Loading Image Data:

dataset = datasets.ImageFolder ('path',
transform = transforms)

image folder structure:

root/dog/1.png
root/dog/2.png
root/cat/3.png
root/cat/4.png

Transforms:

transforms = transforms.Compose([transforms.Resize(255),
transforms.CenterCrop(224),
transforms.ToTensor()])

Converting to tensor —— transforms.ToTensor()
Data Loaders:

data_loader = torch.utils.data.DataLoader
  (dataset, batch_size = 32,
   shuffle = True)

next(iter(data_loader))

data augmentation:
  Introducing randomness in input data itself.
  transforms & RandomRotation, RandomResizedCrop, etc.

Normalization:
  transforms. Normalize
  
  input[channel] = (input[channel] - mean[channel]) / std[channel]

  Helps keep data near zero, making backpropagation more stable.