

predict_SSD-MobileNet

May 5, 2020

1 Preparation for SSD

Source: [Single Shot MultiBox Detector Implementation in Pytorch](#)

VGG SSD

```
wget -P models https://storage.googleapis.com/models-hao/vgg16-ssd-mp-0_7726.pth
wget -P models https://storage.googleapis.com/models-hao/voc-model-labels.txt
```

MobileNetV1 SSD

```
wget -P models https://storage.googleapis.com/models-hao/mobilenet-v1-ssd-mp-0_675.pth
wget -P models https://storage.googleapis.com/models-hao/voc-model-labels.txt
```

MobileNetV2 SSD Lite

```
wget -P models https://storage.googleapis.com/models-hao/mb2-ssd-lite-mp-0_686.pth
wget -P models https://storage.googleapis.com/models-hao/voc-model-labels.txt
```

2 SSD with MobileNetV1, 2 & VGG

```
[1]: from vision.ssd.vgg_ssd import create_vgg_ssd, create_vgg_ssd_predictor
from vision.ssd.mobilenetv1_ssd import create_mobilenetv1_ssd,
    ↳ create_mobilenetv1_ssd_predictor
from vision.ssd.mobilenetv1_ssd_lite import create_mobilenetv1_ssd_lite,
    ↳ create_mobilenetv1_ssd_lite_predictor
from vision.ssd.squeezenet_ssd_lite import create_squeezenet_ssd_lite,
    ↳ create_squeezenet_ssd_lite_predictor
from vision.ssd.mobilenet_v2_ssd_lite import create_mobilenetv2_ssd_lite,
    ↳ create_mobilenetv2_ssd_lite_predictor
from vision.utils.misc import Timer
import cv2
import sys
import numpy as np
```

```
[2]: label_path = 'models/voc-model-labels.txt'
img_path = 'testset-img/'
det_dir = 'predicted_boxes/'
```

```

class_names = [name.strip() for name in open(label_path).readlines()]
num_classes = len(class_names)
net_types = ['vgg16-ssd', 'mb1-ssd', 'mb2-ssd-lite']

```

```

[3]: def make_predictor(net_type):
    if net_type == 'vgg16-ssd':
        model_path = 'models/vgg16-ssd-mp-0_7726.pth'
        net = create_vgg_ssd(len(class_names), is_test=True)
        net.load(model_path)
        predictor = create_vgg_ssd_predictor(net, candidate_size=200)
    elif net_type == 'mb1-ssd':
        model_path = 'models/mobilenetv1-ssd-mp-0_675.pth'
        net = create_mobilenetv1_ssd(len(class_names), is_test=True)
        net.load(model_path)
        predictor = create_mobilenetv1_ssd_predictor(net, candidate_size=200)
    elif net_type == 'mb2-ssd-lite':
        model_path = 'models/mb2-ssd-lite-mp-0_686.pth'
        net = create_mobilenetv2_ssd_lite(len(class_names), is_test=True)
        net.load(model_path)
        predictor = create_mobilenetv2_ssd_lite_predictor(net,
↪candidate_size=200)
    else:
        print("The net type is wrong. It should be one of vgg16-ssd, mb1-ssd,
↪and mb1-ssd-lite.")

    return predictor

```

3 Result

```

[4]: from os import listdir
    from os.path import isfile, join
    import json

```

```

[5]: def read_image(image_file):
    image = cv2.imread(image_file)
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
    return image

```

```

[6]: def batch_prediction(net_type, img_path, det_dir, print_progress = True):
    # loads a model
    predictor = make_predictor(net_type)
    if print_progress:
        print('%s is loaded.' % (net_type))

    # detects objects

```

```

img_names = [f for f in listdir(img_path) if f.endswith('.jpg')]

result_dict = {}
label = ['person', 'car']
for l in label:
    result_dict[l] = {}
    for f in img_names:
        image = read_image(img_path + f)
        boxes, classes, confs = predictor.predict(image, 10, 0.4)
        result_dict[l][f] = {}
        result_dict[l][f]['boxes'] = []
        result_dict[l][f]['scores'] = []
        for i, c in enumerate(classes):
            # 6: bus, 7: car, 15: person
            if (l == 'person' and c == 15) or (l == 'car' and c in [6, 7]):
                result_dict[l][f]['boxes'].append(boxes[i].int().tolist())
                result_dict[l][f]['scores'].append(confs[i].float().
→tolist())
        if print_progress:
            print('%s is predicted in %d images.' % (l, len(img_names)))

    for l in label:
        with open(det_dir+'predicted_boxes-'+net_type+'-'+l+'.json', 'w') as fp:
            json.dump(result_dict[l], fp)
        if print_progress:
            print('%s is writeen.' % (l))

```

```

[7]: for net_type in net_types:
    batch_prediction(net_type, img_path, det_dir)

```

```

vgg16-ssd is loaded.
person is predicted in 100 images.
car is predicted in 100 images.
person is writeen.
car is writeen.
mb1-ssd is loaded.
person is predicted in 100 images.
car is predicted in 100 images.
person is writeen.
car is writeen.
mb2-ssd-lite is loaded.
person is predicted in 100 images.
car is predicted in 100 images.
person is writeen.
car is writeen.

```