□ Scientific and Specialized Datasets

- **CERN Open Data**: Provides 300TB of data from the Large Hadron Collider experiments, accessible for educational and research purposes.
- **80 Million Tiny Images**: A dataset containing 79 million low-resolution images, useful for various computer vision tasks.

Scientific and Specialized Datasets

🔬 1. CERN Open Data

The **CERN Open Data Portal** offers over 300TB of data from experiments at the Large Hadron Collider (LHC), primarily in the **ROOT** file format—commonly used in high-energy physics. You can access and analyze these datasets using Python with the `uproot` library.

✓ *Prerequisites*

Install necessary libraries:

```
bash
CopyEdit
pip install uproot pandas
```

✓ *Sample Code: Accessing a ROOT File*

```
python
CopyEdit
import uproot
import pandas as pd

# Example CERN Open Data ROOT file (update with actual file or download
locally)
root_file_url = "https://opendata.cern.ch/record/12346/files/ATLAS_data.root"

# Open the ROOT file using uproot
with uproot.open(root_file_url) as file:
    print("Available Trees:")
    print(file.keys())  # Lists all trees in the file

    # Load a specific tree (update based on actual tree name)
    tree = file["mini"]  # Replace with actual tree key
    df = tree.arrays(library="pd")  # Convert to pandas DataFrame

    print("Sample data:")
    print(df.head())
```

📌 **Note**: Tree names and structure may vary depending on the dataset. You can explore CERN datasets at: https://opendata.cern.ch

---

🖼️ 2. 80 Million Tiny Images

Originally released by MIT, the **80 Million Tiny Images** dataset consisted of 79 million 32×32 color images collected from the web. It was widely used in computer vision research. However, the dataset was **withdrawn in 2020** due to issues related to offensive labels.

⚠️ Ethical Consideration

Due to the nature of the original labeling, the dataset has been officially deprecated. As an alternative, you may use **CIFAR-10** or **TinyImageNet**, which are curated and safe for academic use.

✅ Alternative: Using CIFAR-10 as a Tiny Image Dataset

```
python
CopyEdit
from torchvision.datasets import CIFAR10
from torchvision import transforms
import matplotlib.pyplot as plt

# Define a transform to simulate low-resolution image processing
transform = transforms.Compose([
    transforms.Resize((32, 32)),
    transforms.ToTensor()
])

# Load CIFAR-10 dataset
dataset = CIFAR10(root='./data', train=True, download=True,
transform=transform)

# Display a few sample images
for i in range(5):
    image, label = dataset[i]
    plt.imshow(image.permute(1, 2, 0))
    plt.title(f"Class Label: {label}")
    plt.axis('off')
    plt.show()
```