Компьютерное зрение

Практический курс

Савельева Юлия Олеговна
i.o.saveleva.kpfu@gmail.com

2-й семестр, 13.02.2021 г.

Организационная информация

- **Оценки за практику:** https://docs.google.com/spreadsheets/d/1BVscwoYSx5l82HAlbdFoVOocNySBDJOMhXK10UDQzcA/edit?usp=sharing
- **Ссылка на материалы курса:** https://github.com/IuliiaSaveleva/Computer_Vision_course_students_2021
- **Мой username на Bitbucket:** **IuliiaSaveleva**

Image Retrieval

Stanford Online Products

1) Скачать датасет **Stanford Online Products** (SOP):

https://cvgl.stanford.edu/projects/lifted_struct/

Структура датасета:

SOP/<class_name>_final/<product_id>_<img_no>.JPG

2) Перестроить датасет на основе pickle файла разбиения на train, valid и test (Practice1/SOP_train_valid_test_split.pickle на github).

Структура пикла:

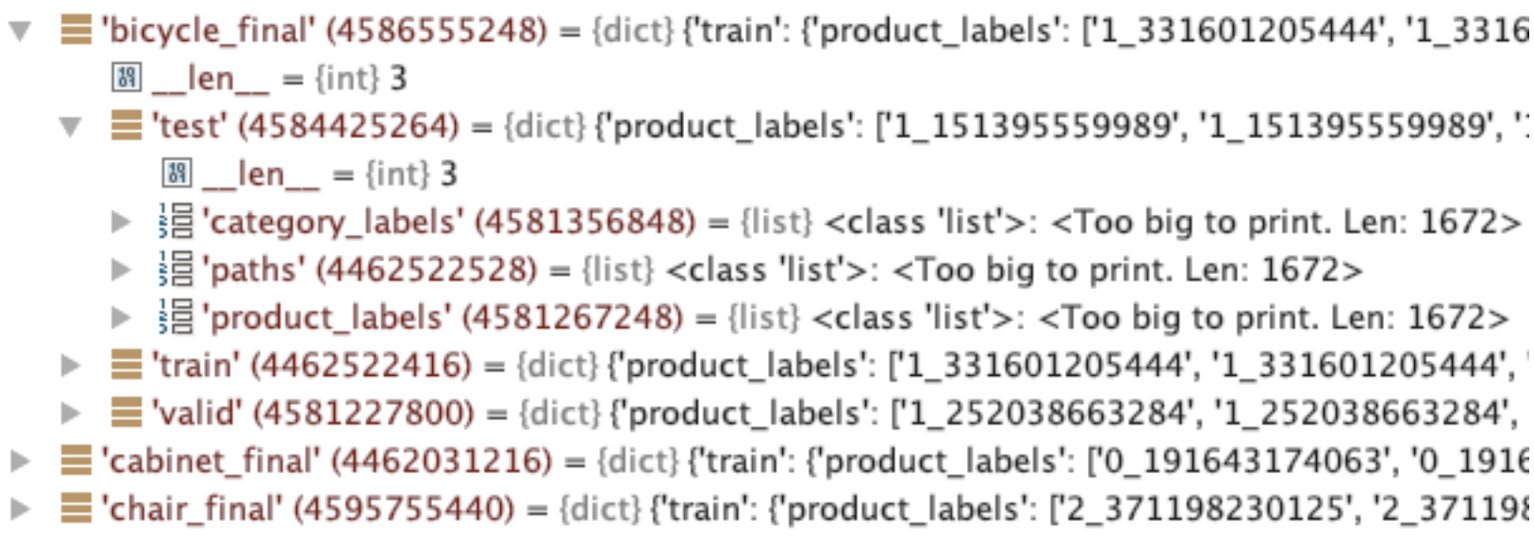

Image Retrieval

Stanford Online Products

▼ 'paths' (4462522528) = {list} <class 'list'>: <Too big to print. Len: 1672>
- __len__ = {int} 1672
- 0000 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_10.JPG
- 0001 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_4.JPG
- 0002 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_7.JPG
- 0003 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_1.JPG
- 0004 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_0.JPG
- 0005 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_2.JPG
- 0006 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_5.JPG
- 0007 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_8.JPG
- 0008 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_6.JPG
- 0009 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_9.JPG
- 0010 = {str_} /Stanford_Online_Products/bicycle_final/151395559989_3.JPG
- 0011 = {str_} /Stanford_Online_Products/bicycle_final/400967143545_6.JPG
- 0012 = {str_} /Stanford_Online_Products/bicycle_final/400967143545_2.JPG
- 0013 = {str_} /Stanford_Online_Products/bicycle_final/400967143545_7.JPG
- 0014 = {str_} /Stanford_Online_Products/bicycle_final/400967143545_10.JPG
- 0015 = {str_} /Stanford_Online_Products/bicycle_final/400967143545_8.JPG
- 0016 = {str_} /Stanford_Online_Products/bicycle_final/400967143545_3.JPG
- 0017 = {str_} /Stanford_Online_Products/bicycle_final/400967143545_4.JPG

▼ 'product_labels' (4581267248) = {lis
- __len__ = {int} 1672
- 0000 = {str} '1_151395559989'
- 0001 = {str} '1_151395559989'
- 0002 = {str} '1_151395559989'
- 0003 = {str} '1_151395559989'
- 0004 = {str} '1_151395559989'
- 0005 = {str} '1_151395559989'
- 0006 = {str} '1_151395559989'
- 0007 = {str} '1_151395559989'
- 0008 = {str} '1_151395559989'
- 0009 = {str} '1_151395559989'
- 0010 = {str} '1_151395559989'
- 0011 = {str} '1_400967143545'
- 0012 = {str} '1_400967143545'
- 0013 = {str} '1_400967143545'
- 0014 = {str} '1_400967143545'
- 0015 = {str} '1_400967143545'
- 0016 = {str} '1_400967143545'
- 0017 = {str} '1_400967143545'

Сформировать датасет со структурой:

SOP_retrieval/<set>/<product_label>/<class_name>/<class_name>_<product_id>_<img_no>.JPG

Например: SOP_retrieval/test/1_151395559989/bicycle_final/bicycle_final_151395559989_10.JPG

Image Retrieval

Архитектура

3) Забрать код ResNet:

https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py

Мы будем использовать ResNet50, предобученный на ImageNet, но необходимо предусмотреть возможность подменять последний полносвязный слой на слой с нужным нам количеством классов, который будет случайно инициализирован.

```
def _resnet(
    arch: str,
    block: Type[Union[BasicBlock, Bottleneck]],
    layers: List[int],
    pretrained: bool,
    progress: bool,
    **kwargs: Any
) -> ResNet:
    model = ResNet(block, layers, **kwargs)
    if pretrained:
        state_dict = load_state_dict_from_url(model_urls[arch],
                                              progress=progress)
        model.load_state_dict(state_dict)
    return model
```

Можно модифицировать метод _resnet() так, чтобы при выполнении условия *pretrained* из *state_dict* с предобученными весами удалялась информация о весах для тех слоев, чьи размеры с размерами слоя текущей модели не совпадают:

```
state_dict = {k: v for k, v in state_dict.items() if k in model_dict and v.size() == model_dict[k].size()}
```

Image Retrieval

Препроцессинг

```
from torchvision import transforms, datasets
```

4) Использовать следующий препроцессинг на этапе тестирования сети:
https://pytorch.org/hub/pytorch_vision_resnet/

```
preprocess = transforms.Compose([
    transforms.Resize(256),
    transforms.CenterCrop(224),
    transforms.ToTensor(),
    transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
])
```

На этапе обучения необходимо подменить 2 первые операции:

```
transforms.RandomResizedCrop(224),
transforms.RandomHorizontalFlip()
```

5) Создайте обучающий датасет:
datasets.ImageFolder(root=…/SOP_retrieval/train, transform=train_preprocess)

Дедлайн 20.02.2020 00:00