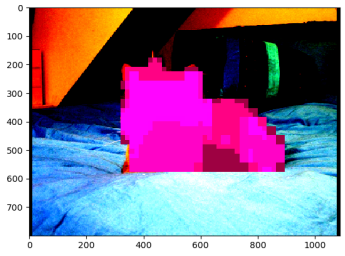
CIS 680: Project 3

Jianxiong Cai, Junfan Pan

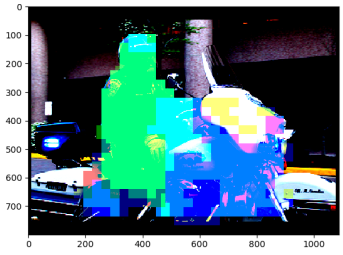
10/21/2020

1 Final instance segmentation prediction plot

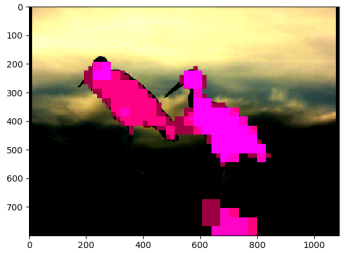


(a) SOLO instance segmentation example 1

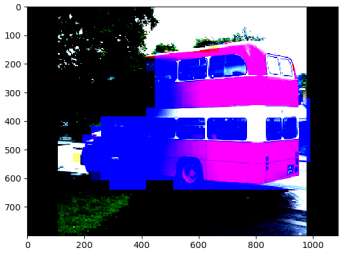
(b) SOLO instance segmentation example 2



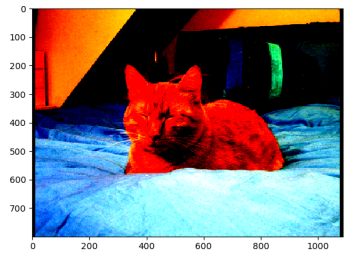
(c) SOLO instance segmentation example 3



(d) SOLO instance segmentation example 4

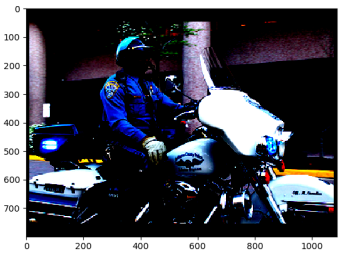


(e) SOLO instance segmentation example 5

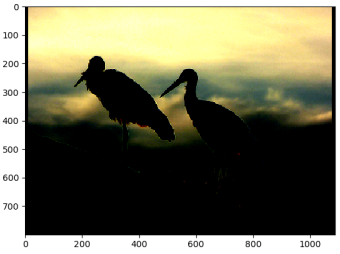


(a) Original instance example 1

(b) Original instance example 2



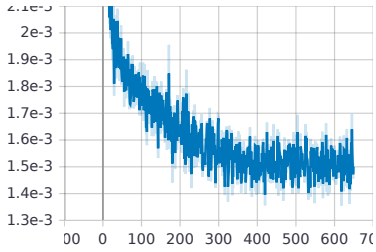
(c) Original instance example 3



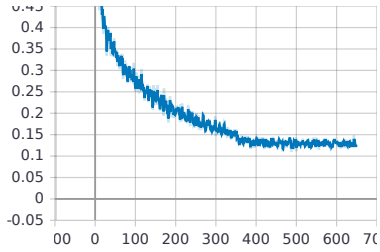
(d) Original instance example 4

(e) Original instance example 5

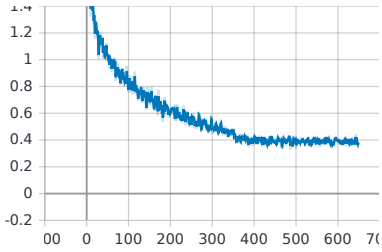
2 Loss curve w.r.t. iterations (log per 100 iterations)



(a) Cate Loss



(b) Mask Loss



(c) Total Loss

Figure 3: Training Loss. x-axis: number of iteration (iterations/100), y-axis: loss values

We apply a smoothing of 0.3 when plotting the loss curves w.r.t. iterations. We use a batch size of 2 which results in 1300 iterations in one epoch and therefore a total of 13 log iterations when drawing the loss curve in one epoch. We have trained a total of 50 epochs and by observation, all the loss curves continually decrease and converge around 35 epochs (455 iterations).

3 Parameter Setting

During training, we use the random seed 0 for numpy and random seed 1 for torch for reproducibility. We use a batch size of 2 and trained a total of 50 epochs. We use a SGD optimizer which has an initial learning rate of 0.01/8, a momentum of 0.9 and a weight decay of 0.0001. We use a learning rate scheduler with multiple step setting which multiply learning rate by 0.1 respectively at 27th and 33rd epochs. The other parameter settings are exactly the same as it is given in the SOLOHEAD self

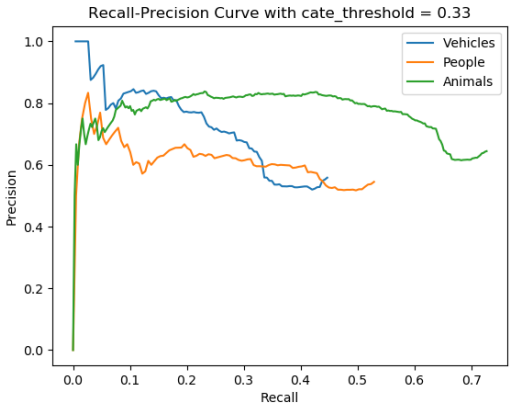
parameter settings during training which can be referenced as below.

During Infer stage, we use cate_threshold of 0.33 when calculating mAP for each category. The other parameter setting are kept the same as before.

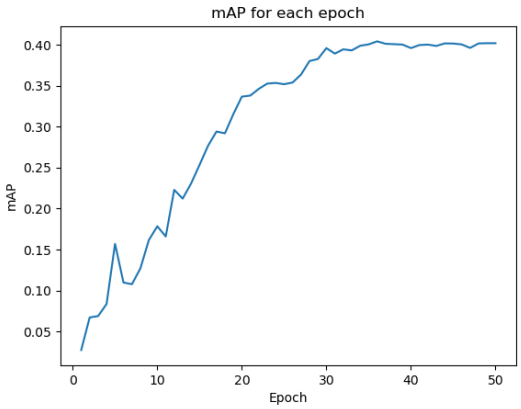
```
class SOLOHead(nn.Module):
    def __init__(self,
        num_classes,
        in_channels=256,
        seg_feat_channels=256,
        stacked_convs=7,
        strides=[8, 8, 16, 32, 32],
        scale_ranges=((1, 96), (48, 192), (96, 384), (192, 768), (384, 2048)),
        epsilon=0.2,
        num_grids=[40, 36, 24, 16, 12],
        cate_down_pos=0,
        with_deform=False,
        mask_loss_cfg=dict(weight=3),
        cate_loss_cfg=dict(gamma=2,
                           alpha=0.25,
                           weight=1),
        postprocess_cfg=dict(cate_thresh=0.2,
                             ins_thresh=0.5,
                             pre_NMS_num=50,
                             keep_instance=5,
                             IoU_thresh=0.5)):
        super(SOLOHead, self).__init__()
```

Figure 4: Parameter Setting

4 mAP and Recall-Precision Curve



(a) Recall-Precision Curve



(b) mAP for each epoch with cate_threshold = 0.33

We use the Iou_threshold = 0.5, cate_threshold = 0.33, all other threshold value keeps the same as it is in the provided code setting. The final mAP our obtain is about 0.4018 which is quite satisfying.

5 Data Augmentation

We perform a horizontal flipping augmentation when dealing with original images in the dataset. The procedure is shown below:
1. If data augmentation option is True, the images in the original dataset would have a probability of 0.5 to be horizontally flipped.
2. Bounding boxed need to be adjusted correspondingly with the horizontally flipped images by keeping the same relative x and y position of the image.