

```

import cv2
import os

def load_images(folder_path):
    images = []
    for filename in os.listdir(folder_path):
        # Check if file is an image file
        if filename.endswith('.jpg') or filename.endswith('.png'):
            # Load the image using OpenCV and resize it
            image = cv2.imread(os.path.join(folder_path, filename))
            images.append(image)
    return images

def resize_images(images, width, height):
    resized_images = []
    for image in images:
        resized_image = cv2.resize(image, (width, height))
        resized_images.append(resized_image)
    return resized_images

def stitch_images(images):
    stitcher = cv2.Stitcher.create()
    (status, stitched_image) = stitcher.stitch(images)
    if status == cv2.STITCHER_OK:
        return stitched_image
    else:
        return None

def crop_image(image):
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    thresh = cv2.threshold(gray, 0, 255, cv2.THRESH_BINARY)[1]
    contours = cv2.findContours(thresh, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)[0]
    x, y, w, h = cv2.boundingRect(contours[0])
    cropped_image = image[y:y + h, x:x + w]
    return cropped_image

def preview_and_save_image(image, folder_path, folder_name):
    # Display the stitched image
    cv2.namedWindow('Stitched Image', cv2.WINDOW_NORMAL)
    cv2.imshow('Stitched Image', image)
    cv2.waitKey(0)

    # Save the stitched image
    output_filename = os.path.join(folder_path, folder_name + 'images.png')
    cv2.imwrite(output_filename, image)
    print('Stitched image saved for folder:', folder_name)

def stitch_folder(folder_path, width=800, height=800):
    # Stitch all images in a folder and save the result.
    # Load the images from the folder
    images = load_images(folder_path)

    # Check if there are at least two images in the folder
    if len(images) < 2:
        print('Not enough images in folder:', folder_path)
        return

    # Resize the images
    resized_images = resize_images(images, width, height)

    # Stitch the images
    stitched_image = stitch_images(resized_images)
    if stitched_image is None:
        print('Stitching failed for folder:', folder_path)
        return

    # Crop the stitched image
    cropped_image = crop_image(stitched_image)
    # Preview and save the stitched image
    folder_name = os.path.basename(folder_path)
    preview_and_save_image(cropped_image, folder_path, folder_name)

stitch_folder('sample_images')

```

הפונקציה `load_images` קוראת כל קובץ תמונה מתיקיית הקלט ומאחסנת אותם ברשימה.

הפונקציה `resize_images` משנה את גודל התמונות לרוחב וגובה מוגדרים מראש.

הפונקציה `stitch_images` תופרת את כל התמונות לתמונה פנורמית אחת.

הפונקציה `crop_image` חותכת את התמונה התפורה כדי להסיר גבולות שחורים או קצוות לא סדירים.

הפונקציה `preview_and_save_image` מציגה את התמונה החתוכה ושומרת אותה כקובץ בתיקיית הפלט.

הפונקציה `stitch_folder` מתזמרת את כל הפעולות הנ"ל בתיקיית הקלט. הוא מטפל בטעינת תמונות, שינוי גודלן, תפירתן, חיתוך התוצאה ולבסוף הצגת ושמירה של תמונת הפלט.