import cv2

from PIL import Image

import numpy as np

def preprocess(img, mode='block', block\_shape=None, img\_shape=None):

# if img is a path

if type(img) == str:

img = cv2.imread(img)

imheight, imwidth = img.shape[0], img.shape[1]

if mode == 'block':

new\_size = (

int(np.round(imwidth/block\_shape[1]) \* block\_shape[1]),

int(np.round(imheight/block\_shape[0]) \* block\_shape[0])

)

img = cv2.resize(img, new\_size)

blocks = []

for i in range(int(img.shape[0]/block\_shape[0])):

for j in range(int(img.shape[1]/block\_shape[1])):

block = img[

i\*block\_shape[0]:i\*block\_shape[0] + block\_shape[0],

j\*block\_shape[1]:j\*block\_shape[1] + block\_shape[1]

]

blocks.append(block)

blocks = np.array(blocks)

return blocks

else:

# choose the best dimension for padding

width\_change = np.round(imheight / img\_shape[0] \* img\_shape[1] - imwidth)

heigh\_change = np.round(imwidth / img\_shape[1] \* img\_shape[0] - imheight)

# pad image

if heigh\_change < 0 or width\_change < heigh\_change:

padsize = int(width\_change / 2)

img = np.pad(img, ((0, 0), (padsize, padsize), (0, 0)))

else:

padsize = int(height\_change / 2)

img = np.pad(img, ((padsize, padsize), (0, 0), (0, 0)))

# resize image

img = cv2.resize(img, (img\_shape[1], img\_shape[0]))

return img