

```

import os
import cv2
import numpy as np
import pandas as pd
import heapq

BASE_INPUT = "/kaggle/input"
DATASET_NAME = os.listdir(BASE_INPUT)[0]
TEST_DIR = os.path.join(BASE_INPUT, DATASET_NAME, "test")

print("Using dataset:", DATASET_NAME)
print("Test dir:", TEST_DIR)
print("Total test images:", len(os.listdir(TEST_DIR)))

def image_to_grid(img):
    img = cv2.resize(img, (20, 20))
    gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

    grid = np.zeros((20, 20), dtype=int)

    for i in range(20):
        for j in range(20):
            v = gray[i, j]
            if v < 40:
                grid[i][j] = 1
            elif v < 120:
                grid[i][j] = 2
            else:
                grid[i][j] = 0

    grid[0][0] = 3
    grid[19][19] = 4
    return grid

def step_cost(cell):
    if cell == 0: return 1
    if cell == 2: return 3
    if cell == 3: return 1
    if cell == 4: return 1
    return 1e9

def dijkstra(grid, start, goal):
    pq = [(0, start)]
    dist = {start: 0}
    parent = {}

    moves = {
        'u': (-1, 0),
        'd': (1, 0),
        'l': (0, -1),
        'r': (0, 1)
    }

    while pq:
        cost, (x, y) = heapq.heappop(pq)

        if (x, y) == goal:
            break

        for m, (dx, dy) in moves.items():
            nx, ny = x + dx, y + dy

            if 0 <= nx < 20 and 0 <= ny < 20 and grid[nx][ny] != 1:
                new_cost = cost + step_cost(grid[nx][ny])
                if ny not in dist or new_cost < dist[ny]:
                    dist[ny] = new_cost
                    parent[ny] = (x, y)
                    heapq.heappush(pq, (new_cost, (nx, ny)))

```

```

if 0 <= nx < 20 and 0 <= ny < 20:
    if grid[nx][ny] != 1:
        new_cost = cost + step_cost(grid[nx][ny])
        if (nx, ny) not in dist or new_cost < dist[(nx, ny)]:
            dist[(nx, ny)] = new_cost
            parent[(nx, ny)] = ((x, y), m)
            heapq.heappush(pq, (new_cost, (nx, ny)))

    cur = goal
    path = ""
    while cur != start:
        if cur not in parent:
            return "r"*19 + "d"*19
        cur, move = parent[cur]
        path += move
    return path[::-1]

rows = []
for file in sorted(os.listdir(TEST_DIR)):
    if not file.endswith(".png"):
        continue

    image_id = file.split(".")[0]
    img = cv2.imread(os.path.join(TEST_DIR, file))

    grid = image_to_grid(img)
    start = (0, 0)
    goal = (19, 19)

    path = dijkstra(grid, start, goal)
    rows.append([image_id, path])

df = pd.DataFrame(rows, columns=["image_id", "path"])
df.to_csv("submission.csv", index=False)

print("submission.csv GENERATED SUCCESSFULLY")
print(df.head())

```