

파이썬프로그래밍

문제 2: 영상 파일 검색 및 출력 프로그램

2020년 12월 10일 목

컴퓨터공학과

2019305059

이현수

■ 문제 2: 영상 파일 검색 및 출력 프로그램

1. 디렉토리 안에 존재하는 영상 파일의 수와 총 용량을 출력하기 스트링 변수, src_folder로 지정된 폴더 안에(서브 폴더 포함) 존재하는 모든 파일 중에서 영상 파일을 찾아 그 파일들의 개수와 이름과 파일의 크기의 총합을 파악하여 출력.
2. 그림 파일 중에서 1) 용량이 제일 큰 파일과, 2) 용량이 제일 적은 파일, 3) 영상의 높이가 높은 파일, 4) 영상의 가로가 제일 넓은 파일 총 4개의 사진을 2x2 서브화면에 출력한다. 1번과 2번 서브 화면의 타이틀에는 그림과 같이 "파일이름:파일용량"을 보이고, 3번과 4번 서브 화면에는 "화면이름:가로x세로" 정보를 타이틀에 출력한다. 파일 이름에는 경로는 포함하지 않아야 하고, 확장자까지 보여야함.

■ 코드

```
1  src_folder = 'mission/'
2
3  import os
4  import time
5  import numpy as np
6  import matplotlib.image as img
7  import matplotlib.pyplot as plt
8
9  sz = os.path.getsize(__file__)
10 print(f'Program Size = {sz:#,}\n')#단계1. 파일 사이즈 출력(주석문포함 크기)
11
12 def print_num(num):
13     str_num = str(num) #정수를 문자열로 변환
14     num_len = len(str_num) #문자열 길이
15     if num_len < 4: print(str_num, end='') #단계2. 1,000 단위 쉼표(,) 넣기 ...문자열길이가 4미만이라면 그냥출력
16     else: #문자열길이가 4이상이라면
17         numlist=list(str_num) #문자열을 리스트로 변환
18         index = num_len - 3
19         numlist.insert(index,",") #, 삽입
20         while True:
21             index = index - 3 #인덱스 3빼기
22             if index<=0: #인덱스가 맨앞자리거나 음수면 반복문 빠져나감
23                 break
24             else:
25                 numlist.insert(index,",")#아니면 인덱스에 , 삽입
26         str_num2=''.join(numlist) #리스트를 문자열로 변환
27         print(str_num2, end='') #출력
28     if str_num == '0': #단계3. 10000 이하의 자릿수에 대한 아라비아 숫자와 한글표현 ... s
29         print('(') #()만 출력
30     elif num_len==1: #숫자길이가 1이라면
31         print('{0}'.format(str_num)) #({숫자} 출력
32     elif num_len==2: #숫자길이가 2라면
33         print('(', end='')
34         if str_num[0]!='1': print(str_num[0], end='')#십의자리가 1이 아닐때 십의자리 숫자출력
35         print('십',end='') #'십'출력
36         if str_num[1]!='0': print(str_num[1], end='')#일의자리가 0이 아닐때 일의자리 숫자출력
37         print(')')
```

파일사이즈 출력(주석문포함 크기)

문제1. Print_num함수 코드
총파일 사이즈 출력시 이용

```

38 elif num_len==3: #숫자길이가 3 이라면
39     print('(', end='')
40     if str_num[0]!='1': print(str_num[0], end='') #백의자리가 '1'이 아니면 백의자리 숫자 출력
41     print('백', end='') #백 출력
42     if str_num[1]!='0': #십의자리숫자가 '0'이 아닐때
43         if str_num[1]=='1': print('십', end='') #십의자리숫자가 '0'이 아니면서 '1'일때 '십'만 출력
44         else : print(str_num[1]+'십', end='') #그외 숫자+ '십' 형태로 출력
45     if str_num[2]!='0': print(str_num[2], end='') #일의자리숫자가 '0'이 아니면 일의자리숫자 그대로 출력
46     print(')')
47 elif num_len==4: #숫자길이가 4 라면
48     print('(', end='')
49     if str_num[0] != '1': print(str_num[0], end='') #천의자리숫자가 '1'이 아니면 천의자리 숫자 출력
50     print('천', end='') #천 출력
51     if str_num[1] != '0': #백의자리가 '0'이 아닐때
52         if str_num[1]=='1': print('백', end='') #백의자리숫자가 '0'이 아니면서 '1'일때 '백'만 출력
53         else: print(str_num[1] + '백', end='') #그외 숫자+ '백' 형태로 출력
54     if str_num[2] != '0': #십의자리숫자가 '0'이 아닐때
55         if str_num[2]=='1': print('십', end='') #십의자리숫자가 '0'이 아니면서 '1'일때 '십'만 출력
56         else : print(str_num[2] + '십', end='') #십의자리숫자가 '0'이 아니면 숫자+ '십' 형태로 출력
57     if str_num[3] != '0': print(str_num[3], end='') #일의자리숫자가 '0'이 아니면 일의자리숫자 그대로 출력
58     print(')')
59 else: #숫자길이가 5 이상이라면
60     print('(', end='')
61     if(num_len==5): #숫자길이가 5 일경우
62         if str_num[0]!='1': print(str_num[0], end='') #만의자리수사가 '1'이 아니면 만의자리숫자출력
63         print('만', end='') #만 출력
64     else: print(str_num[:len(str_num)-4] + '만', end='') #길이6 이상일때 인덱스0 부터 만의자리숫자까지 출력후 '만' 출력
65     if str_num[len(str_num)-4] != '0': #천의자리가 '0'이 아닐때
66         if str_num[len(str_num)-4]=='1': print('천', end='') #천의자리숫자가 '0'이 아니면서 '1'일때 '천'만 출력
67         else: print(str_num[len(str_num)-4] + '천', end='') #천의자리숫자가 '0'이 아니면 숫자+ '천' 형태로 출력
68     if str_num[len(str_num)-3] != '0': #백의자리가 '0'이 아닐때
69         if str_num[len(str_num)-3]=='1': print('백', end='') #백의자리숫자가 '0'이 아니면서 '1'일때 '백'만 출력
70         else: print(str_num[len(str_num)-3] + '백', end='') #백의자리숫자가 '0'이 아니면 숫자+ '백' 형태로 출력
71     if str_num[len(str_num)-2] != '0': #십의자리숫자가 '0'이 아닐때
72         if str_num[len(str_num)-2]=='1': print('십', end='') #십의자리숫자가 '0'이 아니면서 '1'일때 '십'만 출력
73         else: print(str_num[len(str_num)-2] + '십', end='') #십의자리숫자가 '0'이 아니면 숫자+ '십' 형태로 출력
74     if str_num[len(str_num)-1] != '0': print(str_num[len(str_num)-1], end='') #일의자리숫자가 '0'이 아니면 숫자 그대로 출력
75     print(')')

```

문제1. Print_num함수 코드
총파일 사이즈 출력시 이용

```

77 s_time = time.time() #단계2. 시간측정 시작
78
79 lst_jpg = [] #jpg 사진이 담길 리스트
80 lst_png = [] #png 사진이 담길 리스트
81 lst_tif = [] #tif 사진이 담길 리스트
82 totalFilesize = 0 #총 파일 사이즈 저장 변수
83 for i, (path, sub_dir, files) in enumerate(os.walk(src_folder)):
84     for filename in files:
85         ext = os.path.splitext(filename)[-1] # 파일 확장자를 반환함.
86         ext_and_lst = zip(['.jpg', '.png', '.tif'], [lst_jpg, lst_png, lst_tif])
87         for ext_nm, lst_n in ext_and_lst:
88             if ext_nm == ext.lower(): # 소문자로 바꾸어 비교한다.
89                 a = os.path.join(path, filename)
90                 b = os.path.getsize(path+"/"+ filename)
91                 totalFilesize = totalFilesize + b
92                 lst_temp = [a,b]
93                 lst_n.append(lst_temp)
94             break
95 #단계3. 파일 수, 확장자별 파일 개수 출력
96 print('Total number of picture files : {0}(jpg:{1}, png:{2}, tif:{3})'.format(
97     len(lst_jpg)+len(lst_png)+len(lst_tif), len(lst_jpg), len(lst_png), len(lst_tif))
98 print('Total size of files: ', end='')
99 print_num(totalFilesize)

```

src_folder 위치의
jpg, png, tif
확장자 파일 정보
얻기.
파일수, 확장자별
파일수, 총 파일
사이즈 출력

4가지 영상 이미지 출력 설정 및 출력

■ 코드설명

```
1 src_folder = 'mission/'
2
3 import os
4 import time
5 import numpy as np
6 import matplotlib.image as img
7 import matplotlib.pyplot as plt
8
9 sz = os.path.getsize(__file__)
10 print(f'Program Size = {sz:#,}\n') #단계1. 파일 사이즈 출력(주석문 포함)
11
```

최상단에 src_folder에 분석할 폴더 위치를 적는다. 그리고 os모듈을 사용해 코드파일 크기를 출력한다.

(코드12-75) print_num함수(문제1번과 동일한 코드) 파일크기의 총합을 아래사진처럼 출력하기 위함.

```
Total number of picture files: 32(jpg: 25, png:5, tif:2)
Total size of files: 31,408,029(3140만5천2십9)
```

```
77 s_time = time.time() #단계2. 시간측정 시작
78
79 lst_jpg = [] #jpg사진이 담길 리스트
80 lst_png = [] #png사진이 담길 리스트
81 lst_tif = [] #tif사진이 담길 리스트
82 totalFileSize = 0 #총 파일 사이즈 저장 변수
83 for i, (path, sub_dir, files) in enumerate(os.walk(src_folder)):
84     for filename in files:
85         ext = os.path.splitext(filename)[-1] # 파일 확장자를 반환함.
86         ext_and_lst = zip(['.jpg', '.png', '.tif'], [lst_jpg, lst_png, lst_tif])
87         for ext_nm, lst_n in ext_and_lst:
88             if ext_nm == ext.lower(): # 소문자로 바꾸어 비교한다.
89                 a = os.path.join(path, filename)
90                 b = os.path.getsize(path+"/"+filename)
91                 totalFileSize = totalFileSize + b
92                 lst_temp = [a,b]
93                 lst_n.append(lst_temp)
94             break
95 #단계3. 파일 수, 확장자별 파일 개수 출력
96 print('Total number of picture files : {0}(jpg:{1}, png:{2}, tif:{3})'.format(
97     len(lst_jpg)+len(lst_png)+len(lst_tif), len(lst_jpg),len(lst_png),len(lst_tif))
98 print('Total size of files: ',end='')
99 print_num(totalFileSize)
100
```

시간측정을 위해 s_time = time.time() 코드를 작성한다.

Jpg, png, tif 확장자 파일 정보를 담을 리스트 lst_jpg, lst_png, lst_tif를 만든다.

for 반복문에서 os모듈의 walk을 이용해 폴더를 분석한다. 그리고 그 중에서 files를 뽑아내 또 for반복문을 돌린다. os.path.splitext(filename)[-1]을 통해 파일 확장자를 얻는다. 이런식으로 a변수에는 파일위치와 이름을 저장하고 변수 b에는 os.path.getsize(path+"/"+filename)을 통해서 파일 사이즈를 저장한다. 그리고 totalFileSize에 b를 누적 저장한다. 그 후 a와 b를 lst_temp리스트에 원소로 넣고 lst_temp를 해당확장자 리스트에 append 한다. 그결과 src_folder위치의 모든 하위폴더에 있는 jpg, png, tif 확장자 파일 정보를 해당 리스트에 저장시킬 수 있다.

전체 파일수, 확장자별 파일 개수를 출력하고, 총 파일 크기의 총합을 print_num함수를 이용해 출력한다.

```

101 lst = [lst_jpg, lst_png, lst_tif] #lst리스트에 확장자별 사진정보가 있는리스트 저장
102 sizeMax = ['', 0] #사이즈가 가장 큰 사진을 가정해 이름, 사이즈 두개정보를 sizeMax 리스트에 저장
103 sizeMin = ['', 999999999999999999] #사이즈가 가장 작은 사진을 가정해 이름, 사이즈 두개정보를 sizeMin 리스트에 저장
104 for i in lst:
105     for j in i:
106         if j[1]>sizeMax[1]: #사이즈가 더 크게 발견되면
107             sizeMax[0]=j[0] #sizeMax[0]에 j[0] 즉 사진 이름 저장
108             sizeMax[1]=j[1] #sizeMax[1]에 j[1] 즉 사진 사이즈 저장
109 for i in lst:
110     for j in i:
111         if j[1]<sizeMin[1]: #사이즈가 더 작으게 발견되면
112             sizeMin[0]=j[0] #sizeMin[0]에 j[0] 즉 사진 이름저장
113             sizeMin[1]=j[1] #sizeMin[1]에 j[1] 즉 사진 사이즈 저장
114
115 img1 = img.imread(sizeMax[0]) #사이즈가 가장 큰 사진읽기
116 img2 = img.imread(sizeMin[0]) #사이즈가 가장 작은 사진 읽기
117

```

lst 리스트 안에 lst_jpg, lst_png, lst_tif를 저장한다. sizeMax, sizeMin리스트 첫번째 원소는 사진이름, 두번째 원소는 사이즈가 들어간다. sizeMax, sizeMin 첫번째 원소 초기값은 ""으로, sizeMax 두번째 원소는 0, sizeMin 두번째 원소는 나중에 비교할 때 제약이 없을 정도로 큰 정수값을 저장한다.

(코드104-108) (코드109-113) lst에 있는 모든 리스트를 i에 담아서 for 반복문으로 반복하고 i가 비어있지 않다면 for반복문에서 i안에 있는 리스트 j를 담아서 분석한다. 용량이 가장 큰 것을 구할때에는 if 조건문을 통해 j[1]이 sizeMax[1]보다 크면 sizeMax리스트를 갱신한다. 용량이 가장 작은 것을 구할때는 if 조건문을 통해 j[1]이 sizeMin[1]보다 작으면 sizeMin리스트를 갱신한다.

sizeMax[0]영상을 imread해 img1에 담고, sizeMin[0]영상을 imread해 img2에 담는다.

```

118 temp=0 #비교변수
119 tall=['',0,0] #세로길이가 가장 긴 사진정보를 담은 리스트
120 for i in lst:
121     if len(i)!=0: #리스트i가 비어있지 않다면
122         for j in i:
123             temp_img = img.imread(j[0]) #반복문으로 모든 사진들 읽어오기
124             n = np.array(temp_img.shape) #n에 읽어온사진 shape를 np.array를 통해 변환해 저장
125             if (temp < n[0]): #그중에 세로길이가 더 크게 발견되면
126                 tall[0] = j[0] #tail[0]에 j[0] 즉 사진 이름저장
127                 tall[1] = n[0] #tail[1]에 n[0] 즉 사진 세로길이 저장
128                 tall[2] = n[1] #tail[2]에 n[1] 즉 사진 가로길이 저장
129                 temp=n[0] #비교변수를 갱신

```

세로길이가 가장 긴 사진정보를 찾기위한 코드이다. temp는 비교변수로 0으로 초기화한다.

세로길이가 가장 긴 영상정보를 담기위한 tall리스트를 만든다. 첫번째 원소는 "", 두번째, 세번째 원소는 0으로 저장한다. 첫번째 원소에는 영상이름이, 두번째,세번째에는 영상 세로, 가로 길이가 저장된다.

(코드120-129) lst에 있는 모든 리스트를 i에 담아서 for 반복문으로 반복하고 i가 비어있지 않다면 for반복문에서 i안에 있는 리스트 j를 담아서 분석한다. j[0]에는 사진 이름이 있어 imread해 temp_img에 담는다. 그리고 temp_img.shape를 np.array시켜 n을 만든다. 만약 비교변수 temp보다 n[0]즉 세로길이가 더 크면 tall리스트에 각 정보를 담고 temp비교변수를 n[0]로 갱신한다.


```

130 temp=0 #비교변수
131 wide=['',0,0]#가로길이가 가장 긴 사진정보를 담을 리스트
132 for i in lst:
133     if len(i)!=0:#리스트i가 비어있지 않다면
134         for j in i:
135             temp_img = img.imread(j[0]) #반복문으로 모든 사진들 읽어오기
136             n = np.array(temp_img.shape) #n에 읽어온사진 shape를 np.array를 통해 변환해 저장
137             if (temp < n[1]):#그중에 가로길이가 더 크게 발견되면
138                 wide[0] = j[0] #wide[0]에 j[0]즉 사진 이름저장
139                 wide[1] = n[0] #wide[1]에 n[0]즉 사진 세로길이 저장
140                 wide[2] = n[1] #wide[2]에 n[1]즉 사진 가로길이 저장
141                 temp=n[1] #비교변수를 갱신
142
143 img3=img.imread(tall[0])#세로길이가 가장 긴 사진 읽기
144 img4=img.imread(wide[0])#가로길이가 가장 긴 사진 읽기
145
146 c_time = time.time()#시간측정 중단
147 print('\nTotal Execution Time = {0:.4f}[sec]'.format(c_time - s_time)) #소요시간 출력

```

가로길이가 가장 긴 사진정보를 찾기위한 코드이다. temp는 비교변수로 0으로 초기화한다.

가로길이가 가장 긴 영상정보를 담기위한 wide리스트를 만든다. 첫번째 원소는 "", 두번째, 세번째 원소는 0으로 저장한다. 첫번째 원소에는 영상이름이, 두번째,세번째에는 영상 세로, 가로 길이가 저장된다.

(코드132-141) lst에 있는 모든 리스트를 i에 담아서 for 반복문으로 반복하고 i가 비어있지 않다면 for반복문에서 i안에 있는 리스트 j를 담아서 분석한다. j[0]에는 사진 이름이 있어 imread해 temp_img에 담는다. 그리고 temp_img.shape를 np.array시켜 n을 만든다. 만약 비교변수 temp보다 n[1]즉 가로길이가 더 크면 wide리스트에 각 정보를 담고 temp비교변수를 n[1]로 갱신한다.

그 후 tall[0]의 영상을 imread해 img3에 담고, wide[0]의 영상을 imread해 img4에 담는다.

그 후 시간측정을 중단하고 소요시간을 출력한다.

```

148 #단계4. 그림 출력
149 ch='\\'
150 plt.figure(figsize=(15, 8), num='prob2') #창사이즈를 가로:15, 세로:8로 설정. 창 이름은 'prob2'로 설정
151 plt.subplot(221) # 2x2 배열의 subplot 창에서 1번째 창 지정
152 plt.imshow(img1)
153 plt.title(f"max=\",{sizeMax[0][sizeMax[0].rfind(ch)+1]}\"{sizeMax[1]:#},\")#해당창의 타이틀
154 plt.axis('off') # 영상의 경계가 보이지 않는다.
155
156 plt.subplot(222) # 2x2 배열의 subplot 창에서 2번째 창 지정
157 plt.imshow(img2)
158 plt.title(f"min=\",{sizeMin[0][sizeMin[0].rfind(ch)+1]}\"{sizeMin[1]:#},\")#해당창의 타이틀
159 plt.axis('off') # 영상의 경계가 보이지 않는다.
160
161 plt.subplot(223) # 2x2 배열의 subplot 창에서 3번째 창 지정
162 plt.imshow(img3)
163 plt.title('tall='+''+tall[0][tall[0].rfind('\\')+1]+'\"('+str(tall[1])+', '+str(tall[2]))')#해당창의 타이틀
164 plt.axis('off') # 영상의 경계가 보이지 않는다.
165
166 plt.subplot(224) # 2x2 배열의 subplot 창에서 4번째 창 지정
167 plt.imshow(img4)
168 plt.title('wide='+''+wide[0][wide[0].rfind('\\')+1]+'\"('+str(wide[1])+', '+str(wide[2]))')#해당창의 타이틀
169 plt.axis('off') # 영상의 경계가 보이지 않는다.
170 plt.show()#사진출력

```

figsize = (15,8)을 통해 창 크기를 가로 15, 세로 8로 설정하고 타이틀바 이름은 'prob2'로 설정한다.

4가지 영상을 2*2배열의 subplot 창에서 1,2,3,4번째 창에 지정한다.

마지막에 plt.show()를 통해 창을 띄우고 사진을 출력한다. 그리고 각각 plt.imshow(img1)...을 하고 subplot의 title을 문제대로 출력시키고 각각 plt.axis('off')를 통해 영상의 경계를 보이지 않게 한다.

Plt.show()를 해 창을 띄워 영상사진을 출력한다.

■ 실행

```
이현수2 ×
"C:\Program Files\Python38\python.exe" C:/Users/nec/Desktop/파이썬프로그래밍과제2/prob2/이현수2.py
Program Size = 10,211

Total number of picture files : 6(jpg:4, png:2, tif:0)
Total size of files: 810,637(81만6백3십7)

Total Execution Time = 0.0688[sec]

Process finished with exit code 0
```

파일크기, 파일 개수, 확장자별 개수, 파일 사이즈, 실행시간이 모두 정상적으로 출력된다.



사진도 정상적으로 출력된다.

■교훈 및 시행착오

이미지를 로드할 때 cv2 모듈을 이용해 imread를 했을 때 gif이미지를 읽어오지 못해서 `import matplotlib.image as img` 를 사용해 이미지를 읽었더니 정상적으로 됐다.