Approach:

1. Start with a cell containing 1,
2. Perform bfs/dfs to find nearby land areas that together form an island, mark them to avoid recounting
3. Repeat until all cells convered

Code:

class Solution:

def numIslands(self, grid: List[List[str]]) -> int:

r = len(grid)

c = len(grid[0])

def bfs(u,v):

queue = deque()

queue.append((u,v))

visited = set()

while queue:

u,v = queue.popleft()

if (u,v) in visited:

continue

# add to visited

visited.add((u,v))

# change/mark field

grid[u][v] = "-1"

# find all neighbours with "1" entry

nei = [(u-1,v), (u,v-1), (u+1,v), (u,v+1)]

for n in nei:

x,y = n

if x>=0 and x<r and y>=0 and y<c and grid[x][y] == "1":

queue.append((x,y))

count =0

for i in range(r):

for j in range(c):

if grid[i][j] == "1":

bfs(i,j)

count += 1

return count