Tải dữ liệu hình ảnh

Bộ dữ liệu hình ảnh có các cột loại Hình ảnh chứa các đối tượng PIL.

> [!TIP]
> Để làm việc với tập dữ liệu hình ảnh, bạn cần cài đặt phần phụ thuộc tầm nhìn. Kiểm tra hướng dẫn cài đặt để tìm hiểu cách cài đặt nó.

Khi bạn tải tập dữ liệu hình ảnh và gọi cột hình ảnh, hình ảnh sẽ được giải mã dưới dạng PIL Hình ảnh:

```
>>> from datasets import load_dataset, Image

>>> dataset = load_dataset("beans", split="train")
>>> dataset[0]["image"]
```

> [!WARNING]
> Lập chỉ mục vào tập dữ liệu hình ảnh bằng cách sử dụng chỉ mục hàng trước rồi đến cột hình ảnh - dataset[0]["image"] - to avoid decoding and resampling all the image objects in the tập dữ liệu. Mặt khác, đây có thể là một quá trình chậm và tốn thời gian nếu bạn có một lượng lớn tập dữ liệu.

Để biết hướng dẫn về cách tải bất kỳ loại tập dữ liệu nào, hãy xem hướng dẫn tải chung.

Tệp cục bộ

You can load a dataset from the image path. Use the cast_column() function to accept a cột đường dẫn file ảnh và giải mã thành ảnh PIL bằng tính năng Image:

```
>>> from datasets import Dataset, Image

>>> dataset = Dataset.from_dict({"image": ["path/to/image_1", "path/to/image_2", ..., "path/to/ima
>>> dataset[0]["image"]
<PIL.PngImagePlugin.PngImageFile image mode=RGBA size=1200x215 at 0x15E6D7160>]
```

Nếu bạn chỉ muốn tải đường dẫn cơ bản tới tập dữ liệu hình ảnh mà không giải mã hình ảnh

object, set decode=False in the Image feature:

```
>>> dataset = load_dataset("beans", split="train").cast_column("image", Image(decode=False))
>>> dataset[0]["image"]
{'bytes': None,
 'đường dẫn': '/root/.cache/huggingface/datasets/downloads/extracted/b0a21163f78769a2cf11f58dfc767f
```

Thư mục hình ảnh

Bạn cũng có thể tải tập dữ liệu bằng trình tạo tập dữ liệu ImageFolder mà không yêu cầu
viết một trình tải dữ liệu tùy chỉnh. Điều này làm cho ImageFolder trở nên lý tưởng để tạo và tải nhanh chóng
bộ dữ liệu hình ảnh với hàng nghìn hình ảnh cho các nhiệm vụ thị giác khác nhau. Tập dữ liệu hình ảnh của b
cấu trúc sẽ trông như thế này:

```
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
```

Ngoài ra, nó phải có siêu dữ liệu, ví dụ:

```
thư mục/train/metadata.csv
folder/train/0001.png
folder/train/0002.png
folder/train/0003.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("username/dataset_name")
>>> # OR locally:
>>> dataset = load_dataset("/path/to/folder")
```

For local datasets, this is equivalent to passing imagefolder manually in load_dataset() and
thư mục trong data_dir :

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

Sau đó, bạn có thể truy cập video dưới dạng đối tượng PIL.Image:

```
>>> dataset["train"][0]
{"image": <PIL.PngImagePlugin.PngImageFile image mode=RGBA size=1200x215 at 0x15E6D7160>, "label'

>>> dataset["train"][-1]
{"image": <PIL.PngImagePlugin.PngImageFile image mode=RGBA size=1200x215 at 0x15E8DAD30>, "label
```

To ignore the information in the metadata file, set drop_metadata=True in load_dataset():

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("username/dataset_with_metadata", drop_metadata=True)
```

Nếu bạn không có tệp siêu dữ liệu, ImageFolder sẽ tự động suy ra tên nhãn từ
tên thư mục.
If you want to drop automatically created labels, set drop_labels=True .
Trong trường hợp này, tập dữ liệu của bạn sẽ chỉ chứa một cột hình ảnh:

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("username/dataset_without_metadata", drop_labels=True)
```

Cuối cùng, đối số bộ lọc cho phép bạn chỉ tải một tập hợp con của tập dữ liệu, dựa trên một điều kiện
trên nhãn hoặc siêu dữ liệu. Điều này đặc biệt hữu ích nếu siêu dữ liệu ở định dạng Parquet,

vì định dạng này cho phép lọc nhanh. Bạn cũng nên sử dụng đối số này với streaming=True , because by default the dataset is fully downloaded before filtering.

```
>>> filters = [("label", "=", 0)]
>>> dataset = load_dataset("username/dataset_name", streaming=True, filters=filters)
```

[!TIP]
Để biết thêm thông tin về cách tạo tập dữ liệu ImageFolder của riêng bạn, hãy xem Tạo hướng dẫn tập dữ liệu hình ảnh.

Bộ dữ liệu Web

Định dạng WebDataset dựa trên một thư mục lưu trữ TAR và phù hợp với hình ảnh lớn bộ dữ liệu.
Because of their size, WebDatasets are generally loaded in streaming mode (using streaming=True).

Bạn có thể tải WebDataset như thế này:

```
>>> from datasets import load_dataset

>>> dataset = load_dataset("webdataset", data_dir="/path/to/folder", streaming=True)
```

Giải mã hình ảnh

Theo mặc định, hình ảnh được giải mã tuần tự dưới dạng PIL.Images khi bạn lặp lại trên tập dữ liệu. Tuy nhiên, có thể tăng tốc đáng kể tập dữ liệu bằng cách sử dụng giải mã đa luồng:

```
>>> import os
>>> num_threads = num_threads = min(32, (os.cpu_count() or 1) + 4)
>>> dataset = dataset.decode(num_threads=num_threads)
>>> for example in dataset:# up to 20 times faster !
None
```

Bạn có thể kích hoạt đa luồng bằng cách sử dụng num_threads. Điều này đặc biệt hữu ích để tăng tốc

truyền dữ liệu từ xa.
However it can be slower than num_threads=0 for local data on fast disks.

Nếu bạn không quan tâm đến những hình ảnh được giải mã dưới dạng PIL.Images và muốn truy cập thay vào đó, path/byte, bạn có thể tắt giải mã:

```
>>> dataset = dataset.decode(False)
```

Note: IterableDataset.decode() is only available for streaming datasets at the moment.