Task: Actor-Action Segmentation

Author: ZhixianHuang, Runhua Gao

Preprocessing: Turn ground truth label(224x224) into one-hot label(44x224x224).

We try 3 works: **FCN32s, UNet(changed), and SegNet.**

FCN32s+NLLLoss(Weight=[0.5,1,1,…]):the best result using training dataset on validation dataset.

Result: Acc=39.41 IoU=26.92

Though we didn't achieve good results on other network, we still had a lot of fun playing on them.

Here are the report on our experiments.

1. **FCN32s.**
   a. FCN32s pretrained on VGGNet.
   b. Loss function: NLLLoss
   c. Optimizer: SGD
   d. Learning rate: 1e-10
   e. Epochs of convergence: 50
   f. Batch_size: 16

   We find that labels in our dataset actually are not balanced and there are too many background points, so we set weight for NLLLoss, trying to withdraw the loss from the background points.

   Weight1 = [1,1,1,…](1x44 tensor) Result: Acc=35.8 IoU=25.73

   Weight2 = [0.5,1,1,…](1x44 tensor) Result: Acc=39.41 IoU=26.92

   Weight3 = [0.1,1,1,…](1x44 tensor) Result: Acc=42.64 IoU=23.72

   Weight4 = [0.01,1,1,…](1x44 tensor) Result: Acc=40.2 IoU=17.42

2. **UNet.**
   a. Pretrained on VGGNet
   b. Loss function: Dice_loss and NLLLoss(Weight = [0.1,1,1,…])
   c. Optimizer: Adam
   d. Learning rate: 1e-10

e. Epochs of converge: 80

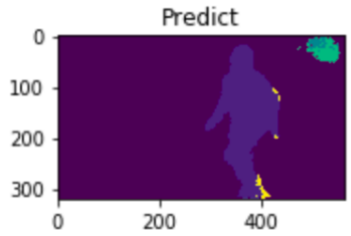
f. Batch_size: 16

After a lot of experiments, we find something interesting in UNet.

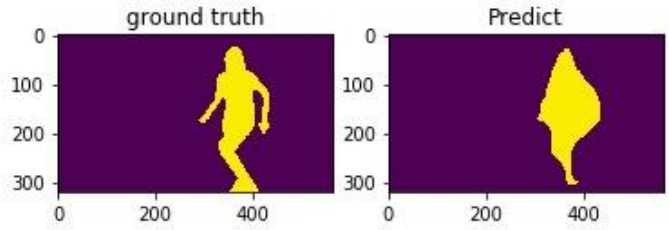
A. We use two loss function and do a lot of experiments. Finally we find Dice_loss suffers from unbalanced dataset and cannot lead to reasonable results, though it is start-of-art loss in some segmentation mission.

B. We first implement UNet architecture in original paper, but this network's output can always be all zeros. After checking every layers' gradient when training, we find gradients vanishing in 4th up layer. To overcome the gradient vanishing, we get rid of the 4th up layer's activation function and output has normal result.

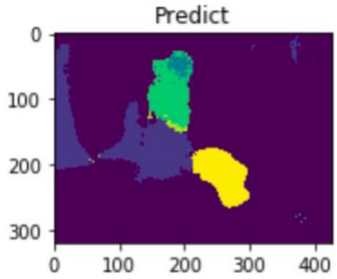
C. Though UNet cannot reach a very good result compared to FCN and sometimes cannot make the right prediction of labels, but it always has better segmentation edge. Here are some examples.



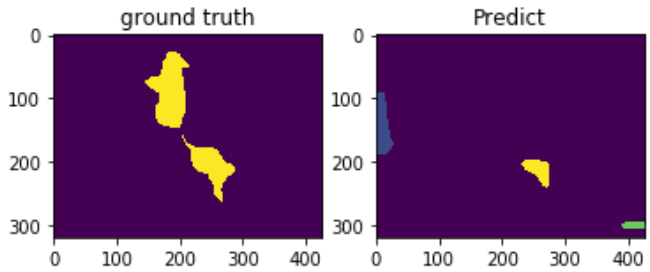
(UNet prediction)



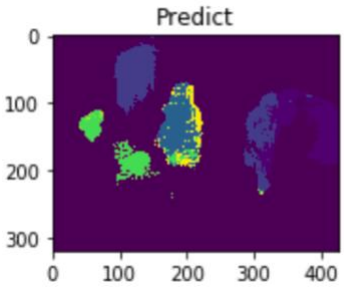
(ground truth and FCN32s prediction)



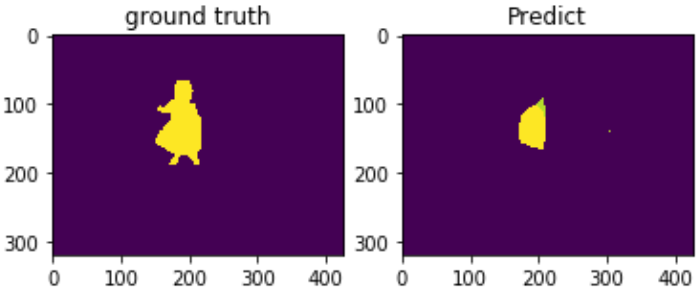
(UNet prediction)



(ground truth and FCN32s prediction)



(UNet prediction)



(ground truth and FCN32s prediction)

Result of UNet: Acc=41.89 IoU= 9.18

We can see from these examples the reason why UNet has high accuracy but low IoU is UNet always makes many wrong prediction on where is actually background.

3. SegNet

a. Pretrained on VGGNet
b. Loss function: NLLLoss
c. Optimizer: Adam
d. Learning rate: 1e-10
e. Epochs of converge: 50
f. Batch_size: 16

SegNet's performance is close to UNet, but it converges faster than UNet, which is reasonable since SegNet is smaller than UNet.

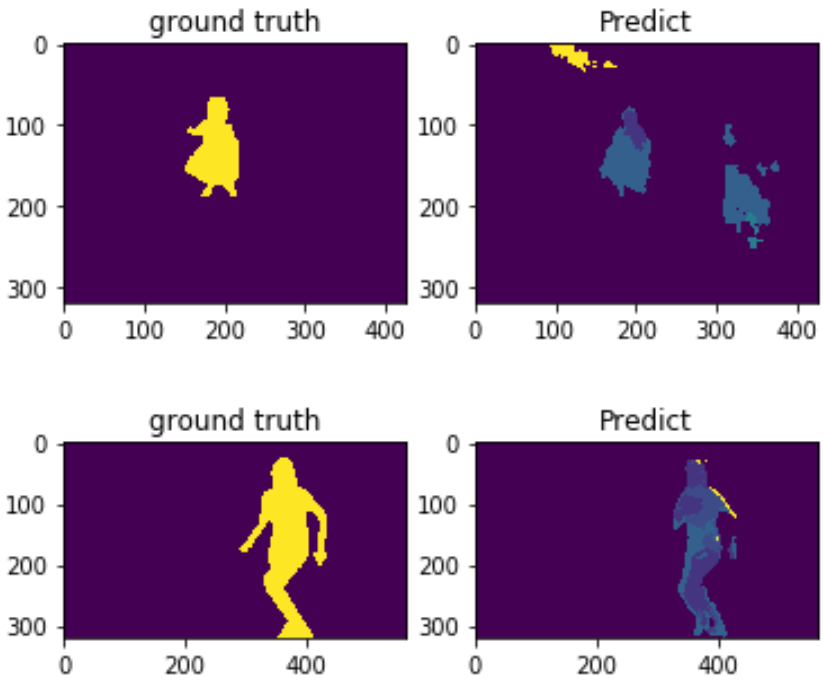
4. Ensemble SegNet and FCN32s

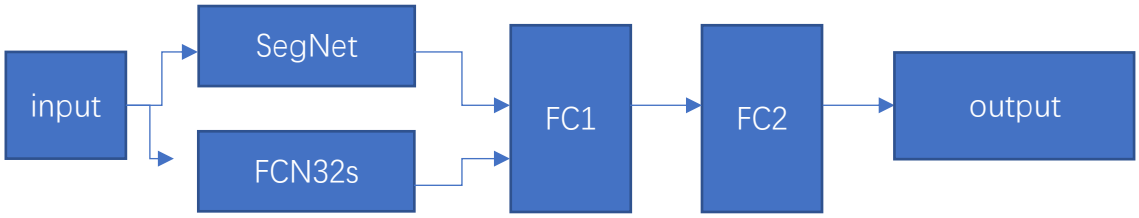
To get good segmentation edge and right prediction label, we try to ensemble SegNet and FCN32s, which are all pretrained (on train dataset before). For this part, we only update parameters of FC1 and FC2.

a. Pretrained on train dataset
b. Loss function: NLLLoss
c. Optimizer: Adam
d. Learning rate: 1e-3
e. Epochs : 30(not converge, but we have no time to do more experiments)
f. Batch_size: 16

Result : Acc=18.89 IoU= 8.96

However, we fail on this experiment, the ensemble hasn't converged.




input
SegNet
FCN32s
FC1
FC2
output