Loading Image Data:

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

image folder structure:

```
root/dog/~.png
root/dog/~.png
  :
root/cat/~.png
root/cat/~.png
  :
```

one folder for each class

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)

$$\text{next}\,(\underbrace{\text{iter}\,(\underbrace{\text{data loader}}_{\text{generator}})}_{\text{iterator}})$$

Data Augmentation:

Introducing randomness in input data itself.

transforms $\leftarrow$ RandomRotation, RandomResizedCrop, etc.

Normalization:

transforms.Normalize

input[channel] = (input[channel] - mean[channel]) / std[channel]

Helps keep data near zero, making backpropagation more stable.