

[디지털 영상 처리] 과제3. Object recognition in videos

컴퓨터공학부 201911228 홍지우

Contents

[피부색으로 객체 추출 – color 사용]

1. Hand Video2.mov – 손 추출
2. Project –hand gesture.AVI – 얼굴 및 손 추출

[배경 제거로 움직이는 객체 추출]

1. Project_outdoor video1.mov – vehicles 추출
2. Car video2.mp4 – vehicles 추출

피부색으로 객체 추출 - color 사용

피부색으로 객체 추출 (Hand Video2.mov) – 추출 방식



1. 영상을 cv2.VideoCapture를 통해 가져온다.
2. 프레임마다 아래 과정을 수행한다.
 - Gaussian filter를 통해 이미지를 blurring한다.
 - 해당 프레임을 YCrCb로 색을 추출한다.
 - 피부색 범위를 포함하는 mask를 생성한다.
 - 프레임과 mask를 and연산하여 배경과 object를 구분한다.
 - 결과를 프레임별로 png파일 및 비디오로 저장하고, anaconda에 원본과 결과 이미지를 출력한다.

피부색으로 객체 추출 (Hand Video2.mov) - 소스코드

```
import cv2
import numpy as np
import ipywidgets as widgets
from ipywidgets import Box
import IPython.display as display
```

```
cap = cv2.VideoCapture('./example/Hand Video2.mov')
```

```
width = int(cap.get(3)) # 가로 길이
height = int(cap.get(4)) # 세로 길이
fps = 5 # 프레임 수
```

```
count = 0
```

```
# 동영상 output으로 저장할 때
```

```
fcc = cv2.VideoWriter_fourcc('D', 'I', 'V', 'X')
out = cv2.VideoWriter('output.avi', fcc, fps, (width, height))
```

```
widget1 = widgets.Image(layout = widgets.Layout(border="solid"), width="50%")
widget2 = widgets.Image(layout = widgets.Layout(border="solid"), width="50%")
```

```
output = [widget1, widget2]
box = Box(children=output)
display.display(box)
```

```
while (cap.isOpened()) :
    ret, image = cap.read()
    if ret : #frame이 존재한다면
        #1. 블러링
        image = cv2.GaussianBlur(image, (5,5),0)

        #2. 피부 검출
        YCrCb = cv2.cvtColor(image, cv2.COLOR_BGR2YCrCb)
        mask_hand = cv2.inRange(YCrCb, np.array([0,120,77]), np.array([255,180,127]))
        mask_color = cv2.bitwise_and(image, image, mask=mask_hand)

        #3. 후처리
        mask_color = cv2.dilate(mask_color, None, 4)

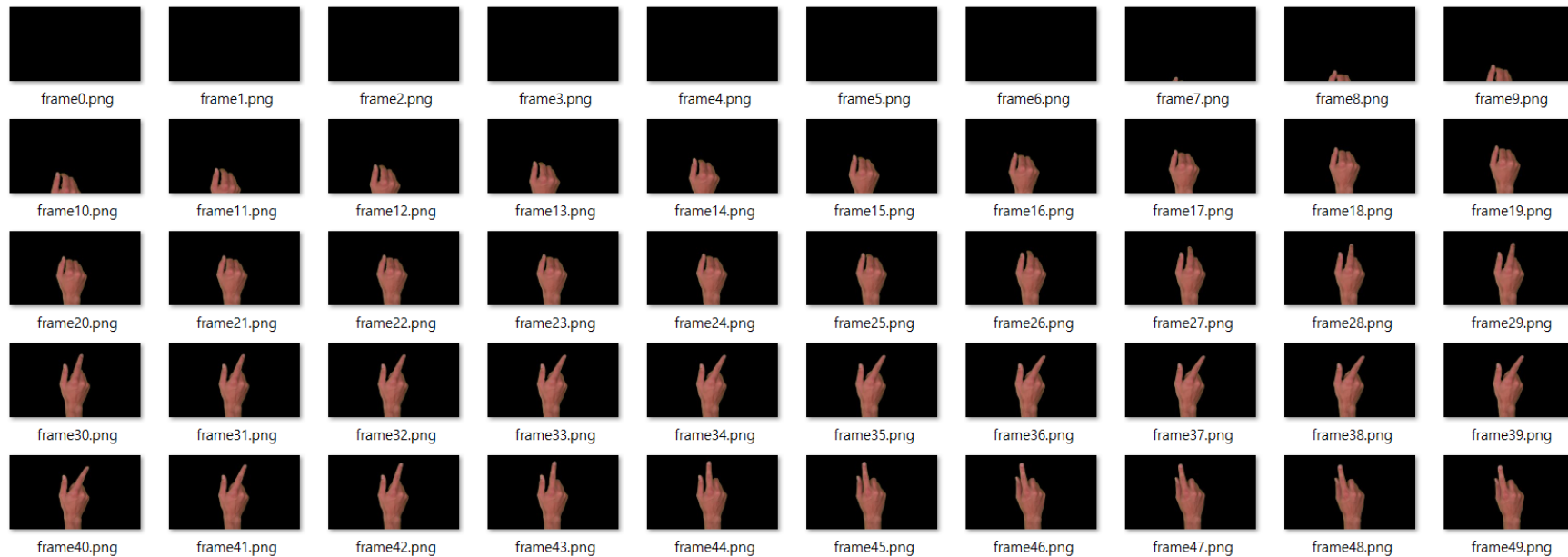
        #4. 이미지 저장 및 출력
        cv2.imwrite("./result/frame%d.png" % count, mask_color)
        count += 1
        out.write(mask_color)
        widget1.value = cv2.imencode(".jpeg", image)[1].tobytes()
        widget2.value = cv2.imencode(".jpeg", mask_color)[1].tobytes()

        if cv2.waitKey(1) & 0xFF == ord('q') : break
    else :
        print("finish")
        break

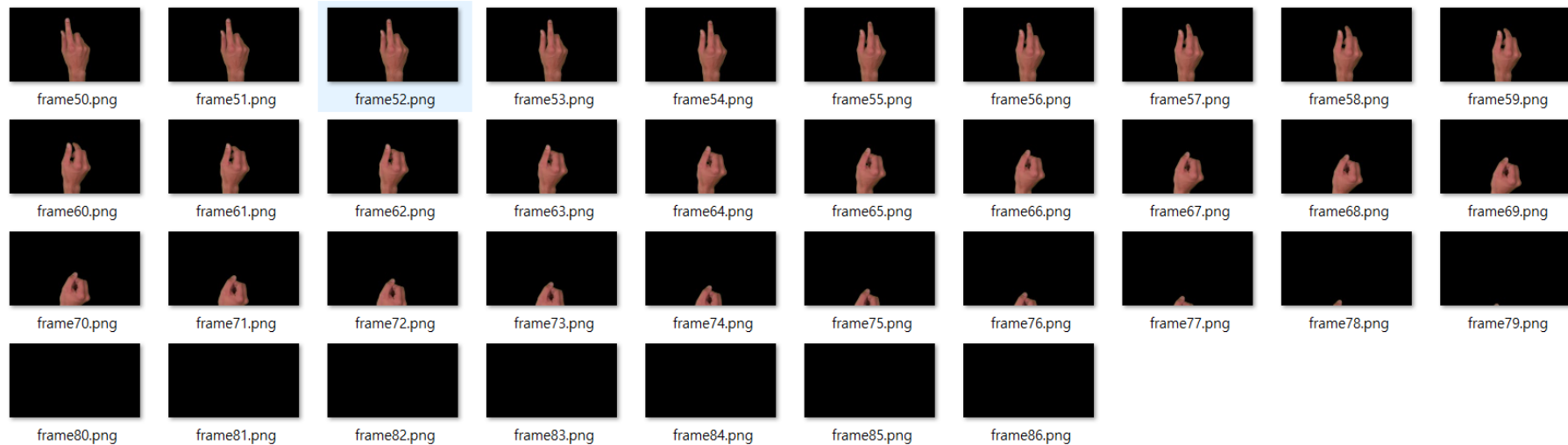
if cap.isOpened():
    cap.release()

if out.isOpened():
    out.release()
```

피부색으로 객체 추출 (Hand Video2.mov) – 결과1



피부색으로 객체 추출 (Hand Video2.mov) – 결과2



피부색으로 객체 추출 (Project- hand gesture.AVI) – 추출 방식



1. 영상을 cv2.VideoCapture를 통해 가져온다.
2. 프레임마다 아래 과정을 수행한다.
 - Gaussian filter를 통해 이미지를 blurring한다.
 - 해당 프레임을 YCrCb로 색을 추출한다.
 - 피부색 범위를 포함하는 mask를 생성한다.
(대비가 명확하지 않은 이미지이기 때문에 앞의 범위보다 좀더 타이트하게 범위를 수정)
 - 프레임과 mask를 and연산하여 배경과 object를 구분한다.
 - 결과를 프레임별로 png파일 및 비디오로 저장하고, anaconda에 원본과 결과 이미지를 출력한다.

피부색으로 객체 추출 (Project- hand gesture.AVI) - 소스코드

```
import cv2
import numpy as np
import ipywidgets as widgets
from ipywidgets import Box
import IPython.display as display
```

```
cap = cv2.VideoCapture('./example/Project- hand gesture.AVI')
```

```
width = int(cap.get(3)) # 가로 길이
height = int(cap.get(4)) # 세로 길이
fps = 5 # 프레임 수
```

```
count = 0
```

```
# 동영상들 output으로 저장할 때
```

```
fcc = cv2.VideoWriter_fourcc('D', 'I', 'V', 'X')
out = cv2.VideoWriter('output.avi', fcc, fps, (width, height))
```

```
widget1 = widgets.Image(layout = widgets.Layout(border="solid"), width="50%")
widget2 = widgets.Image(layout = widgets.Layout(border="solid"), width="50%")
```

```
output = [widget1, widget2]
box = Box(children=output)
display.display(box)
```

```
while (cap.isOpened()) :
    ret, image = cap.read()
    if ret : #frame01 존재한다면
        #1. 전처리
        image = cv2.GaussianBlur(image, (5,5),0)

        #2. 피부 검출
        YCrCb = cv2.cvtColor(image, cv2.COLOR_BGR2YCrCb)
        mask_hand = cv2.inRange(YCrCb,np.array([0,138,76]),np.array([255,175,127]))
        mask_color = cv2.bitwise_and(image,image, mask=mask_hand)

        #3. 후처리
        mask_color = cv2.erode(mask_color,None,1)

        #4. 이미지 저장 및 출력
        cv2.imwrite("./result/frame%d.png" % count, mask_color)
        count += 1
        out.write(mask_color)
        widget1.value = cv2.imencode(".jpeg", image)[1].tobytes()
        widget2.value = cv2.imencode(".jpeg", mask_color)[1].tobytes()

        if cv2.waitKey(1) & 0xFF == ord('q') : break
    else :
        print("finish")
        break

if cap.isOpened():
    cap.release()

if out.isOpened():
    out.release()
```

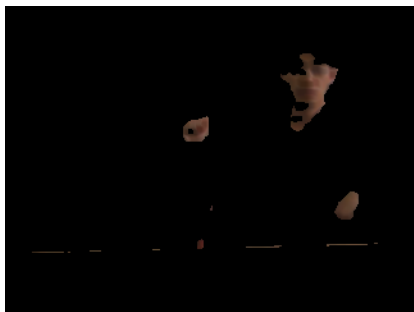
피부색으로 객체 추출 (Project- hand gesture.AVI) - 결과1



기본 동작



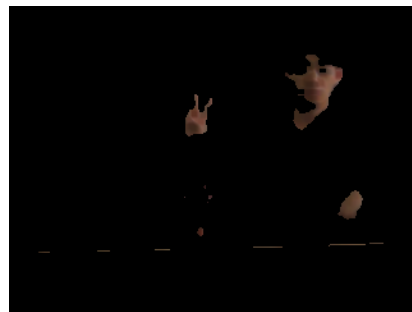
다섯 손가락 모두 펼치는 동작



주먹 동작



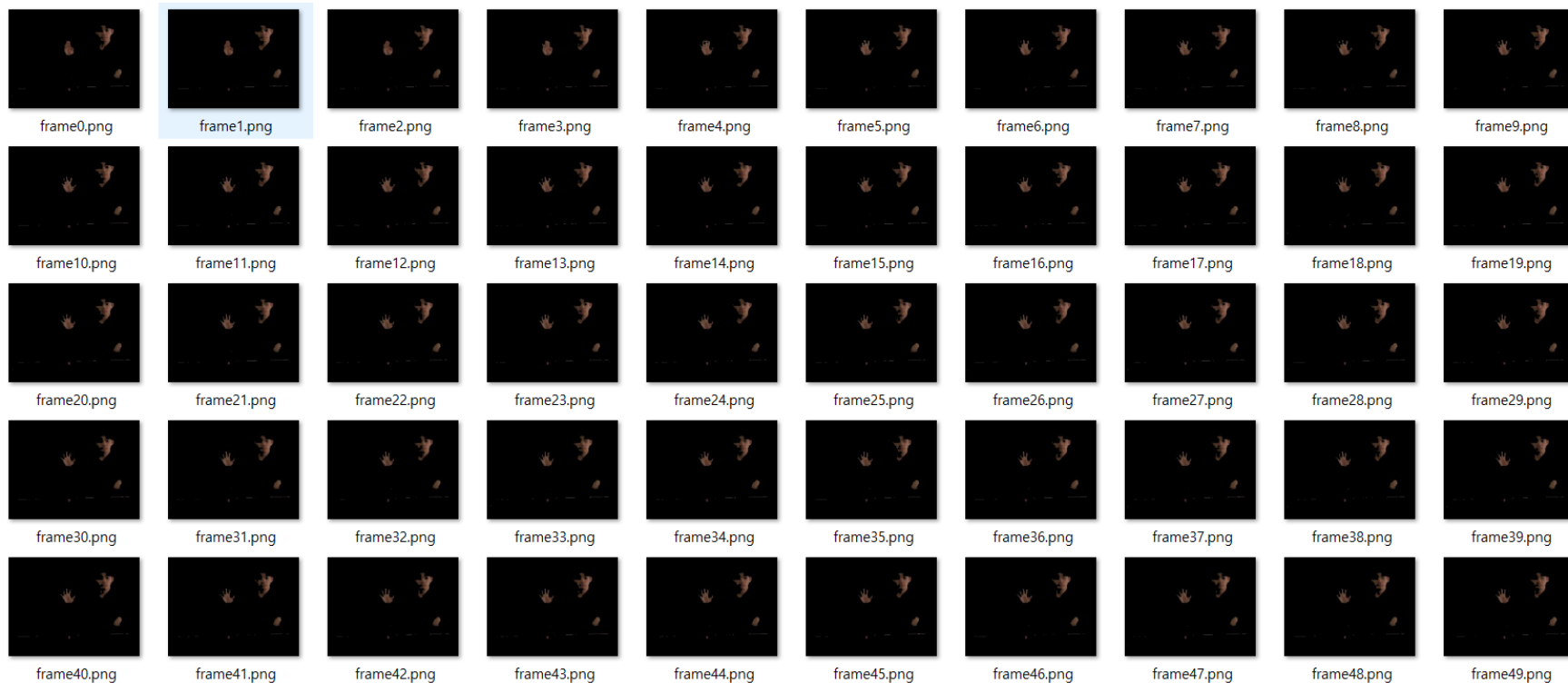
손가락 1개 펼치는 동작



손가락 2개 펼치는 동작

- 피부색으로 objec를 추출한다.
- 각 동작이 나타내는 object의 크기를 통해 동작을 구분할 수 있다.

피부색으로 객체 추출 (Project- hand gesture.AVI) – 결과2



frame50 ~ frame 777 생략

배경 제거로 움직이는 객체 추출

배경 제거로 움직이는 객체 추출 - 배경 추출 방식



input

output

[배경 추출]

1. 영상에서 프레임을 추출한다.
2. 각 프레임마다 다음 단계를 진행한다.
 - 현재 프레임의 각 픽셀에 값을 누적한다.
 - 누적된 픽셀을 count로 나누어 평균을 취한다.
 - 값을 8비트 unsigned int로 변경하여 저장한다.

배경 제거로 움직이는 객체 추출 - 배경 추출 소스코드

```
cap = cv2.VideoCapture('./example/Car video2.mp4')

width = int(cap.get(3)) # 가로 길이
height = int(cap.get(4)) # 세로 길이

back_img = np.zeros(shape=(height,width,3),dtype=np.float32)

widget1 = widgets.Image(layout = widgets.Layout(border="solid"), width="50%")
widget2 = widgets.Image(layout = widgets.Layout(border="solid"), width="50%")
items = [widget1, widget2]
box = Box(children=items)
display.display(box)

count = 0

while (cap.isOpened()):
    try:
        ret, frame = cap.read()
        if not ret:
            break
        count += 1
        cv2.accumulate(frame,back_img)
        average_back = back_img/count
        result_img = cv2.convertScaleAbs(average_back)

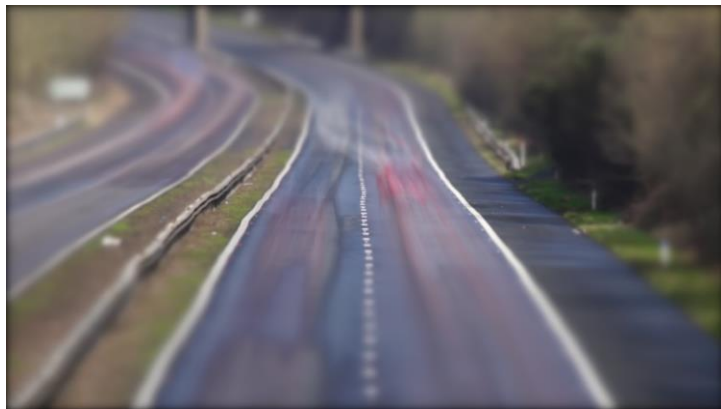
        widget1.value = cv2.imencode(".jpeg", frame)[1].tobytes()
        widget2.value = cv2.imencode(".jpeg", result_img)[1].tobytes()

    except KeyboardInterrupt: break

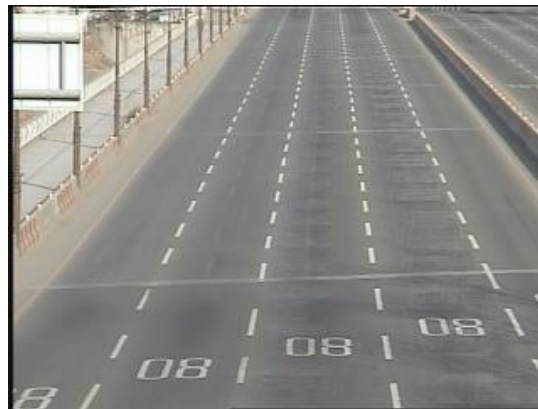
if cap.isOpened():
    cap.release()

cv2.imwrite('./background/back_img.png',result_img)
```

배경 제거로 움직이는 객체 추출 - 배경 추출 결과



Car video2.mp4 - 175 frame 결과



Project_outdoor video1.mov - 28 frame 결과

배경 제거로 움직이는 객체 추출 - 자동차 추출 방식



CCL



threshold



CCL



threshold

[자동차 추출]

1. background 이미지를 가져온다
2. 영상을 프레임으로 추출한다.
3. 각 프레임마다 다음 단계를 진행한다.
 - 현재 프레임과 background 이미지를 subtraction한다.
 - RGB마다 threshold를 취해 배경은 0으로 이미지는 1로 변환한다.
 - CCL을 통해 객체의 사각형 꼭짓점을 구한다.
 - 객체의 넓이가 일정 threshold보다 큰 경우 원본 영상에 rectangle을 그린다.

배경 제거로 움직이는 객체 추출 - 자동차 추출 소스코드

```
import cv2
import numpy as np
import ipywidgets as widgets
import IPython.display as display
from ipywidgets import Box
```

```
cap = cv2.VideoCapture('./example/Project_outdoor_video1.mov')

width = int(cap.get(3)) # 가로 길이
height = int(cap.get(4)) # 세로 길이

count = 0

back_img = cv2.imread('./background/back_img.png')
back_img = cv2.resize(back_img, dsize=(width,height), interpolation=cv2.INTER_AREA)

widget1 = widgets.Image(layout = widgets.Layout(border="solid"), width="50%")
widget2 = widgets.Image(layout = widgets.Layout(border="solid"), width="50%")

items = [widget1, widget2]
box = Box(children=items)
display.display(box)
kernel = np.ones((3,3), np.uint8)
kernel2 = np.ones((4,4), np.uint8)
```

```
while True:
    try:
        ret, frame = cap.read()
        if not ret:
            break
        sub_img = cv2.absdiff(frame, back_img)

        B,G,R = cv2.split(sub_img)
        ret,B = cv2.threshold(B, 35, 255, cv2.THRESH_BINARY)
        ret,G = cv2.threshold(G, 35, 255, cv2.THRESH_BINARY)
        ret,R = cv2.threshold(R, 35, 255, cv2.THRESH_BINARY)

        thres_img = cv2.bitwise_or(B,G)
        thres_img = cv2.bitwise_or(R, thres_img)

        thres_img = cv2.dilate(thres_img, kernel, 2)
        thres_img = cv2.erode(thres_img, kernel2, 1)

        box_round, temp = cv2.findContours(thres_img, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

        for i, now in enumerate(box_round):
            area = cv2.contourArea(now)
            if area > 110:
                x,y,width,height = cv2.boundingRect(now)
                cv2.rectangle(frame, (x,y), (x+width, y+height), (255, 0, 0), 1)

                widget1.value = cv2.imencode(".png", frame)[1].tobytes()
                widget2.value = cv2.imencode(".png", thres_img)[1].tobytes()
                cv2.imwrite("./result/frame%d.png" % count, frame)
                count += 1

    except KeyboardInterrupt:
        break

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

Car video2.mp4 는 위 코드와 유사하며, dilate와 erode를 좀 더 수정한 코드입니다.

배경 제거로 움직이는 객체 추출 - 자동차 추출 결과 (Project_outdoor video1.mov)



frame0.png



frame1.png



frame2.png



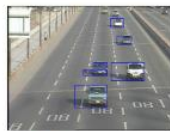
frame3.png



frame4.png



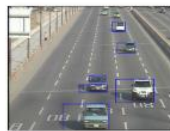
frame5.png



frame6.png



frame7.png



frame8.png



frame9.png



frame10.png



frame11.png



frame12.png



frame13.png



frame14.png



frame15.png



frame16.png



frame17.png



frame18.png



frame19.png



frame20.png



frame21.png



frame22.png



frame23.png



frame24.png



frame25.png



frame26.png



frame27.png

이후 frame은 생략

Color 및 Subtraction background를 사용했을 때 분석

성능 분석

color 사용 -> YCrCb 변환, mask와 AND연산, dilate 후처리
background 사용 -> background이미지 추출(average), subtraction,
R,G,B마다 threshold, dilate, erode연산 + CCL



실행 시간

color 사용 < background 사용

object 추출

color 사용 -> outside에서는 추출이 정확하지 않음
background 사용 -> outside에서도 잘 추출됨



object 추출 정확도

color 사용 < background 사용

“실행시간 성능만을 고려한다면 color를 추출 기준으로 사용”

“야외 영상이라면 background를 추출 기준으로 사용”

감사합니다