

학번 / 이름	2171262 박소희
전공	IT융합공학부 사물인터넷트랙
주제	Homework #1 보고서

1) 소스코드

```

import cv2          #OPENCV 라이브러리 import
import torch
import numpy as np

class TensorManipulator: #클래스이므로 생성자 필요, 멤버 메서드 필요
    #TODO: 작성 필요
    #1) numpy array -> tensor 로 변환
    def __init__(self, img1, img2, img3): #생성자에서 초기화
        self.t_img1 = torch.tensor(img1)
        self.t_img2 = torch.tensor(img2)
        self.t_img3 = torch.tensor(img3)
        #print(self.t_img1.shape)
    #2) (480, 640, 3) 텐서 -> (3, 3, 480, 680) 크기의 텐서를 반환
    def concatenation(self):
        #각 이미지의 차원을 0을 기준으로 증가
        self.t_img1 = torch.unsqueeze(self.t_img1, dim = 0)
        self.t_img2 = torch.unsqueeze(self.t_img2, dim = 0)
        self.t_img3 = torch.unsqueeze(self.t_img3, dim = 0)
        #cat 이미지 텐서 결합
        tcat = torch.cat([self.t_img1, self.t_img2, self.t_img3], dim=0)
        #permute 차원 재배치
        tcat = tcat.permute(0,3,1,2)
        return tcat
    #3) (3, 3, 480, 640) 텐서 -> (3, 921600) 크기로 변환하여 반환
    def flatten(self, tcat):
        tshape = tcat.reshape([3, 921600])
        return tshape
    #4) 각 이미지 텐서들의 평균을 반환
    def average(self, tshape):
        tshape = tshape.type(dtype=torch.FloatTensor) #4) float 형 캐스팅
        return tshape.mean(dim = 1)
pass

```

```

if __name__ == "__main__":
    img1 = cv2.imread('./1.jpg', cv2.COLOR_BGR2RGB)
    img2 = cv2.imread('./2.jpg', cv2.COLOR_BGR2RGB)
    img3 = cv2.imread('./3.jpg', cv2.COLOR_BGR2RGB) # ndarray: (480, 640, 3)

    if (img1, img2, img3) is not None: # 이미지 보여주기
        cv2.imshow('1.jpg', img1)
        cv2.imshow('2.jpg', img2)
        cv2.imshow('3.jpg', img3)
        cv2.waitKey()
        cv2.destroyAllWindows()

    obj = TensorManipulator(img1, img2, img3)
    out = obj.concatenation()
    out_flt = obj.flatten(out)
    out_avg = obj.average(out_flt)

    print(out.shape)
    print(out_flt.shape)
    print(out_avg)

```

2) 소스코드 설명

- Opencv 라이브러리 사용

이미지를 불러오고 읽고 쓰는 opencv 라이브러리를 사용하기 위해 터미널에

\$ pip install opencv-python 명령어를 현재 내가 사용할 conda 가상환경 방에 입력하였다. 또한 3개의 이미지들을 프로젝트 폴더 패키지에 포함시켰다.

- 이미지 읽어보기

Cv2에 정의된 멤버함수 imread를 통해 현재 폴더 내에 있는 1, 2, 3.jpg 3개의 이미지 파일을 불러온다. COLOR_BGR2RGB는 매개변수 옵션으로 BGR 컬러 이미지를 RGB 컬러 이미지로 반환하겠다는 뜻이다.

- 이미지 출력하기

읽어온 3개의 이미지 파일을 멤버함수 `imshow`를 통해 매개변수로 주어진 이름으로 화면에 띄운다. `Waitkey()`는 사용자가 볼 수 있도록 기다리다가 키가 입력되면 `destroyAllWindows()`를 통해 창을 닫겠다는 뜻이다.

- Obj 객체 생성 및 생성자 호출

`obj = TensorManipulator(img1, img2, img3)` 를 통해 객체 `obj`를 선언 및 초기화 하였다. 객체를 생성함과 동시에 `TensorManipulator` 클래스에 정의된 `__init__` 생성자 함수가 실행되면서 객체의 `stack` 메모리 공간이 점유된다. 생성자 함수를 살펴보면 매개변수로 받은 이미지 3개를 `tensor`로 변환하였다. 즉, `numpy array` 타입이었던 이미지를 `torch.tensor()`를 통해 `tensor`형으로 캐스팅 해주었다.

- 멤버 함수 `Concatenation()` 호출

객체의 멤버함수인 `concatenation()`을 호출하여 반환된 값을 변수 `out`에 할당하였다. 해당 함수에서는 (480, 640, 3) 모양의 텐서를 (3, 3, 480, 680) 모양으로 반환해주는 역할을 한다. 우선, 텐서 차원이 증가했으므로 각 이미지 텐서의 차원을 `torch.unsqueeze(이미지, dim = 0)`를 통해 증가시켜주었다. 이후의 텐서는 (1, 480, 640, 3)의 모양을 갖는다. 그리고 3개의 텐서를 0차원을 기준으로 결합해주기 위해 `torch.cat`을 통해 `dim=0`을 기준으로 연결해준다. 그러면 (3, 480, 640, 3)의 모양을 갖게 된다. 우리가 원하는 모양은 (3, 3, 480, 680)의 텐서이므로 `.permute(0, 3, 1, 2)` 를 통해 각 차원을 재배치해주고 반환된 텐서 값을 리턴한다.

- 멤버 함수 `flatten(out)` 호출

`Flatten(out)`을 호출하여 (3, 3, 480, 640) 크기의 텐서를 (3, 921600) 크기의 2차원 텐서로 반환해준다. `Concatenation()`함수의 반환값인 `out`을 매개변수로 넣어주었다. 단순히 텐서의 모양을 변경해주는 역할을 하므로 `reshape()` 메소드를 활용하였다. `.reshape([3, 921600])`으로 나온 결과값을 반환한다.

- 멤버 함수 `average(out_ft)` 호출

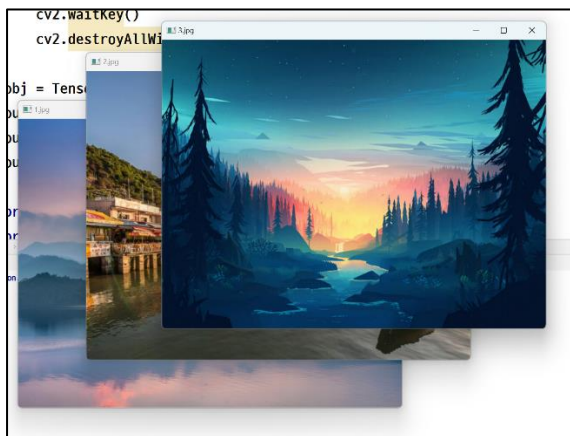
마찬가지로 이전 함수를 호출하여 반환된 값을 매개변수로 넣어 `average()` 함수를 호출한다. `Average()`에서는 2d 모양 텐서의 평균값을 반환하는 역할을 한다.

각 행에는 1, 2, 3.jpg 3개 이미지의 rgb텐서값들이 일렬로 나열되어 있다. 평균값은 실수형으로 반환되므로 `.type()` 메소드를 통해서 dtype을 `FloatTensor`로 바꾸어준다. 그리고 각 행의 평균값을 구하는 것이기 때문에 `.mean(dim=1)`의 값을 반환하면 된다.

3) 결과 캡처

```
main (2) x
C:\ProgramData\Anaconda3\envs\my_python\python.exe C:\Users\smile\pycharm\aip1\hw1\main.py
torch.Size([3, 3, 480, 640])
torch.Size([3, 921600])
tensor([144.0534, 136.8974, 85.5971])

Process finished with exit code 0
```



각 멤버함수를 호출하여 반환된 값의 텐서 모양을 각각 print 해보면 다음과 같이 출력이 된다. Opencv 라이브러리를 사용하여 이미지 값을 읽고 텐서의 모양을 바꾸어보는 실습을 해보았다. Pytorch에 정의된 다양한 라이브러리 및 함수들을 이용하여 간편하게 텐서를 다루어 볼 수 있었던 재밌는 실습이었다.