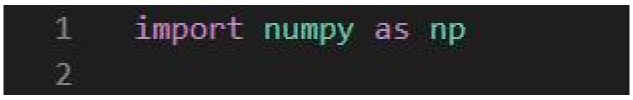
程设第七次作业

20377383 樊思涵

Task1

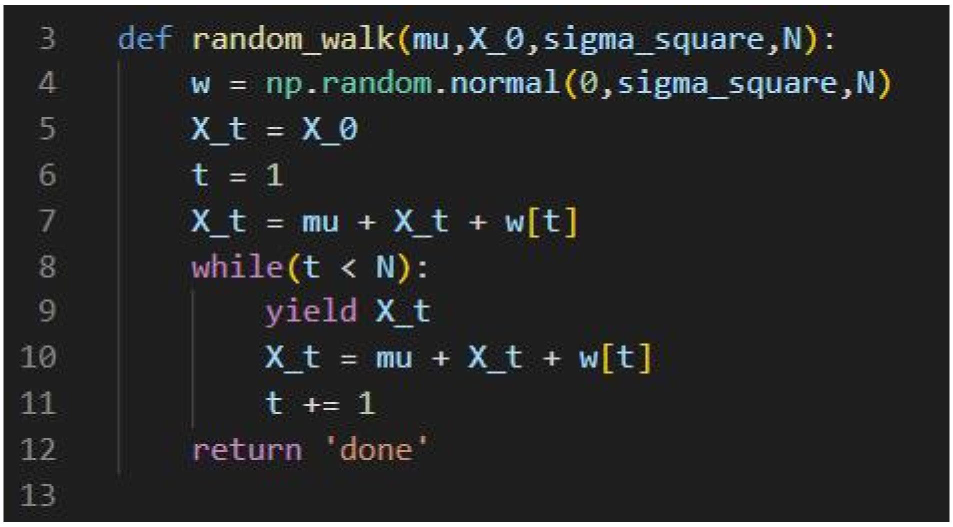
调用库



```python
import numpy as np

```

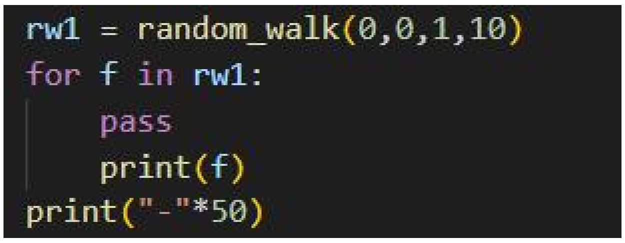
numpy：用于生成正态分布的随机变量

实现 random_walk 生成器

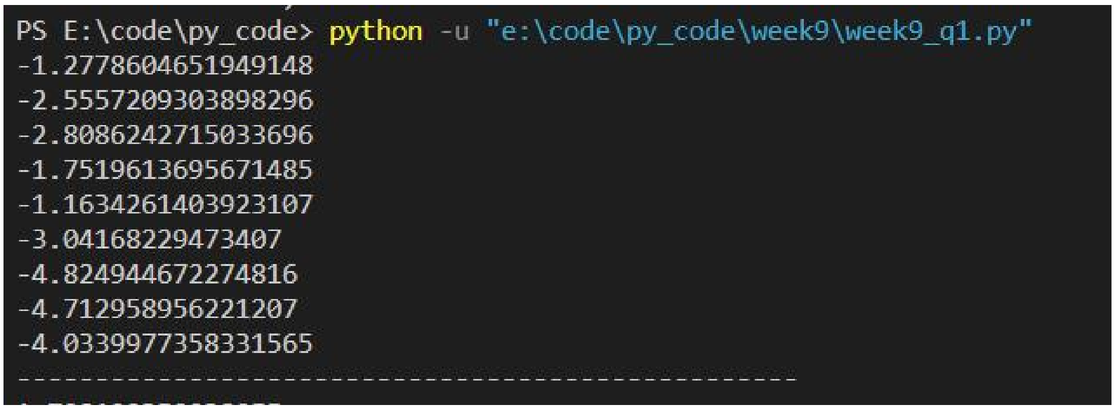


```python
def random_walk(mu,X_0,sigma_square,N):
    w = np.random.normal(0,sigma_square,N)
    X_t = X_0
    t = 1
    X_t = mu + X_t + w[t]
    while(t < N):
        yield X_t
        X_t = mu + X_t + w[t]
        t += 1
    return 'done'

```

检测输出结果

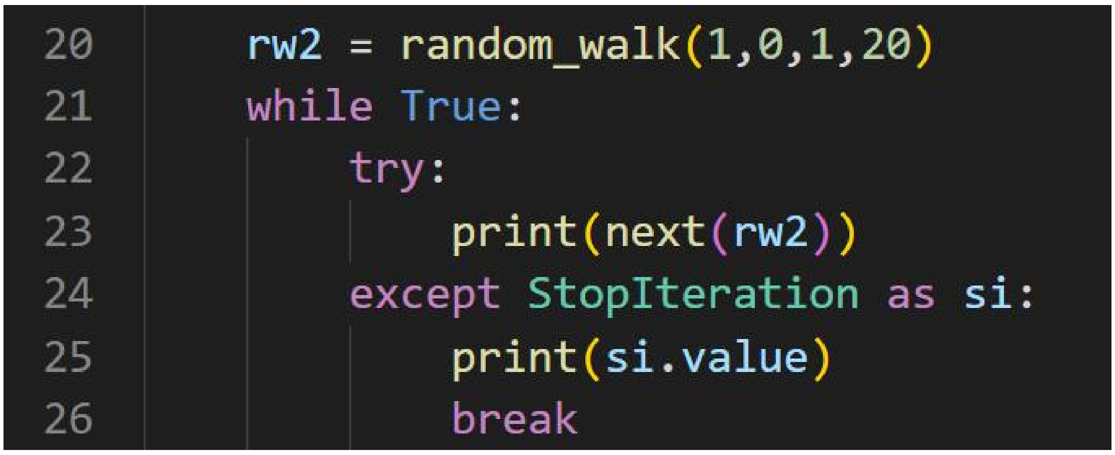


```python
rw1 = random_walk(0,0,1,10)
for f in rw1:
    pass
    print(f)
print("-"*50)
```

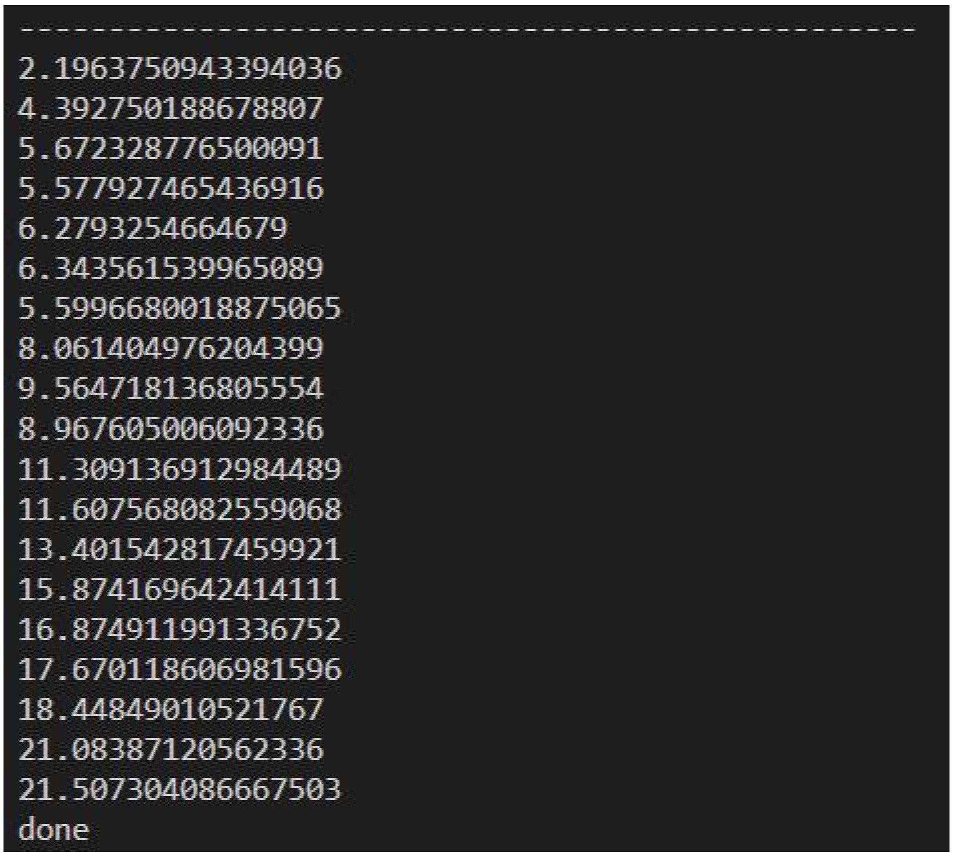


```
PS E:\code\py_code> python -u "e:\code\py_code\week9\week9_q1.py"
-1.2778604651949148
-2.5557209303898296
-2.8086242715033696
-1.7519613695671485
-1.1634261403923107
-3.04168229473407
-4.824944672274816
-4.712958956221207
-4.0339977358331565
--------------------------------------------------
```

尝试捕获生成器的错误信息

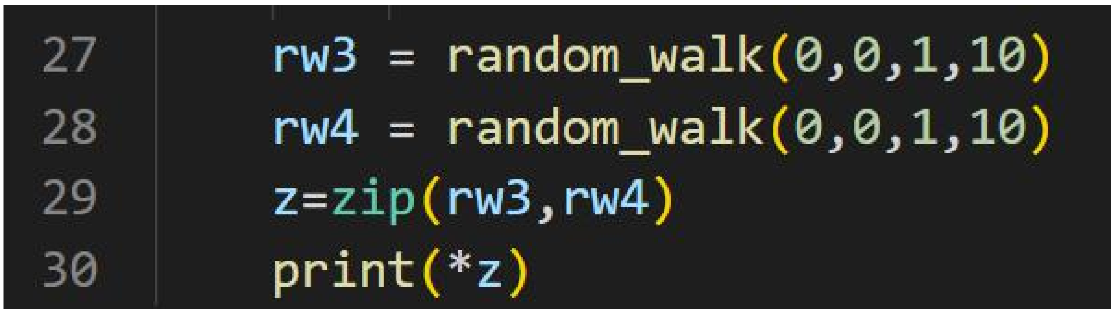


```python
    rw2 = random_walk(1,0,1,20)
    while True:
        try:
            print(next(rw2))
        except StopIteration as si:
            print(si.value)
            break
```

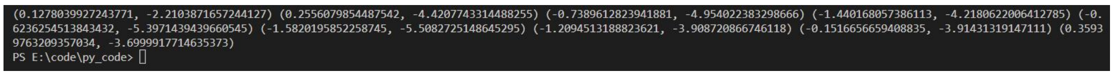


```
--------------------------------------------------
2.1963750943394036
4.392750188678807
5.672328776500091
5.577927465436916
6.2793254664679
6.343561539965089
5.5996680018875065
8.061404976204399
9.564718136805554
8.967605006092336
11.309136912984489
11.607568082559068
13.401542817459921
15.874169642414111
16.874911991336752
17.670118606981596
18.44849010521767
21.08387120562336
21.507304086667503
done
```

实现拼合多个 random_walk 的生成器



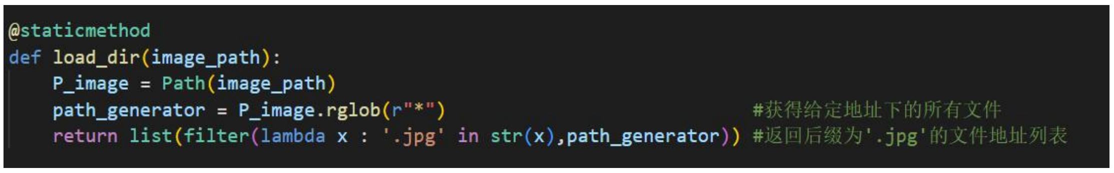
```python
    rw3 = random_walk(0,0,1,10)
    rw4 = random_walk(0,0,1,10)
    z=zip(rw3,rw4)
    print(*z)
```



```
(0.1278039927243771, -2.2103871657244127) (0.2556079854487542, -4.4207743314488255) (-0.7389612823941881, -4.954022383298666) (-1.440168057386113, -4.2180622006412785) (-0.
6236254513843432, -5.3971439439660545) (-1.5820195852258745, -5.5082725148645295) (-1.2094513188823621, -3.908720866746118) (-0.1516656659408835, -3.91431319147111) (0.3593
9763209357034, -3.6999917714635373)
PS E:\code\py_code> 
```

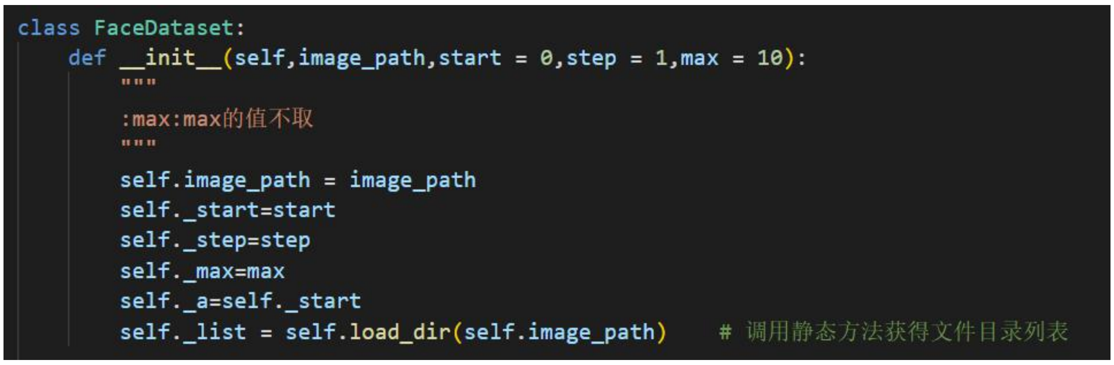
Task2

实现静态方法获得地址列表



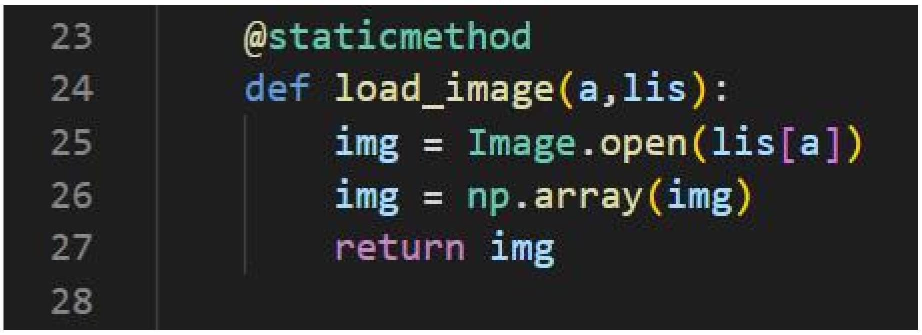
```python
@staticmethod
def load_dir(image_path):
    P_image = Path(image_path)
    path_generator = P_image.rglob(r"*")                               #获得给定地址下的所有文件
    return list(filter(lambda x : '.jpg' in str(x),path_generator)) #返回后缀为'.jpg'的文件地址列表
```

类的初始化



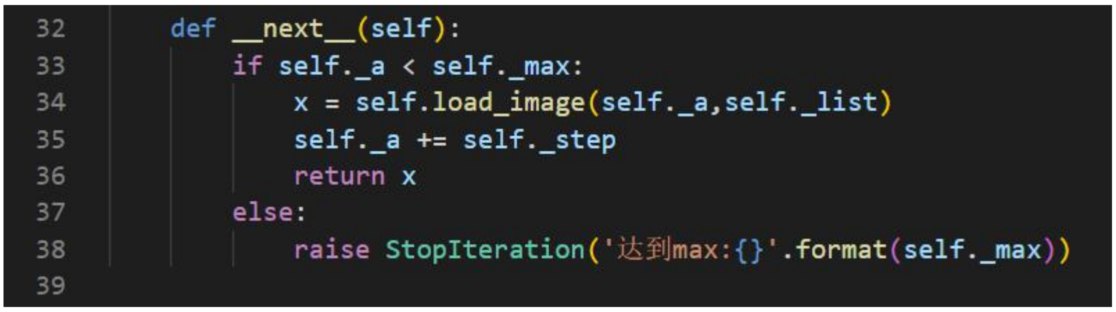
```python
class FaceDataset:
    def __init__(self,image_path,start = 0,step = 1,max = 10):
        """
        :max:max的值不取
        """
        self.image_path = image_path
        self._start=start
        self._step=step
        self._max=max
        self._a=self._start
        self._list = self.load_dir(self.image_path)    # 调用静态方法获得文件目录列表
```

实现静态方法将一张图片数据以 ndarray 的形式返回



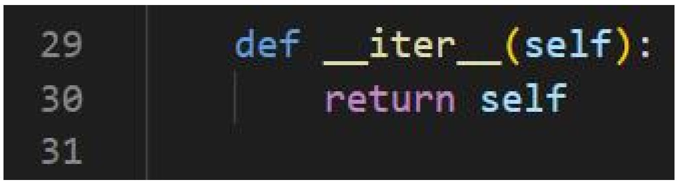
```python
    @staticmethod
    def load_image(a,lis):
        img = Image.open(lis[a])
        img = np.array(img)
        return img

```

实现__next__方法



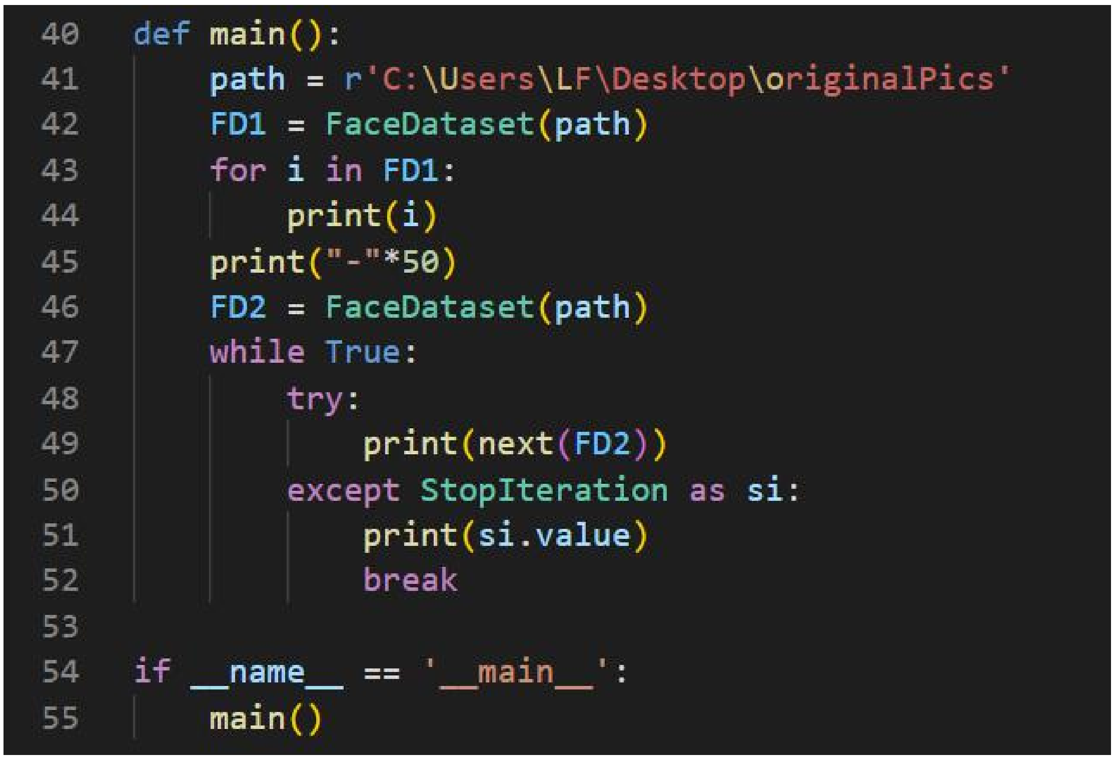
```python
    def __next__(self):
        if self._a < self._max:
            x = self.load_image(self._a,self._list)
            self._a += self._step
            return x
        else:
            raise StopIteration('达到max:{}'.format(self._max))

```

实现__iter__



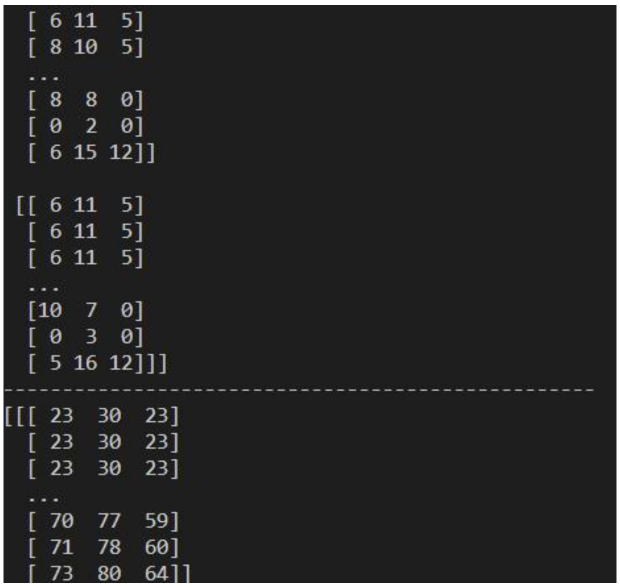
```python
    def __iter__(self):
        return self

```

在主函数中调用并实现

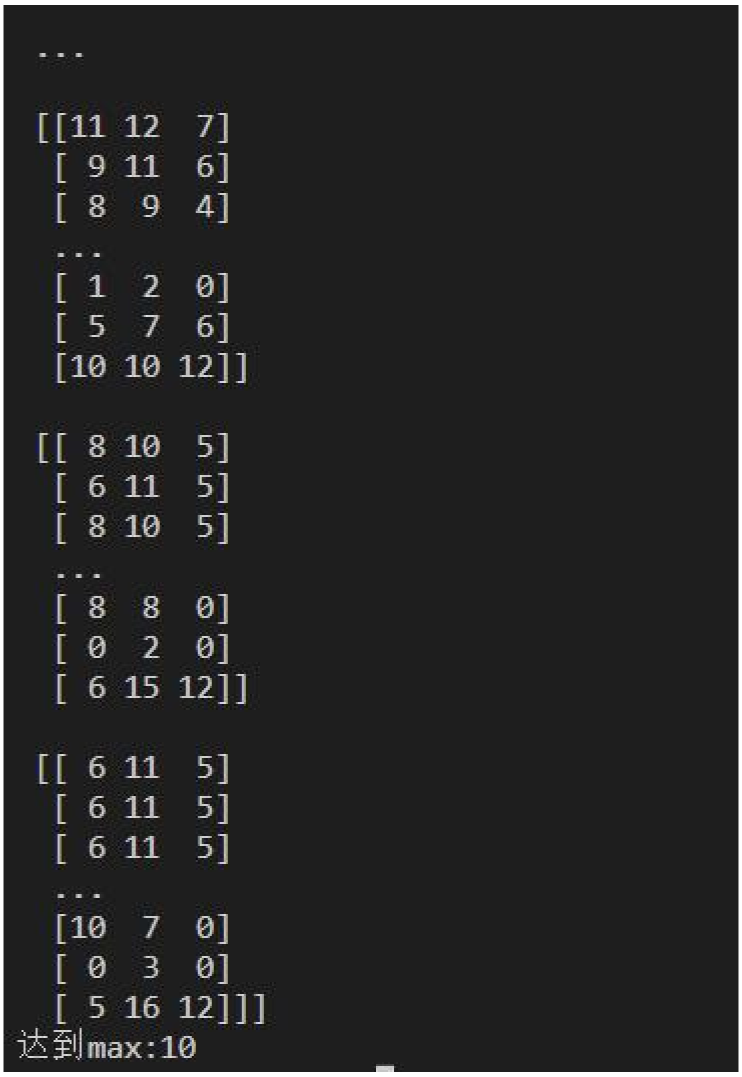


```python
def main():
    path = r'C:\Users\LF\Desktop\originalPics'
    FD1 = FaceDataset(path)
    for i in FD1:
        print(i)
    print("-"*50)
    FD2 = FaceDataset(path)
    while True:
        try:
            print(next(FD2))
        except StopIteration as si:
            print(si.value)
            break

if __name__ == '__main__':
    main()
```

结果展示



```
  [ 6 11  5]
  [ 8 10  5]
  ...
  [ 8  8  0]
  [ 0  2  0]
  [ 6 15 12]]

 [[ 6 11  5]
  [ 6 11  5]
  [ 6 11  5]
  ...
  [10  7  0]
  [ 0  3  0]
  [ 5 16 12]]]
--------------------------------------------------
[[[ 23  30  23]
  [ 23  30  23]
  [ 23  30  23]
  ...
  [ 70  77  59]
  [ 71  78  60]
  [ 73  80  64]]
```



```
 ...

 [[11 12  7]
  [ 9 11  6]
  [ 8  9  4]
  ...
  [ 1  2  0]
  [ 5  7  6]
  [10 10 12]]

 [[ 8 10  5]
  [ 6 11  5]
  [ 8 10  5]
  ...
  [ 8  8  0]
  [ 0  2  0]
  [ 6 15 12]]

 [[ 6 11  5]
  [ 6 11  5]
  [ 6 11  5]
  ...
  [10  7  0]
  [ 0  3  0]
  [ 5 16 12]]]
达到max:10
```

Ref.

使用 Pytorch 中的 Dataset 类构建数据集的方法及其底层逻辑
https://blog.csdn.net/rowevine/article/details/123631144
@staticmethod 和@classmethod 的用法
https://blog.csdn.net/polyhedronx/article/details/81911548
Python 使用 pathlib 库
http://www.qb5200.com/article/487180.html