📽️ Video and Audio Datasets

- **YouTube-8M**: A large-scale video classification dataset with approximately 8 million videos annotated with a vocabulary of 4,800 visual entities.
- **LibriSpeech**: An English speech dataset containing approximately 1,000 hours of audio, suitable for training and evaluating speech recognition systems.

📹 1. YouTube-8M

🔹 Contains only **video features**, not raw videos (due to copyright). You use pre-extracted features (frame-level or video-level) with labels.

✅ Step 1: Download Dataset Features

Google provides TensorFlow-compatible format:

- GCS link: `gs://youtube8m-ml/2/frame/train/train*.tfrecord`
- Use `gsutil` to download from Google Cloud.

Install `gsutil` and run:

```
bash
CopyEdit
gsutil -m cp "gs://youtube8m-ml/2/frame/train/train00*.tfrecord" ./youtube8m/
```

✅ Step 2: Parse the `.tfrecord` files in Python

```
python
CopyEdit
import tensorflow as tf

# Define feature schema
context_features = {
    "id": tf.io.FixedLenFeature([], tf.string),
    "labels": tf.io.VarLenFeature(tf.int64),
}
sequence_features = {
    "rgb": tf.io.FixedLenSequenceFeature([], dtype=tf.string)
}

# Load TFRecord
filename = "youtube8m/train00.tfrecord"
raw_dataset = tf.data.TFRecordDataset([filename])

# Example parse function
def parse_example(serialized_example):
    context, sequence = tf.io.parse_single_sequence_example(
        serialized_example,
        context_features=context_features,
```

```
        sequence_features=sequence_features
    )
    video_id = context["id"]
    labels = tf.sparse.to_dense(context["labels"])
    rgb_frames = sequence["rgb"]
    return video_id, labels, rgb_frames

# Load a few samples
for example in raw_dataset.take(1):
    vid, labels, frames = parse_example(example)
    print("Video ID:", vid.numpy())
    print("Labels:", labels.numpy())
    print("Frame count:", len(frames))
```

🔊 2. LibriSpeech (ASR Dataset)

🔽 Commonly used for speech recognition tasks. Available via `torchaudio` or `datasets`.

✅ Option 1: Use `datasets` (from HuggingFace)

```
bash
CopyEdit
pip install datasets
python
CopyEdit
from datasets import load_dataset

# Load a subset of LibriSpeech
dataset = load_dataset("librispeech_asr", "clean", split="train.100")

# Preview
print(dataset[0]['text'])
print(dataset[0]['audio'])

# Play audio
import IPython.display as ipd
ipd.Audio(dataset[0]['audio']['array'],
rate=dataset[0]['audio']['sampling_rate'])
```

✅ Option 2: Use `torchaudio` (for PyTorch users)

```
bash
CopyEdit
pip install torchaudio
python
CopyEdit
import torchaudio

# Download test-clean split
dataset = torchaudio.datasets.LIBRISPEECH("./data", url="test-clean",
download=True)
```

```
# Access one sample
waveform, sample_rate, transcript, speaker_id, chapter_id, utterance_id =
dataset[0]

print("Transcript:", transcript)
print("Sample rate:", sample_rate)
```