# 수치해석 HW11

2018008613 안상욱

# 1. 이미지 수집

- 다음과 같은 10장의 이미지를 구해서 picture1 , .... , picture10 으로 저장했습니다.










# 2. 코드 작성

- 먼저 그림과 같이 그림을 읽어온 뒤 BGR 형식으로 n이라는 변수에 저장해 놓습니다.
- 그 후 yuv라는 변수에 BGR 형식으로 읽어온 것을 YUV 형식으로 변환하여 저장합니다.
- 2중 for문을 통해 RGB의 R, G, B 원소를 나누고, Y, U, V 원소를 각각 나누어 줍니다.
- Y, U, V 원소에 각각의 Y, U, V값을 1차원 배열에 저장합니다.
- Cv2.imshow를 이용해 Y, U, V 각각의 이미지를 출력합니다.

```
n = cv2.imread("picture1.jpg")
yuv = cv2.cvtColor(n, cv2.COLOR_BGR2YUV)
yuv1 = copy.deepcopy(yuv)
yuv2 = copy.deepcopy(yuv)

xs = len(n)
ys = len(n[0])

n = np.array(n)
nr = np.zeros((xs,ys,3))
ng = np.zeros((xs,ys,3))
nb = np.zeros((xs,ys,3))
SR = np.zeros((xs,ys))
SG = np.zeros((xs,ys))
SB = np.zeros((xs,ys))
Y = []
U = []
V = []

for i in range(xs):
    for j in range(ys):
        SR[i][j] = n[i][j][2]
        SG[i][j] = n[i][j][1]
        SB[i][j] = n[i][j][0]
        nr[i][j][2] = n[i][j][2]
        ng[i][j][1] = n[i][j][1]
        nb[i][j][0] = n[i][j][0]
        yuv1[i][j][0] = yuv1[i][j][1]
        yuv1[i][j][1] = 0
        yuv1[i][j][2] = 0
        yuv2[i][j][0] = 0
        yuv2[i][j][1] = 0
        Y.append(yuv[i][j][0])
        U.append(yuv[i][j][1])
        V.append(yuv[i][j][2])

cv2.imshow('Y', yuv[:,:,0])
cv2.imshow('U', yuv1)
cv2.imshow('V', yuv2)
```

# 2. 코드 작성

- Plt.imshow를 이용해 R, G, B 각각의 이미지를 출력합니다.
- R, G, B 각각의 변수에 R, G, B 값을 1차원 배열에 저장합니다.

```
nr = nr.astype(np.int64)
ng = ng.astype(np.int64)
nb = nb.astype(np.int64)
nr = list(np.array(nr))
ng = list(ng)
nb = list(nb)

plt.imshow(nr)
plt.show()
plt.imshow(ng)
plt.show()
plt.imshow(nb)
plt.show()

SR = SR.astype(np.int64)
SG = SG.astype(np.int64)
SB = SB.astype(np.int64)
SR = list(SR)
SG = list(SG)
SB = list(SB)
R = []
for i in range(xs):
    for j in range(ys):
        R.append(SR[i][j])
G = []
for i in range(xs):
    for j in range(ys):
        G.append(SG[i][j])
B = []
for i in range(xs):
    for j in range(ys):
        B.append(SB[i][j])
```

# 2. 코드 작성

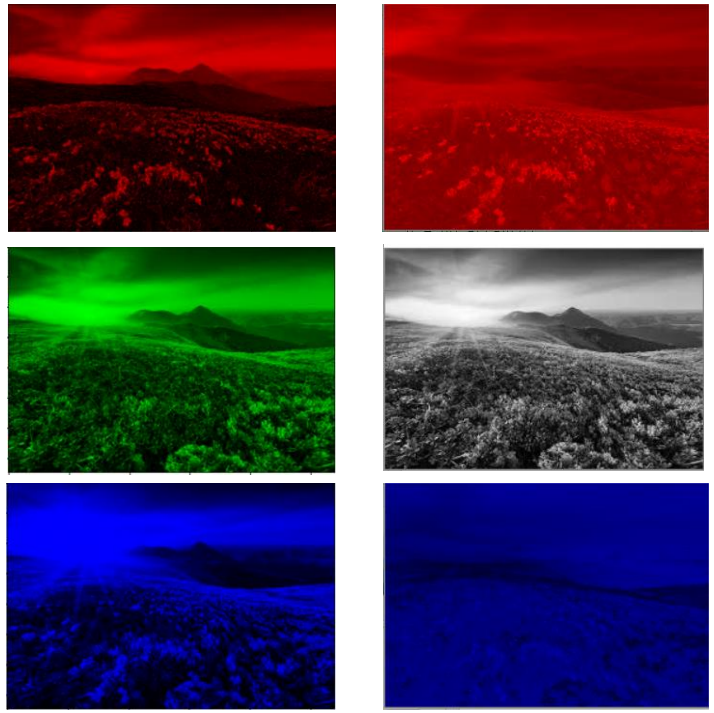
- Arr 배열에 R, G, B 값을 각각 담고 있는 1차원 배열을 넣고, yuvarr 배열에 Y, U, V 값을 각각 담고 있는 1차원 배열을 담은 뒤 pandas 모듈을 이용해 R과 G, G와 B, B와 R의 correlation coefficient와 Y와 U, U와 V, V와 Y의 correlation coefficient 값을 구해 출력해줍니다.

```
arr = [R, G, B]

df = pd.DataFrame(arr).T
corr = df.corr(method = 'pearson')
print(corr)

yuvarr = [Y, U, V]
yuvdf = pd.DataFrame(yuvarr).T
yuvcorr = yuvdf.corr(method = 'pearson')
print(yuvcorr)
```

# 3. 출력 결과

- Picure1의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient  값을 의미합니다.

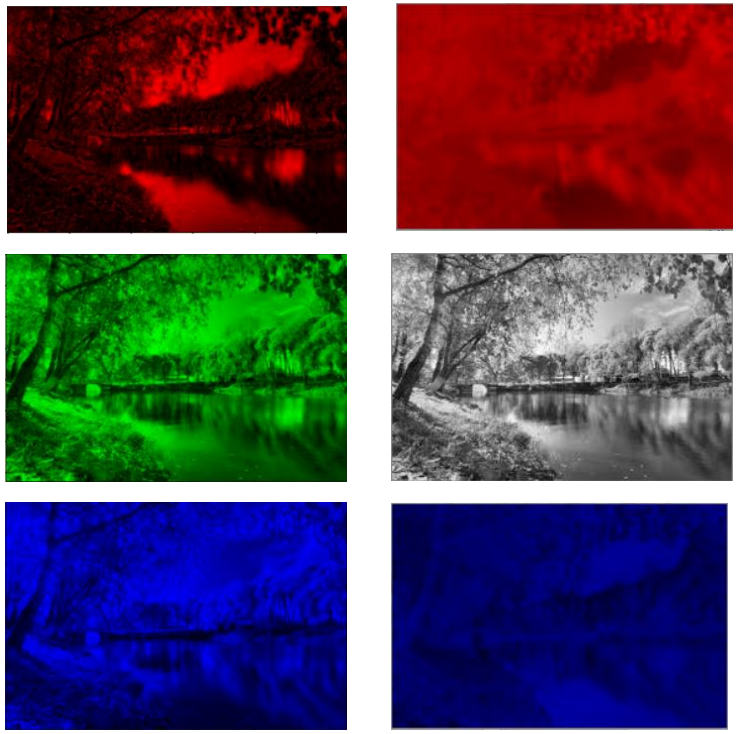
```
          0         1         2
0  1.000000  0.853406  0.673066
1  0.853406  1.000000  0.663643
2  0.673066  0.663643  1.000000
          0         1         2
0  1.000000 -0.260237  0.420621
1 -0.260237  1.000000 -0.265349
2  0.420621 -0.265349  1.000000
```



| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 1 | 0.853406 | 0.673066 | 0.663643 | -0.26024 | 0.420621 | -0.26535 |

# 3. 출력 결과

- Picure2의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient 값을 의미합니다.

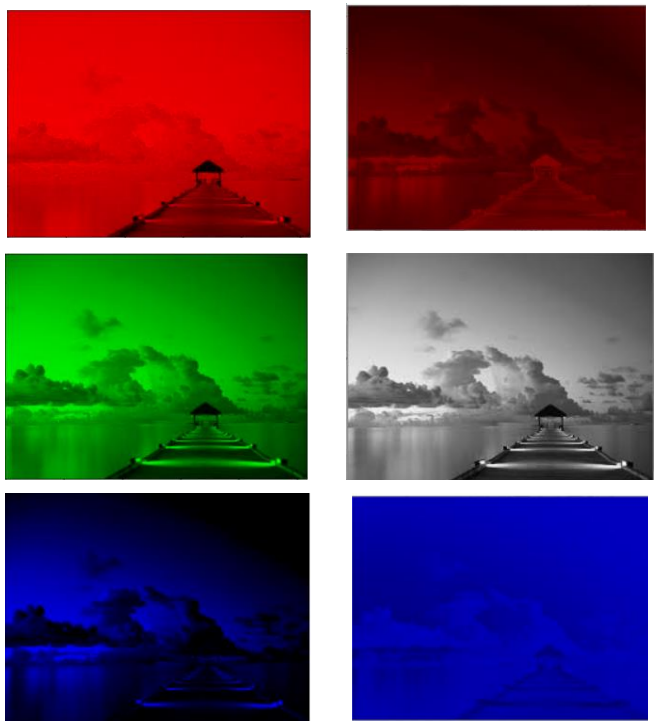


```
          0         1         2
0  1.000000  0.816919  0.398149
1  0.816919  1.000000  0.728353
2  0.398149  0.728353  1.000000
          0         1         2
0  1.000000 -0.133444 -0.094181
1 -0.133444  1.000000 -0.725880
2 -0.094181 -0.725880  1.000000
```

| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 2 | 0.816919 | 0.398149 | 0.728353 | -0.13344 | -0.09418 | -0.72588 |

# 3. 출력 결과

- Picure3의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient 값을 의미합니다.

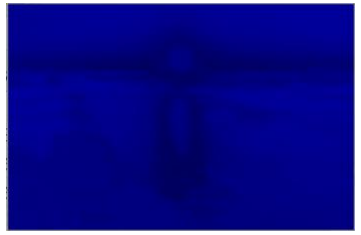


```
          0         1         2
0  1.000000  0.846525  0.451724
1  0.846525  1.000000  0.799890
2  0.451724  0.799890  1.000000
          0         1         2
0  1.000000 -0.530526  0.230285
1 -0.530526  1.000000 -0.831629
2  0.230285 -0.831629  1.000000
```

| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 3 | 0.846525 | 0.451724 | 0.79989 | -0.53053 | 0.230285 | -0.83163 |

# 3. 출력 결과

- Picure4의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient 값을 의미합니다.



```
---------------------- RESTART: C:/
          0         1         2
0  1.000000  0.956503  0.685689
1  0.956503  1.000000  0.860010
2  0.685689  0.860010  1.000000
          0         1         2
0  1.000000 -0.354666  0.297383
1 -0.354666  1.000000 -0.968120
2  0.297383 -0.968120  1.000000
```

| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 4 | 0.956503 | 0.685689 | 0.86001 | -0.35467 | 0.297383 | -0.96812 |

# 3. 출력 결과

- Picure5의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient 값을 의미합니다.

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1.000000 | 0.900710 | -0.243707 |
| 1 | 0.900710 | 1.000000 | -0.078518 |
| 2 | -0.243707 | -0.078518 | 1.000000 |

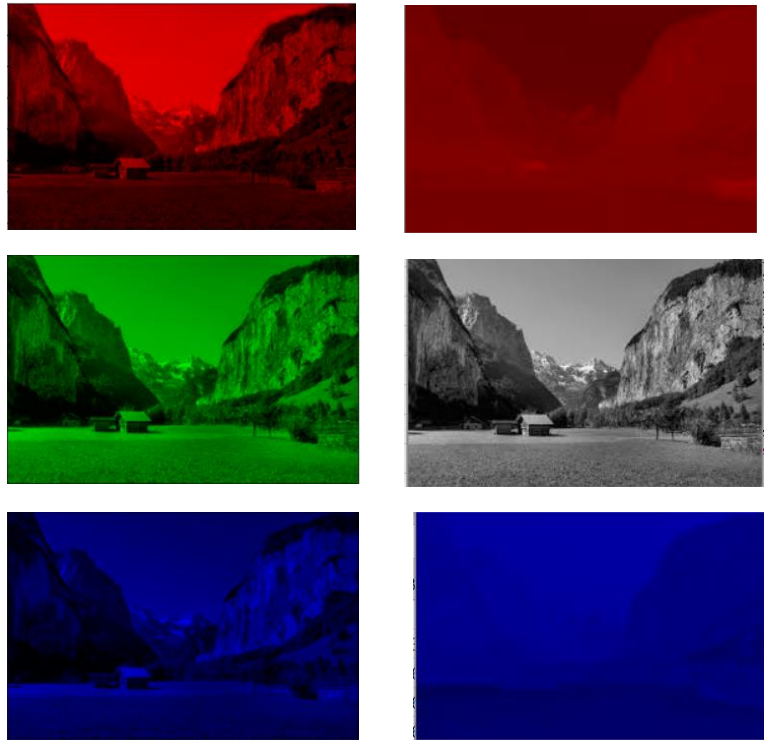
| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1.000000 | -0.537373 | 0.691477 |
| 1 | -0.537373 | 1.000000 | -0.833667 |
| 2 | 0.691477 | -0.833667 | 1.000000 |

| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 5 | 0.90071 | -0.24371 | -0.07852 | -0.53737 | 0.691477 | -0.83367 |

# 3. 출력 결과

- Picure6의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient 값을 의미합니다.

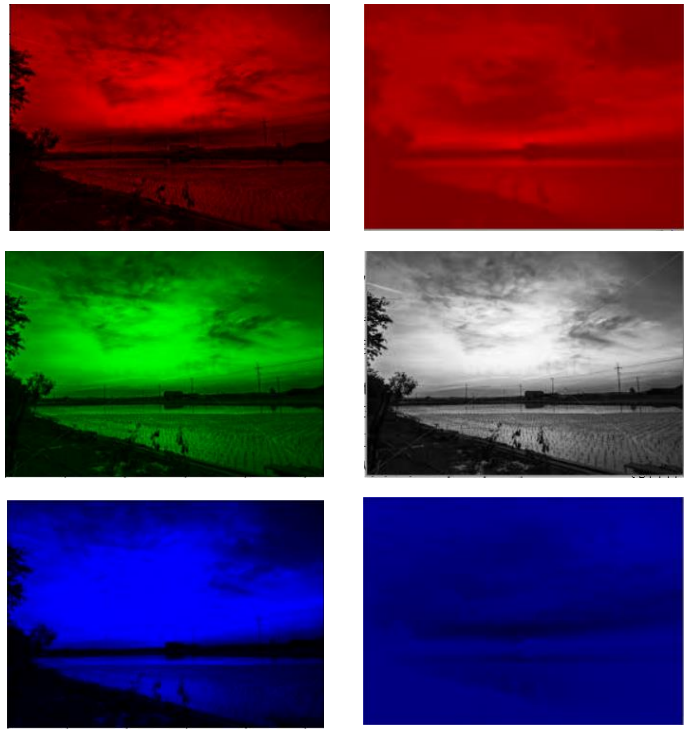
```
          0         1         2
0  1.000000  0.946907  0.666531
1  0.946907  1.000000  0.748904
2  0.666531  0.748904  1.000000
          0         1         2
0  1.000000  0.221438 -0.495001
1  0.221438  1.000000 -0.654555
2 -0.495001 -0.654555  1.000000
```



| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 6 | 0.946907 | 0.666531 | 0.748904 | 0.221438 | -0.495 | -0.65456 |

# 3. 출력 결과

- Picure7의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient 값을 의미합니다.

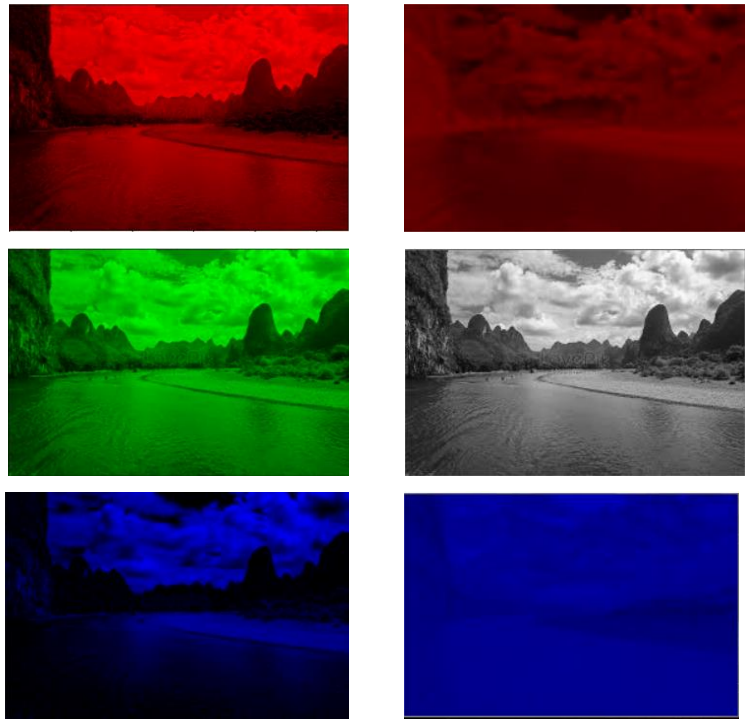


```
          0         1         2
0  1.000000  0.948069  0.792500
1  0.948069  1.000000  0.853629
2  0.792500  0.853629  1.000000
          0         1         2
0  1.000000 -0.405513  0.615782
1 -0.405513  1.000000 -0.629279
2  0.615782 -0.629279  1.000000
```

| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 7 | 0.948069 | 0.7925 | 0.853629 | -0.40551 | 0.615782 | -0.62928 |

# 3. 출력 결과

- Picure8의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient 값을 의미합니다.



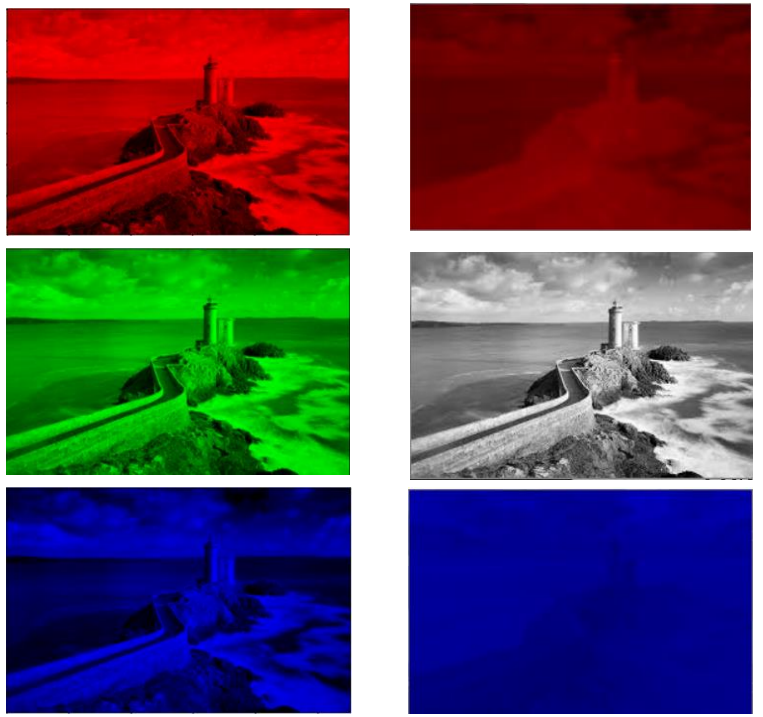
| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1.000000 | 0.917567 | 0.820304 |
| 1 | 0.917567 | 1.000000 | 0.953164 |
| 2 | 0.820304 | 0.953164 | 1.000000 |

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1.000000 | 0.245027 | 0.190414 |
| 1 | 0.245027 | 1.000000 | -0.710868 |
| 2 | 0.190414 | -0.710868 | 1.000000 |

| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 8 | 0.917567 | 0.820304 | 0.953164 | 0.245027 | 0.190414 | -0.71087 |

# 3. 출력 결과

- Picure9의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient 값을 의미합니다.



```
==================== RESTART: C:
          0         1         2
0  1.000000  0.890983  0.637145
1  0.890983  1.000000  0.902320
2  0.637145  0.902320  1.000000
          0         1         2
0  1.000000 -0.015535  0.093790
1 -0.015535  1.000000 -0.941269
2  0.093790 -0.941269  1.000000
```

| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 9 | 0.890983 | 0.637145 | 0.90232 | -0.01554 | 0.09379 | -0.94127 |

# 3. 출력 결과

- Picure10의 사진입니다.
- 왼쪽은 순서대로 R, G, B이고, 오른쪽은 순서대로 V, Y, U 이미지입니다.
- correlation coefficient 값은 다음과 같이 나왔는데, 이를 엑셀에서 정리하면 다음과 같은 결과를 확인할 수 있습니다.
  왼쪽에서부터 RG, RB, GB, YU, YV, UV에 대한 correlation coefficient 값을 의미합니다.

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1.000000 | 0.967293 | 0.792538 |
| 1 | 0.967293 | 1.000000 | 0.897029 |
| 2 | 0.792538 | 0.897029 | 1.000000 |

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | 1.000000 | -0.626087 | 0.458531 |
| 1 | -0.626087 | 1.000000 | -0.869708 |
| 2 | 0.458531 | -0.869708 | 1.000000 |

| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 10 | 0.967293 | 0.792538 | 0.897029 | -0.62609 | 0.458531 | -0.86971 |

# 4. 결과 분석

- 실행 결과 다음과 같은 결과가 나옴을 확인해 볼 수 있었습니다.
- 대체적으로 RGB간의 correlation coefficient 값이 YUV 간의 correlation coefficient 값보다 큰 것을 확인해 볼 수 있었습니다.
- YUV를 이용해 출력한 이미지는 RGB를 이용해 출력한 이미지보다 흐릿하고 correlation coefficient 값도 상대적으로 작으므로 영상의 크기를 줄여서 DCT하여 압축하면 더 작은 픽셀을 가지고 복원한 이미지를 만들어도 큰 차이가 없을 것입니다.
- 즉, 압축률이 높아지므로 저장해야 할 픽셀의 양이 적어진다는 결과를 확인해 볼 수 있었습니다.

| | R G | R B | G B | Y U | Y V | U V |
|---|---|---|---|---|---|---|
| 1 | 0.853406 | 0.673066 | 0.663643 | -0.26024 | 0.420621 | -0.26535 |
| 2 | 0.816919 | 0.398149 | 0.728353 | -0.13344 | -0.09418 | -0.72588 |
| 3 | 0.846525 | 0.451724 | 0.79989 | -0.53053 | 0.230285 | -0.83163 |
| 4 | 0.956503 | 0.685689 | 0.86001 | -0.35467 | 0.297383 | -0.96812 |
| 5 | 0.90071 | -0.24371 | -0.07852 | -0.53737 | 0.691477 | -0.83367 |
| 6 | 0.946907 | 0.666531 | 0.748904 | 0.221438 | -0.495 | -0.65456 |
| 7 | 0.948069 | 0.7925 | 0.853629 | -0.40551 | 0.615782 | -0.62928 |
| 8 | 0.917567 | 0.820304 | 0.953164 | 0.245027 | 0.190414 | -0.71087 |
| 9 | 0.890983 | 0.637145 | 0.90232 | -0.01554 | 0.09379 | -0.94127 |
| 10 | 0.967293 | 0.792538 | 0.897029 | -0.62609 | 0.458531 | -0.86971 |