



수치해석 HW#10

2D DFT를 이용한 Pattern Recognition

Python(opencv) 사용

컴퓨터소프트웨어학부 2015005187 최철훈

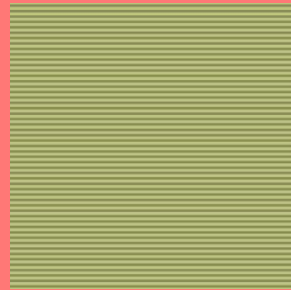
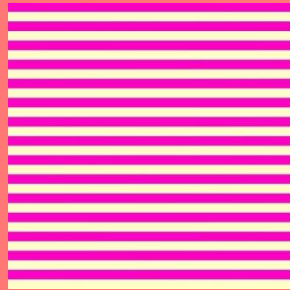
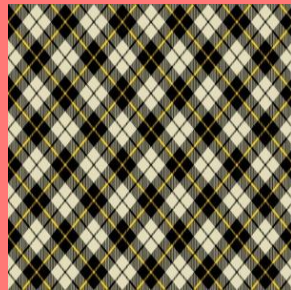
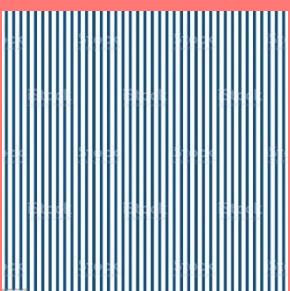
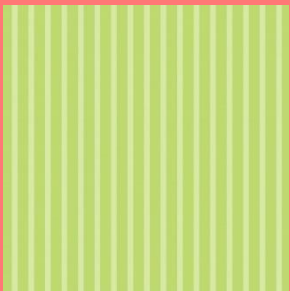
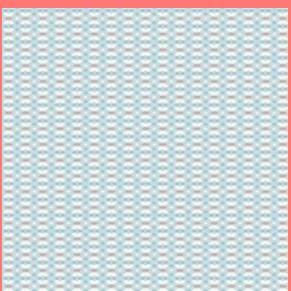
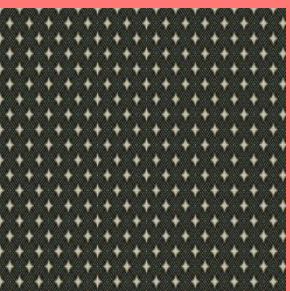
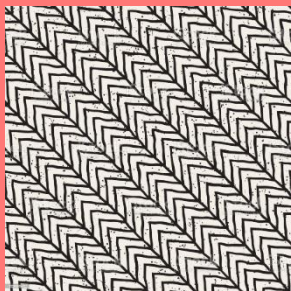
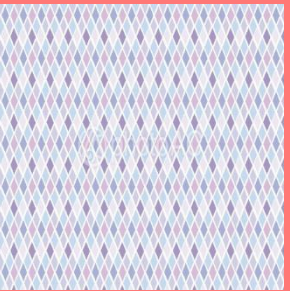
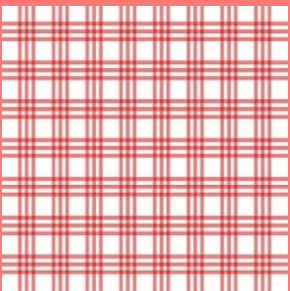
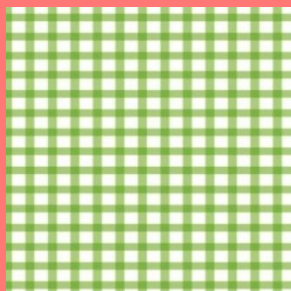
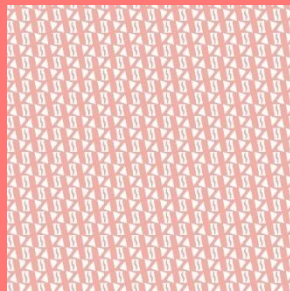


목차

1. 패턴 이미지 Scaling 및 64X64 샘플 구하기
2. DFT계수 구하기
3. Dominant coefficient 선택하기
4. Pattern Recognition
5. 마무리

이미지 Scailing 및 샘플 구하기 scailing

구글링하여 20개의 다양한 패턴 이미지를 모았다.



이미지 Scailing 및 샘플 구하기 sampling

첫 번째 패턴 이미지의 가운데 부분을 64X64 크기로 잘랐더니 1개 이상의 패턴이 블록 안에 다 들어왔다.

첫 번째 패턴이 성공적으로 64X64 샘플을 뽑아냈으므로 첫 번째 패턴 이미지를 기준으로 다른 패턴 이미지들도 사이즈를 비슷하게 맞췄다.

1개 패턴의 크기가 첫 번째 패턴의 크기보다 작거나 비슷하면 64X64 샘플을 뽑았을 때 1개 이상의 패턴이 알맞게 들어갔다.

그러나 첫 번째 패턴보다 크기가 큰 패턴의 경우는 한 블록에 들어가지 않아서 크기를 조금씩 줄여나가면서 64X64 샘플에 1개 이상의 패턴이 들어가도록 조절하였다.

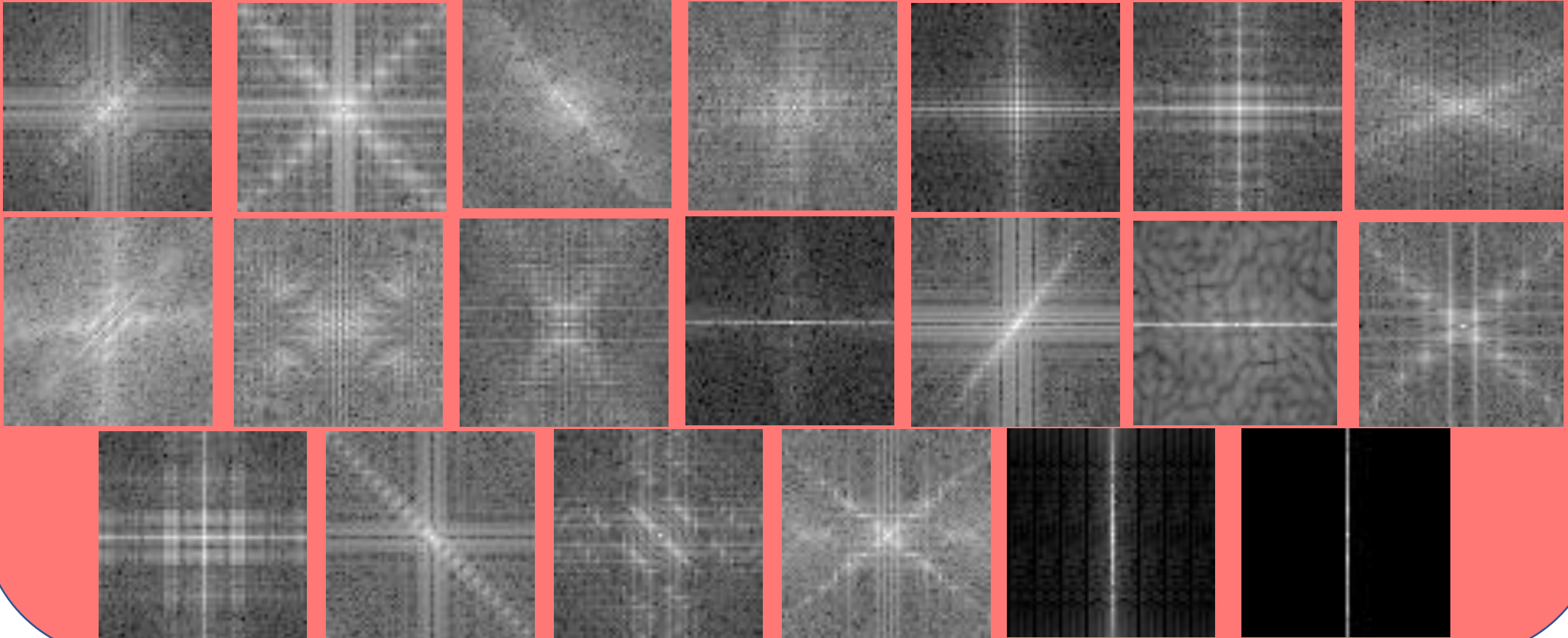
이미지 Scailing 및 샘플 구하기 sampling



DFT계수 구하기 DFT 과정 & 결과

앞에서 뽑은 각각의 64X64 샘플 패턴을 흑백으로 불러오고
DFT계수를 구한다. DFT계수를 구하기 위해 FFT알고리즘을 사용하였다.

```
dft = np.fft.fft2(img)
dft = np.fft.fftshift(dft)
magnitude = 20 * np.log(np.abs(dft))
```



Dominant Coefficient 선택하기 선택과정

각 패턴들의 DFT계수들 중에서 비교에 사용할 값이 큰 coefficient들을 선택한다.

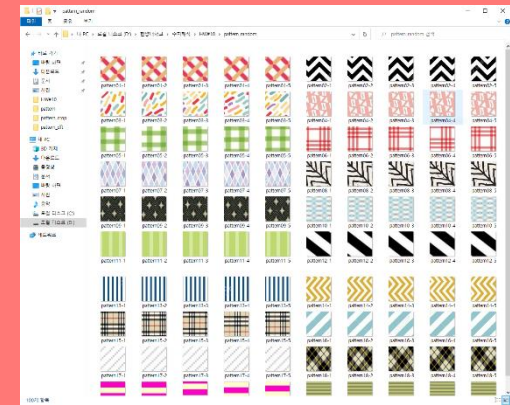
1. 20개의 패턴 각각의 DFT계수들을 출력하여 관찰해본다.
2. 계수들이 원점 대칭인 경우도 있었지만 대부분이 x축 대칭이었다. 그러므로 위에 부분만 남기고 아래는 버린다. DFT 이미지를 만들었을 때와 같이 계수의 크기에 로그를 취하고 20을 곱한 값을 관찰하기 편하게 하기 위해 정수로 변환하여 비교하였다.
3. 위에 부분만을 비교하니 low frequency부근에서 변화가 많고 값이 크며 high frequency는 대부분 0이거나 변화가 적었다. 그러므로 low frequency부분만을 비교 대상으로 삼기로 결정하여 DC성분을 중심으로 왼쪽, 오른쪽, 10줄과 위로 10줄 210개, 윗 부분의 약 10%의 계수만을 선택하였다.

```
part = magnitude[23:33, 22:43]
```

Pattern Recognition 단순 vector distance 과정

앞에서 구한 DFT계수들과 각각의 패턴들에서 랜덤하게 뽑은 64X64 샘플의 DFT계수들의 차이를 단순 비교하여 차이가 가장 작은 패턴을 일치한다고 판단한다.

1. 20개의 패턴들 각각 랜덤하게 64X64 샘플을 5개씩 뽑는다.
2. 랜덤하게 뽑은 패턴들의 DFT계수를 구한다. DC 성분은 제거한다.
3. 랜덤한 샘플 패턴 1개마다 앞서 구한 20개의 DFT계수들의 차이를 구한다. 차이의 절대값의 합이 가장 작은 패턴을 일치한다고 판단한다.



```
compare_sum = np.zeros(20, dtype=np.float32)
for i in range(0, 20):
    compare = part - dft_collect[i]
    compare_sum[i] = np.sum(abs(compare))
match = np.argmin(compare_sum)
print("This is " + str(match + 1) + " Pattern!!")
```


Pattern Recognition 단순 vector distance 결과

100개중 7개를 제외하고 모두 일치하는 결과가 나왔다. Pattern19와 20은 같은 가로 줄무늬 패턴이기 때문에 구별하지 못한 것 같다. Pattern11은 세로 줄무늬 패턴이어서 19와는 coefficient의 분포가 전혀 반대이지만 차이의 합으로 판단하기에 비슷하다고 분류된 것 같다.

Pattern 11

Pattern 19



Pattern20

```
This is 1 Pattern!!
This is 1 Pattern!!
This is 1 Pattern!!
This is 1 Pattern!!
This is 1 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 3 Pattern!!
This is 3 Pattern!!
This is 3 Pattern!!
This is 3 Pattern!!
This is 3 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
```

This is 6 Pattern!!
This is 6 Pattern!!
This is 6 Pattern!!
This is 6 Pattern!!
This is 6 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 10 Pattern!!
This is 10 Pattern!!
This is 10 Pattern!!
This is 10 Pattern!!
This is 10 Pattern!!

```
This is 11 Pattern!!
This is 11 Pattern!!
This is 11 Pattern!!
This is 11 Pattern!!
This is 11 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
```

This is 16 Pattern!!
This is 16 Pattern!!
This is 16 Pattern!!
This is 16 Pattern!!
This is 16 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 11 Pattern!!
This is 11 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!

Pattern Recognition partially vector distance 과정

앞에서 구한 DFT계수들 중에서 값이 높은 의미있는 coefficient만을 선택한다. 각각의 패턴들에서 랜덤하게 뽑은 64X64 샘플의 DFT계수들도 의미있는 coefficient를 뽑아서 차이를 비교하여 차이가 가장 작은 패턴을 일치한다고 판단한다.

1. 앞서 구한 DFT계수들 중에서 150초과의 값만을 남기고 DC 성분을 포함한 나머지는 0으로 만든다.
2. 20개의 패턴들 각각 랜덤하게 64X64 샘플을 5개씩 뽑는다.
3. 랜덤하게 뽑은 패턴들의 DFT계수를 구한다. 의미있는 coefficient만 비교하기 위해서 마찬가지로 150초과인 값들만 남기고 DC성분을 포함한 나머지 값들은 0으로 만들었다.
4. 랜덤한 샘플 패턴 1개마다 앞서 구한 20개의 DFT계수들의 차이를 구한다. 차이의 절대값의 합이 가장 작은 패턴을 일치한다고 판단한다.

```
part = magnitude[23:33, 22:43]
part[9, 10] = 0
for i in range(0, 10):
    for j in range(0, 21):
        if part[i, j] > 150:
            dft_collect[numbers - 1, i, j] = part[i, j]
```

Pattern Recognition partially vector distance 결과

100개중 2개를 제외하고 모두 일치하는 결과가 나왔다. Pattern3가 아무래도 좀 불규칙한 패턴이다보니 랜덤하게 뽑는 위치에 따라서 계수가 다르게 나오기 때문에 100%의 일치율이 안나오는 것 같다. 이렇게 일치율이 높은 이유는 의미있는 coefficient 끼리만 비교했기 때문으로 생각된다.

Pattern 3



```

This is 1 Pattern!!
This is 1 Pattern!!
This is 1 Pattern!!
This is 1 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 3 Pattern!!
This is 2 Pattern!!
This is 3 Pattern!!
This is 3 Pattern!!
This is 2 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!

```

```
This is 6 Pattern!!
This is 6 Pattern!!
This is 6 Pattern!!
This is 6 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 10 Pattern!
This is 10 Pattern!
This is 10 Pattern!
This is 10 Pattern!
```

```
This is 11 Pattern!!
This is 11 Pattern!!
This is 11 Pattern!!
This is 11 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
```

This is 16 Pattern!!
This is 16 Pattern!!
This is 16 Pattern!!
This is 16 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 20 Pattern!!
This is 20 Pattern!!
This is 20 Pattern!!
This is 20 Pattern!!

Pattern Recognition average distance 과정

각각의 패턴에서 랜덤하게 64X64 샘플 5개를 뽑아서 DFT계수를 구하고 평균을 구한다. 평균에서 의미있는 coefficient만을 선택한다. 각각의 패턴들에서 랜덤하게 뽑은 64X64 샘플의 DFT계수들도 의미있는 coefficient를 뽑아서 차이를 비교하여 차이가 가장 작은 패턴을 일치한다고 판단한다.

1. 5개의 랜덤 샘플의 평균을 구한뒤 150초과의 값만을 남기고 DC 성분을 포함한 나머지는 0으로 만든다.
2. 20개의 패턴들 각각 랜덤하게 64X64 샘플을 5개씩 뽑는다.
3. 랜덤하게 뽑은 패턴들의 DFT계수를 구한다. 의미있는 coefficient만 비교하기 위해서 마찬가지로 150초과인 값들만 남기고 DC성분을 포함한 나머지 값들은 0으로 만들었다.
4. 랜덤한 샘플 패턴 1개마다 앞서 구한 20개의 DFT계수들의 평균과의 차이를 구한다. 차이의 절대값의 합이 가장 작은 패턴을 일치한다고 판단한다.

```
mean = mean/5
mean[9, 10] = 0
for i in range(0, 10):
    for j in range(0, 21):
        if mean[i, j] > 150:
            average_dft[n, i, j] = mean[i, j]
n = n + 1
```

Pattern Recognition average distance 결과

100개 모두 일치하는 결과가 나타났다. 앞서 잘 인식하지 못했던 Pattern3도 불규칙한 패턴이지만 평균을 구함으로써 제대로 인식하게 된 것 같다.

```
This is 1 Pattern!!
This is 1 Pattern!!
This is 1 Pattern!!
This is 1 Pattern!!
This is 1 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 2 Pattern!!
This is 3 Pattern!!
This is 3 Pattern!!
This is 3 Pattern!!
This is 3 Pattern!!
This is 3 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 4 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
This is 5 Pattern!!
```

```
This is 6 Pattern!!
This is 6 Pattern!!
This is 6 Pattern!!
This is 6 Pattern!!
This is 6 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 7 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 8 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 9 Pattern!!
This is 10 Pattern!!
This is 10 Pattern!!
This is 10 Pattern!!
This is 10 Pattern!!
This is 10 Pattern!!
```

```
This is 11 Pattern!!
This is 11 Pattern!!
This is 11 Pattern!!
This is 11 Pattern!!
This is 11 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 12 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 13 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 14 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
This is 15 Pattern!!
```

```
This is 16 Pattern!!
This is 16 Pattern!!
This is 16 Pattern!!
This is 16 Pattern!!
This is 16 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 17 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 18 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 19 Pattern!!
This is 20 Pattern!!
This is 20 Pattern!!
This is 20 Pattern!!
This is 20 Pattern!!
This is 20 Pattern!!
```


Pattern Recognition range 과정

각각의 패턴에서 랜덤하게 64X64 샘플 5개를 뽑아서 DFT계수를 구하고 평균을 구한다. 평균에서 의미있는 coefficient만을 선택한다. 평균에서 샘플 5개의 DFT계수들의 차이중 가장 큰 값을 range로 정한다. 각각의 패턴들에서 랜덤하게 뽑은 64X64 샘플의 DFT계수들도 의미있는 coefficient를 뽑아서 range보다 작으면 해당 패턴과 일치한다고 판단한다. 만일 일치한다고 판단되는 패턴이 2개 이상일 시에는 앞서 평균과의 차이가 가장 작은 것으로 판단했을 때 높은 일치율을 보였으므로, 이 패턴들 중 평균과의 차이가 가장 작은 것으로 판단한다.

1. 5개의 랜덤 샘플의 평균을 구한뒤 평균과 각각의 계수들의 차이 중 가장 큰 값을 기준으로 정한다.

```
for i in range(0, 5):  
    difference[i] = np.sum(abs(dft_collect[i] - average_dft[n]))  
minmax[n] = np.max(difference)
```

2. 20개의 패턴들 각각 랜덤하게 64X64 샘플을 5개씩 뽑는다.

3. 랜덤하게 뽑은 패턴들의 DFT계수를 구한다. 앞서 구한 평균과의 차이를 구해서 기준보다 작은 패턴이 존재하는지 판단한다. 없다면 일치하는 패턴이 없다고 판단하고, 2개 이상이라면 이 중 평균과의 차이가 가장 작은 패턴이 일치하는 패턴이라고 판단한다.

Pattern Recognition range 결과

앞선 비교들 보다 일치율이 많이 낮아졌다. 일치하는 패턴이 없다고 나온 결과가 30건 이고, 일치하는 패턴이 2개 이상이라고 판단한 결과는 8건이었다. 일치하는 패턴이 없다고 나온 경우는 복잡하고 무늬가 단순하지 않은 패턴들이 대부분이었다. 일치하는 패턴이 2개 이상이라고 판단한 경우는 Patter11,13과 Pattern19,20과 같은 세로무늬, 가로무늬인데 조밀한 정도만 다른 경우였다.

```
Not exist matching Pattern0
This is 1 Pattern!! 1
This is 1 Pattern!! 2
This is 1 Pattern!! 3
Not exist matching Pattern4
This is 2 Pattern!! 0
This is 2 Pattern!! 1
This is 2 Pattern!! 2
This is 2 Pattern!! 3
This is 2 Pattern!! 4
Not exist matching Pattern0
Not exist matching Pattern1
Not exist matching Pattern2
Not exist matching Pattern3
Not exist matching Pattern4
This is 4 Pattern!! 0
This is 4 Pattern!! 1
This is 4 Pattern!! 2
This is 4 Pattern!! 3
This is 4 Pattern!! 4
This is 5 Pattern!! 0
This is 5 Pattern!! 1
This is 5 Pattern!! 2
Not exist matching Pattern3
Not exist matching Pattern4
```

```
This is 6 Pattern!! 0
This is 6 Pattern!! 1
This is 6 Pattern!! 2
This is 6 Pattern!! 3
This is 6 Pattern!! 4
This is 7 Pattern!! 0
Not exist matching Pattern1
This is 7 Pattern!! 2
Not exist matching Pattern3
Not exist matching Pattern4
This is 8 Pattern!! 0
Not exist matching Pattern1
Not exist matching Pattern2
Not exist matching Pattern3
This is 8 Pattern!! 4
This is 9 Pattern!! 0
Not exist matching Pattern1
Not exist matching Pattern2
Not exist matching Pattern3
This is 9 Pattern!! 4
This is 10 Pattern!! 0
This is 10 Pattern!! 1
Not exist matching Pattern2
Not exist matching Pattern3
This is 10 Pattern!! 4
```

```
This is 11 Pattern!! 0
This is 13 Pattern!! 0
Finally, This is 11Pattern!!
This is 11 Pattern!! 1
This is 13 Pattern!! 1
Finally, This is 11Pattern!!
This is 11 Pattern!! 2
This is 13 Pattern!! 2
Finally, This is 11Pattern!!
This is 13 Pattern!! 3
This is 13 Pattern!! 4
This is 12 Pattern!! 0
This is 12 Pattern!! 1
This is 12 Pattern!! 2
This is 12 Pattern!! 3
This is 12 Pattern!! 4
This is 13 Pattern!! 0
This is 13 Pattern!! 1
Not exist matching Pattern2
This is 13 Pattern!! 3
This is 13 Pattern!! 4
Not exist matching Pattern0
Not exist matching Pattern1
This is 14 Pattern!! 2
This is 14 Pattern!! 3
Not exist matching Pattern4
This is 15 Pattern!! 0
This is 15 Pattern!! 1
This is 15 Pattern!! 2
Not exist matching Pattern3
This is 15 Pattern!! 4
```

```
This is 16 Pattern!! 0
This is 16 Pattern!! 1
Not exist matching Pattern2
Not exist matching Pattern3
This is 16 Pattern!! 4
This is 17 Pattern!! 0
This is 17 Pattern!! 1
Not exist matching Pattern2
This is 17 Pattern!! 3
This is 17 Pattern!! 4
Not exist matching Pattern0
Not exist matching Pattern1
This is 18 Pattern!! 2
This is 18 Pattern!! 3
This is 18 Pattern!! 4
This is 19 Pattern!! 0
This is 20 Pattern!! 0
Finally, This is 19Pattern!!
This is 19 Pattern!! 1
This is 20 Pattern!! 1
Finally, This is 19Pattern!!
This is 19 Pattern!! 2
This is 20 Pattern!! 2
Finally, This is 19Pattern!!
This is 19 Pattern!! 3
This is 20 Pattern!! 3
Finally, This is 19Pattern!!
This is 19 Pattern!! 4
This is 20 Pattern!! 4
Finally, This is 19Pattern!!
This is 20 Pattern!! 0
This is 20 Pattern!! 1
This is 20 Pattern!! 2
This is 20 Pattern!! 3
This is 20 Pattern!! 4
```

마무리 느낀점

1. 이번 과제에서는 64X64 크기의 샘플을 얼마나 잘 뽑는지가 과제의 난이도를 결정하는 중요한 요소였던 것 같다. 이 샘플을 잘 뽑아냈기 때문에 결과가 좋게 나온 것이라 생각한다.
2. 더 많은 의미있는 coefficient를 선택하면 일치율이 올라갈 것이라 예상된다.
3. range비교에서 더 많은 샘플로 비교하고 기준을 잡았으면 일치율이 올라갈 것이라 예상된다.
4. range비교에서 단순한 패턴들만 있었으면 일치율이 올라갔을 것이라 예상된다.
5. range비교에서 범위가 겹쳤을 때에 판별하는 방법이 딱히 떠오르지 않았다. 어떤 방법이 있을지 더 고민해봐야겠다.

마무리

감사합니다.

https://github.com/cheol-hoon/Numerical_Analysis