The Object Detection Dataset

Eng Teong Cheah

Contents

Downloading the Dataset

```
%matplotlib inline
import d2l
from mxnet import gluon, image, np, npx
import os

npx.set_np()

# Saved in the d2l package for later use
def download_pikachu(data_dir):
    root_url = ('https://apache-mxnet.s3-accelerate.amazonaws.com/'
                'gluon/dataset/pikachu/')
    dataset = {'train.rec': 'e6bcb6ffba1ac04ff8a9b1115e650af56ee969c8',
               'train.idx': 'dcf7318b2602c06428b9988470c731621716c393',
               'val.rec': 'd6c33f799b4d058e82f2cb5bd9a976f69d72d520'}
    for k, v in dataset.items():
        gluon.utils.download(
            root_url + k, os.path.join(data_dir, k), sha1_hash=v)
```

Reading the Dataset

```
# Saved in the d2l package for later use
def load_data_pikachu(batch_size, edge_size=256):
    """Load the pikachu dataset"""
    data_dir = '../data/pikachu'
    download_pikachu(data_dir)
    train_iter = image.ImageDetIter(
        path_imgrec=os.path.join(data_dir, 'train.rec'),
        path_imgidx=os.path.join(data_dir, 'train.idx'),
        batch_size=batch_size,
        data_shape=(3, edge_size, edge_size),  # The shape of the output image
        shuffle=True,  # Read the dataset in random order
        rand_crop=1,  # The probability of random cropping is 1
        min_object_covered=0.95, max_attempts=200)
    val_iter = image.ImageDetIter(
        path_imgrec=os.path.join(data_dir, 'val.rec'), batch_size=batch_size,
        data_shape=(3, edge_size, edge_size), shuffle=False)
    return train_iter, val_iter
```

Reading the Dataset

```
batch_size, edge_size = 32, 256
train_iter, _ = load_data_pikachu(batch_size, edge_size)
batch = train_iter.next()
batch.data[0].shape, batch.label[0].shape
```

Demonstration

```
imgs = (batch.data[0][0:10].transpose((0, 2, 3, 1))) / 255
axes = d2l.show_images(imgs, 2, 5).flatten()
for ax, label in zip(axes, batch.label[0][0:10]):
    d2l.show_bboxes(ax, [label[0][1:5] * edge_size], colors=['w'])
```

Thank You !

Does anyone have any questions?

Twitter: @walkercet
Blog: https://ceteongvanness.wordpress.com

Resources

Dive into Deep Learning