Tạo tập dữ liệu hình ảnh

Có hai phương pháp để tạo và chia sẻ tập dữ liệu hình ảnh. Hướng dẫn này sẽ cho bạn thấy làm cách nào để:

- Create an image dataset from local files in python with Dataset.push_to_hub(). This is an cách dễ dàng chỉ cần một vài bước trong python.
- Tạo tập dữ liệu hình ảnh với ImageFolder và một số siêu dữ liệu. Đây là mã không có mã giải pháp tạo nhanh tập dữ liệu hình ảnh với hàng nghìn hình ảnh.

[!TIP]
Bạn có thể kiểm soát quyền truy cập vào tập dữ liệu của mình bằng cách yêu cầu người dùng chia sẻ liên l thông tin đầu tiên. Hãy xem hướng dẫn về bộ dữ liệu Gated để biết thêm thông tin về cách kích hoạt tính năng này trên Hub.

Thư mục hình ảnh

ImageFolder là trình xây dựng tập dữ liệu được thiết kế để tải nhanh tập dữ liệu hình ảnh với một số nghìn hình ảnh mà không yêu cầu bạn phải viết bất kỳ mã nào.

[!TIP]
Hãy xem hệ thống phân cấp Mẫu phân chia để tìm hiểu thêm về cách ImageFolder tạo phân chia tập dữ liệu dựa trên cấu trúc kho lưu trữ tập dữ liệu của bạn.

ImageFolder tự động suy ra nhãn lớp của tập dữ liệu của bạn dựa trên tên thư mục.
Lưu trữ tập dữ liệu của bạn trong cấu trúc thư mục như:

```
folder/train/dog/golden_retriever.png
folder/train/dog/german_shepherd.png
folder/train/dog/chihuahua.png

folder/train/cat/maine_coon.png
folder/train/cat/bengal.png
folder/train/cat/birman.png
```

Nếu tập dữ liệu tuân theo cấu trúc ImageFolder thì bạn có thể tải nó trực tiếp bằng

load_dataset():

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("path/to/folder")
```

This is equivalent to passing imagefolder manually in load_dataset() and the directory in dữ liệu_dir :

```
>>> dataset = load_dataset("imagefolder", data_dir="/path/to/folder")
```

Bạn cũng có thể sử dụng thư mục hình ảnh để tải tập dữ liệu liên quan đến nhiều phần tách. Để làm như vậy thư mục nên có cấu trúc như sau:

```
folder/train/dog/golden_retriever.png
folder/train/cat/maine_coon.png
folder/test/dog/german_shepherd.png
thư mục/test/cat/bengal.png
```

[!WARNING]
Nếu tất cả các tập tin hình ảnh được chứa trong một thư mục duy nhất hoặc nếu chúng không cùng cấp đ cấu trúc thư mục, cột nhãn sẽ không được thêm tự động. Nếu bạn cần thì hãy đặt drop_labels=False explicitly.

Nếu có thông tin bổ sung mà bạn muốn đưa vào về tập dữ liệu của mình, như chú thích văn bản hoặc các hộp giới hạn, hãy thêm nó dưới dạng tệp siêu dữ liệu.csv trong thư mục của bạn. Điều này cho phép bạn bộ dữ liệu cho các tác vụ thị giác máy tính khác nhau như chú thích văn bản hoặc phát hiện đối tượng. bạn c cũng sử dụng tệp JSONL siêu dữ liệu.jsonl hoặc tệp Parquet siêu dữ liệu.parquet .

```
thư mục/train/metadata.csv
folder/train/0001.png
folder/train/0002.png
folder/train/0003.png
```

Bạn cũng có thể nén hình ảnh của mình và trong trường hợp này, mỗi zip phải chứa cả hình ảnh và siêu dữ liệu

```
thư mục/train.zip
thư mục/test.zip
thư mục/xác thực.zip
```

Tệp siêu dữ liệu.csv của bạn phải có trường file_name hoặc *_file_name liên kết các tệp hình ảnh với siêu dữ liệu của họ:

```
tên_tệp,tính năng bổ sung
0001.png,Đây là giá trị đầu tiên của tính năng văn bản bạn đã thêm vào hình ảnh của mình
0002.png,Đây là giá trị thứ hai của tính năng văn bản bạn đã thêm vào hình ảnh của mình
0003.png,Đây là giá trị thứ ba của tính năng văn bản bạn đã thêm vào hình ảnh của mình
```

hoặc sử dụng siêu dữ liệu.jsonl:

```
{"file_name": "0001.png", "additional_feature": "This is a first value of a text feature you added
{"file_name": "0002.png", "additional_feature": "This is a second value of a text feature you adde
{"file_name": "0003.png", "additional_feature": "This is a third value of a text feature you added
```

Ở đây file_name phải là tên của file hình ảnh bên cạnh file siêu dữ liệu. Hơn nói chung, nó phải là đường dẫn tương đối từ thư mục chứa siêu dữ liệu đến hình ảnh tài liệu.

Có thể trỏ tới nhiều hình ảnh trong mỗi hàng trong tập dữ liệu của bạn, chẳng hạn nếu cả hai đầu vào và đầu ra của bạn là hình ảnh:

```
{"input_file_name": "0001.png", "output_file_name": "0001_output.png"}
{"input_file_name": "0002.png", "output_file_name": "0002_output.png"}
{"input_file_name": "0003.png", "output_file_name": "0003_output.png"}
```

Bạn cũng có thể xác định danh sách hình ảnh. Trong trường hợp đó bạn cần đặt tên cho trường file_names h
*_file_names . Đây là một ví dụ:

```
{"frames_file_names": ["0001_t0.png", "0001_t1.png"], label: "moving_up"}
{"frames_file_names": ["0002_t0.png", "0002_t1.png"], label: "moving_down"}
{"frames_file_names": ["0003_t0.png", "0003_t1.png"], label: "moving_right"}
```

Chú thích hình ảnh

Bộ dữ liệu chú thích hình ảnh có văn bản mô tả một hình ảnh. Một ví dụ về siêu dữ liệu.csv có thể trông giống:

```
tên_tệp,văn bản
0001.png,Đây là một chú chó tha mồi vàng đang chơi với một quả bóng
0002.png,Một chú chó chăn cừu Đức
0003.png,Một con chihuahua
```

Tải tập dữ liệu bằng ImageFolder và nó sẽ tạo một cột văn bản cho chú thích hình ảnh:

```
>>> dataset = load_dataset("imagefolder", data_dir="/path/to/folder", split="train")
>>> dataset[0]["text"]
"Đây là một chú chó tha mồi vàng đang chơi với một quả bóng"
```

Phát hiện đối tượng

Bộ dữ liệu phát hiện đối tượng có các hộp giới hạn và danh mục xác định các đối tượng trong ảnh. Một ví dụ về siêu dữ liệu.jsonl có thể trông như sau:

```
{"file_name": "0001.png", "objects": {"bbox": [[302.0, 109.0, 73.0, 52.0]], "categories": [0]}}
{"file_name": "0002.png", "objects": {"bbox": [[810.0, 100.0, 57.0, 28.0]], "categories": [1]}}
{"file_name": "0003.png", "objects": {"bbox": [[160.0, 31.0, 248.0, 616.0], [741.0, 68.0, 202.0, 4
```

Tải tập dữ liệu bằng ImageFolder và nó sẽ tạo một cột đối tượng có giới hạn
hộp và các loại:

```
>>> dataset = load_dataset("imagefolder", data_dir="/path/to/folder", split="train")
>>> dataset[0]["objects"]
{"bbox": [[302.0, 109.0, 73.0, 52.0]], "categories": [0]}
```

Tải tập dữ liệu lên Hub

Once you've created a dataset, you can share it to the Hub with the push_to_hub() method.
Đảm bảo bạn đã cài đặt thư viện ôm mặt_hub và bạn đã đăng nhập vào

Hugging Face account (see the Upload with Python tutorial for more details).

Upload your dataset with push_to_hub():

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("imagefolder", data_dir="/path/to/folder", split="train")
>>> dataset.push_to_hub("stevhliu/my-image-captioning-dataset")
```

Bộ dữ liệu Web

Định dạng WebDataset dựa trên kho lưu trữ TAR và phù hợp với các tập dữ liệu hình ảnh lớn. Indeed you can group your images in TAR archives (e.g. 1GB of images per TAR archive) and có hàng ngàn kho lưu trữ TAR:

```
thư mục/train/00000.tar
thư mục/train/00001.tar
thư mục/train/00002.tar
None
```

Trong kho lưu trữ, mỗi ví dụ được tạo từ các tệp có chung tiền tố:

```
e39871fd9fd74f55.jpg
e39871fd9fd74f55.json
f18b91585c4d3f3e.jpg
f18b91585c4d3f3e.json
ede6e66b2fb59aab.jpg
ede6e66b2fb59aab.json
ed600d57fcee4f94.jpg
ed600d57fcee4f94.json
None
```

Ví dụ: bạn có thể đặt nhãn/chú thích/hộp giới hạn cho hình ảnh của mình bằng cách sử dụng tệp JSON hoặc

Load your WebDataset and it will create on column per file suffix (here "jpg" and "json"):

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("webdataset", data_dir="/path/to/folder", split="train")
>>> dataset[0]["json"]
{"bbox": [[302.0, 109.0, 73.0, 52.0]], "categories": [0]}
```

Cũng có thể có một số hình ảnh cho mỗi ví dụ như thế này:

```
e39871fd9fd74f55.input.jpg
e39871fd9fd74f55.output.jpg
e39871fd9fd74f55.json
f18b91585c4d3f3e.input.jpg
f18b91585c4d3f3e.output.jpg
f18b91585c4d3f3e.json
None
```

Để biết thêm chi tiết về định dạng WebDataset và thư viện python, vui lòng kiểm tra Tài liệu WebDataset.