```
import os
import pandas as pd
import matplotlib.pyplot as plt
from torchvision.io import read_image
from torchvision import transforms
from torch.utils.data import Dataset, DataLoader, random_split


class CustomDataset(Dataset):
    def __init__(self, annotations_file, img_dir, transform=None):
        self.img_labels = pd.read_csv(annotations_file)
        self.img_dir = img_dir
        self.transform = transform

    def __len__(self):
        return len(self.img_labels)

    def __getitem__(self, idx):
      img_path = os.path.join(self.img_dir, self.img_labels.iloc[idx, 0])
      image = read_image(img_path).float() / 255.
      label = self.img_labels.iloc[idx, 1]
      if self.transform:
        image = self.transform(image)
      return image, label



transform = transforms.Compose([
    transforms.Resize((224, 224), antialias=True),
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomRotation(10),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])
])



dataset = CustomDataset(annotations_file='/content/drive/MyDrive/ArcFace/data/dataset_labels.csv', img_dir='/content/drive/MyDrive/ArcFace/d


train_val_split = int(len(dataset) * 0.9)
test_split = len(dataset) - train_val_split
train_val_dataset, test_dataset = random_split(dataset, [train_val_split, test_split])


train_split = int(train_val_split * 0.88)
val_split = train_val_split - train_split
train_dataset, val_dataset = random_split(train_val_dataset, [train_split, val_split])


batch_size = 8
train_loader = DataLoader(train_dataset, batch_size=batch_size, shuffle=True)
val_loader = DataLoader(val_dataset, batch_size=batch_size)
test_loader = DataLoader(test_dataset, batch_size=batch_size)


import matplotlib.pyplot as plt
import torch


def show_image(img, mean, std):
    img = img.permute(1, 2, 0)
    img = img * std + mean
    plt.imshow(img)
    plt.axis('off')
    plt.show()



for images, _ in train_loader:
    first_image = images[0]
    show_image(first_image, mean=torch.tensor([0.485, 0.456, 0.406]), std=torch.tensor([0.229, 0.224, 0.225]))
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB
```