvtColor(self.frame, cv2.COLOR\_BGR2RGB))

print('start capture ...')

def updateFrame(self, k):

self.retval, self.frame = self.cap.read()

self.im.set\_array(cv2.cvtColor(camera.frame, cv2.COLOR\_BGR2RGB))

def close(self):

if self.cap.isOpened():

self.cap.release()

print('finish capture.') # 프로그램 시작

fig = plt.figure()

fig.canvas.set\_window\_title('Video Capture')

plt.axis("off")

camera = Video()

##camera = Video('./data/vtest.avi')

ani = animation.FuncAnimation(fig, camera.updateFrame, interval=50)

plt.show()

camera.close()





```
# 0214.py
```

```
...
```

```
class Video(animation.FuncAnimation):
```

```
    def __init__(self, device=0, fig=None, frames=None,
                  interval=50, repeat_delay=5, blit=False, **kwargs):
```

```
    if fig is None:
```

```
        self.fig = plt.figure()
```

```
        self.fig.canvas.set_window_title('Video Capture')
```

```
        plt.axis("off")
```

```
    super(Video, self).__init__(self.fig, self.updateFrame, init_func=self.init,
```

```
                                frames=frames, interval=interval, blit=blit,
```

```
                                repeat_delay=repeat_delay, **kwargs)
```

```
    self.cap = cv2.VideoCapture(device)
```

```
    print("start capture ...")
```

```
    def init(self):
```

```
        retval, self.frame = self.cap.read()
```

```
        if retval:
```

```
            self.im = plt.imshow(cv2.cvtColor(self.frame, cv2.COLOR_BGR2RGB))
```

```
    def updateFrame(self, k):
```

```
        retval, self.frame = self.cap.read()
```

```
        if retval:
```

```
            self.im.set_array(cv2.cvtColor(camera.frame, cv2.COLOR_BGR2RGB))
```

```
    #     return self.im,
```

```
    def close(self):
```

```
        if self.cap.isOpened():
```

```
            self.cap.release()
```

```
            print("finish capture.")
```

```
    # 프로그램 시작
```

```
    camera = Video()
```

```
    ##camera = Video('./data/vtest.avi')
```

```
    plt.show()
```

```
    camera.close()
```

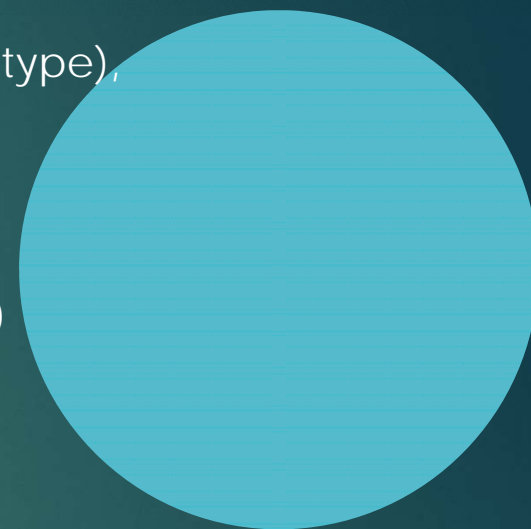
```
# 0215.py
import cv2
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.animation as animation
class Video(animation.FuncAnimation):
    def __init__(self, device=0, fig=None, frames=None,
                  interval=80, repeat_delay=5, blit=False, **kwargs):
        if fig is None:
            self.fig, self.ax = plt.subplots(1, 2, figsize=(10,5))
            self.fig.canvas.set_window_title('Video Capture')
            self.ax[0].set_position([0, 0, 0.5, 1])
            self.ax[0].axis('off')

            self.ax[1].set_position([0.5, 0, 0.5, 1])
            self.ax[1].axis('off')
        ##      plt.subplots_adjust(left=0,bottom=0,right=1,top=1,
        ##                               wspace=0.05,hspace=0.05)
        super(Video, self).__init__(self.fig, self.updateFrame, init_func=self.init,
                                     frames=frames, interval=interval, blit=blit,
                                     repeat_delay=repeat_delay, **kwargs)
        self.cap = cv2.VideoCapture(device)
        print('start capture ...')
```



```
# 0215.py
def init(self):
    retval, self.frame = self.cap.read()
    if retval:
        self.im0 = self.ax[0].imshow(cv2.cvtColor(self.frame, cv2.COLOR_BGR2RGB),
                                     aspect = 'auto')
        self.im1 = self.ax[1].imshow(np.zeros(self.frame.shape, self.frame.dtype),
                                     aspect = 'auto')
def updateFrame(self, k):
    retval, self.frame = self.cap.read()
    if retval:
        self.im0.set_array(cv2.cvtColor(self.frame, cv2.COLOR_BGR2RGB))

        gray = cv2.cvtColor(self.frame, cv2.COLOR_BGR2GRAY)
        self.im1.set_array(cv2.merge((gray,gray,gray)))
def close(self):
    if self.cap.isOpened():
        self.cap.release()
    print('finish capture.')
# 프로그램 시작
camera = Video()
plt.show()
camera.close()
```





## 04. matplotlib 비디오 디스플레이

21

