

Homework 03

202401833 신해솔

1. 필요한 라이브러리 import

```
import cv2
import numpy as np
from collections import deque
```

- OpenCV 라이브러리를 이용해 이미지를 읽고 출력
- numpy 라이브러리를 사용해 배열 연산을 수행함
- deque 모듈을 활용하여 이동 값 배열의 순환(회전) 처리를 간편하게 수행

2. 이미지 로드 및 move 변수 초기화

```
img_path = "cat.jpg"
fixed_img = cv2.imread(img_path, cv2.IMREAD_COLOR_BGR)

move = deque([0, 10, 0, 0])
```

- "cat.jpg" 파일을 OpenCV의 imread 함수를 통해 읽음 (BGR형식)
- 이미지 이동 효과를 주기 위해 각 방향으로 이동할 픽셀 수를 담은 deque 객체를 초기화함.

3. 이미지를 자르기 위한 준비

```
video = []
for i in range(4):
    info = np.array([0, 0, 0, 0])
    for j in range(10):
        background = np.zeros(shape=(450, 400, 3), dtype=np.uint8)
        img = fixed_img
```

- video 변수에 자른 이미지들 저장하기 위해 빈 리스트로 초기화
- move는 각 인덱스 변수에 더해질 값이므로 info 변수에 초기값들을 저장
- 각 방향 마다 초기화해야 하므로 바깥 for문에 위치
- 이후 각 방향에서 10px 이동할 때 마다 백그라운드(검정)와 이미지를 준비

4. 이미지 자르기 및 저장

```
info += np.array(move)
s1, e1, s2, e2 = info

background[e1:450-s1, e2:400-s2] = fixed_img[s1:450-e1, s2:400-e2]
video.append(background)
move.rotate(-1)
```

- 매 프레임마다 info에 현재 방향의 move 값을 더하여, 이미지의 잘라낼 영역을 결정
- s1(start1), e1(end1), s2, e2 변수에 각각 인덱싱 시작과 끝 값을 저장
- 시작과 끝 값에 맞추어 고정된 이미지를 자른 뒤, 백그라운드 ndarray에 저장
- 여기서, 잘라낸 영역을 배경에 넣을 때 인덱스의 시작과 끝을 반대로 사용하여, 잘린 부분이 반대쪽에 빈 공간(검은색 영역)으로 나타나도록 함
- move의 값을 한 칸 왼쪽으로 이동시키기 위해 deque의 rotate 함수 사용
- 이 과정을 통해 이미지가 아래, 위, 오른쪽, 왼쪽으로 움직이는 것처럼 출력됨

5. 비디오 저장 및 출력

```
np.save(file: "202401833.npy", video)
for img in video:
    cv2.imshow(winname: "cat", img)
    cv2.waitKey(250)
```

- 학번을 이름으로 video list를 ndarray 형태로 저장((40, 450, 400, 3))
- video 리스트에 있는 이미지를 for문을 통해 한 프레임씩 출력 (250ms간격)