```
import torch
from torch.utils.data import DataLoader, Dataset
from torchvision import transforms
from torchvision.datasets import CIFAR10
import matplotlib.pyplot as plt

# Define a custom dataset class (you can also use built-in datasets like CIFAR10)
class CustomDataset(Dataset):
    def _init_(self, data, transform=None):
        self.data = data
        self.transform = transform

    def _len_(self):
        return len(self.data)

    def _getitem_(self, idx):
        sample = self.data[idx]

        if self.transform:
            sample = self.transform(sample)

        return sample

# Define any data transformations you need (e.g., resizing, normalization)
transform = transforms.Compose([
    transforms.ToTensor(),  # Convert to tensor
    transforms.Normalize((0.5, 0.5, 0.5), (0.5, 0.5, 0.5))  # Normalize to [-1, 1]
])

# Create a dataset (you can replace this with your own dataset)
# Here, we'll use the CIFAR-10 dataset as an example
cifar_dataset = CIFAR10(root='./data', train=True, download=True, transform=transform)

# Define batch size
batch_size = 64

# Create a DataLoader for your dataset
dataloader = DataLoader(cifar_dataset, batch_size=batch_size, shuffle=True)

# Iterate over the dataset using the DataLoader
for batch_data in dataloader:
    # batch_data will contain a batch of samples (images and labels if available)
    # Perform your training or evaluation here
    images, labels = batch_data  # For example, if CIFAR-10 includes labels

    # Visualize a sample from the batch (optional)
    plt.imshow(images[0].permute(1, 2, 0).numpy())  # Assuming images are tensors in [C, H, W] format
    plt.show()

    break  # Stop after processing one batch for demonstration purpose
```

```
Downloading https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz to ./data/cifar-10-p
100%|██████████| 170498071/170498071 [00:02<00:00, 68893481.13it/s]
Extracting ./data/cifar-10-python.tar.gz to ./data
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data
```