Компьютерное зрение

Практический курс

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

2-й семестр, 27.03.2021 г.

Segmentation

Dataset

Статья с методом:
https://arxiv.org/pdf/1505.04597.pdf

Ссылка на датасет:
https://cocodataset.org/#download
2017 Train images [118K/18GB]
2017 Val images [5K/1GB]
2017 Train/Val annotations [241MB]

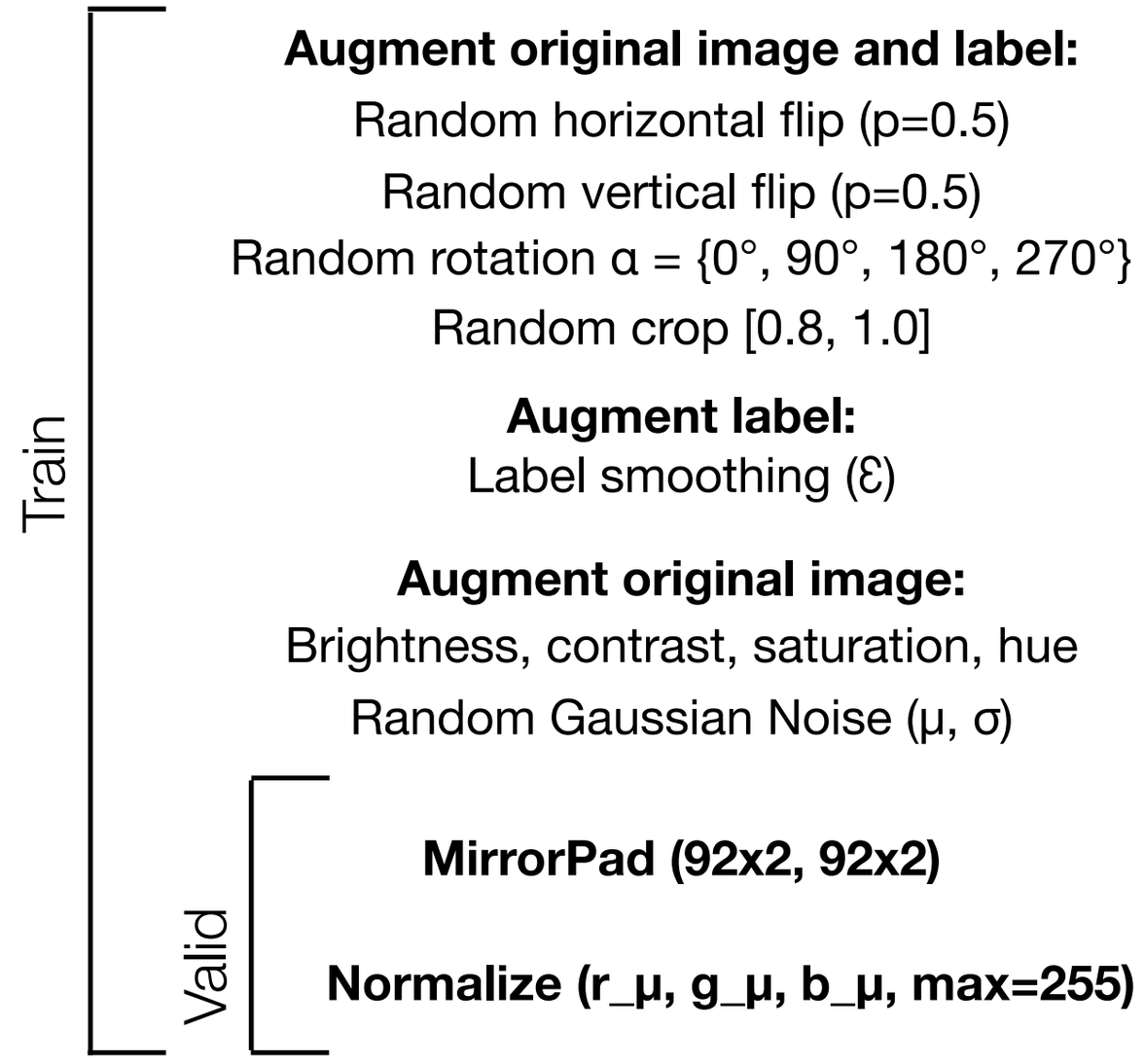
COCO API: https://github.com/cocodataset/cocoapi/blob/master/PythonAPI/pycocoDemo.ipynb - демо для чтения картинок и аннотаций

1) Скачать датасет и настроить даталоудер для обучающей и валидационной выборки на чтение и предобработку картинок.
2) Реализовать архитектуру UNet.
3) Реализовать целевую функцию и обеспечить проход картинки через пайплайн.

Segmentation

Preprocessing

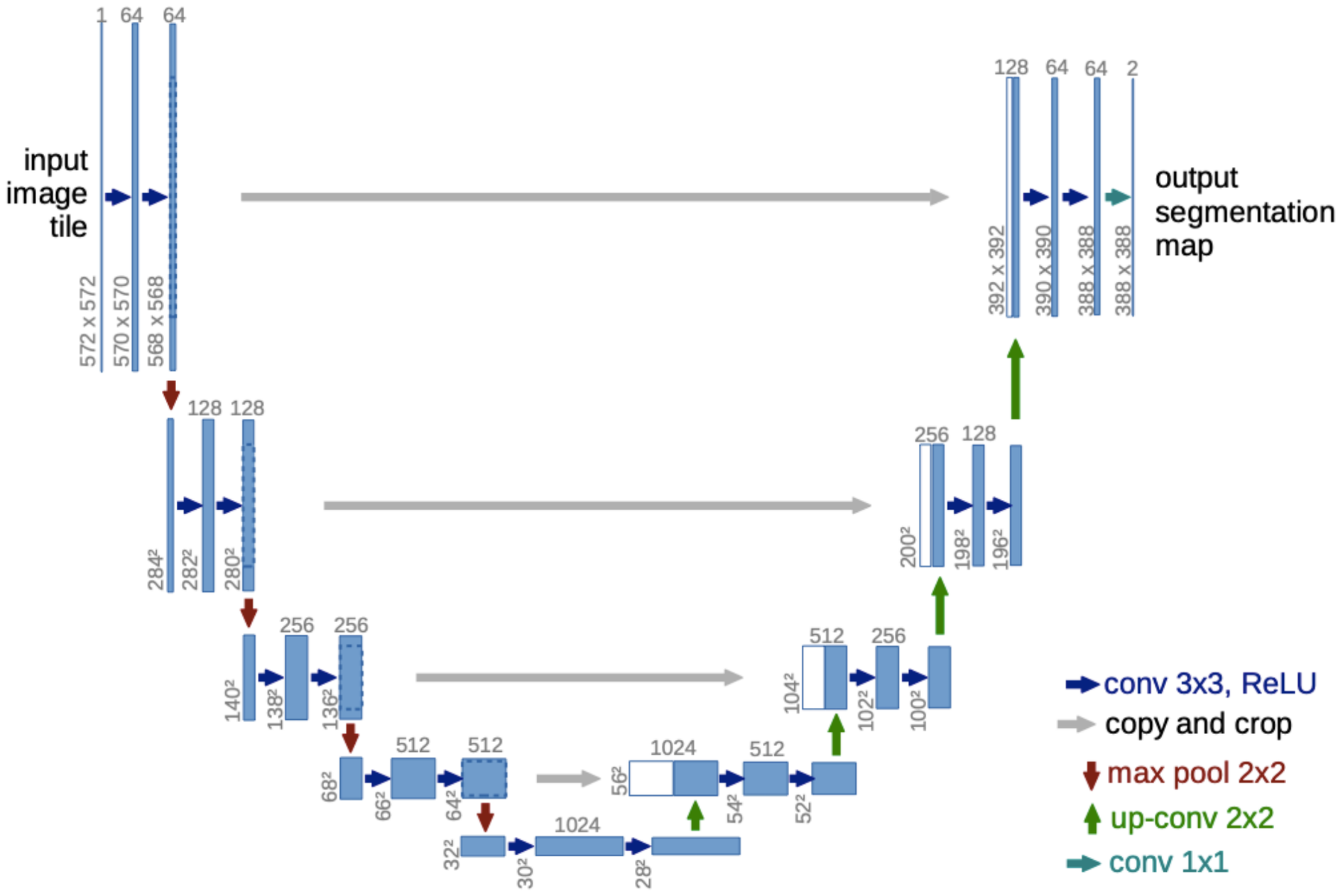
Представить лейблы сегментации как одноканальные картинки, где в каждом пикселе находится номер класса и проводить аугментацию (которая влияет на координаты объектов относительно изображения) для «картинки лейблов» и исходного изображения одинаковыми функциями.




```
torchvision.transforms.ColorJitter(brightness=0.5, contrast=0.5, saturation=0.5, hue=0.5)
```

Segmentation

Architecture



nn.ConvTranspose2d(in_channels=in_channels, out_channels=out_channels, kernel_size=2, stride=2)

Segmentation

Loss

Посчитать баланс классов на всей обучающей выборке и получить матрицу весов **W**

```
class WeightedLabelSmoothLoss(nn.Module):
    def __init__(self, W, smooth_eps=0.0):
        super(LabelSmoothLoss, self).__init__()
        self.smooth_eps = smooth_eps
        self.class_weights = W

    def forward(self, inputs, targets):
        log_probs = nn.functional.log_softmax(inputs, dim=-1)
        smooth_targets = inputs.new_ones(inputs.size()) * \
                         self.smooth_eps / (inputs.size(-1) - 1.)
        smooth_targets.scatter_(-1, targets.unsqueeze(-1), (1. - self.smooth_eps))
        loss = (self.class_weights * smooth_targets * (- log_probs)).sum(dim=-1).mean()
        return loss
```

Проверить, что сеть оверфитит на одной картинке.

Дедлайн 09.04.2020 23:59