

# SRTP初步训练总结汇报

---

- 训练与测试数据指标及指标可视化
- 训练与测试集的图片对比
- 不足之处的思考

## 1. 训练与测试数据指标及指标可视化

---

### 训练与测试数据指标

Epoch	训练集的平均loss	训练集的验证误差	测试集的平均误差
1	0.8848	4.012%	
5	0.6941	3.075%	
10	0.6196	2.881%	
15	0.5565	2.788%	
20	0.5286	2.690%	2.662%

#### 基本说明：

- 损失函数：计算不同的epoch下训练集和测试集的平均loss使用交叉熵损失函数。因为unet使用了dense prediction的方法。所以它对于图像每一个像素，都使用了softmax进行预测(pixel-wise softmax)。预测输出的结果是一个三维向量(width , height , class\_nums)代表图片中不同高度和宽度像素的分类得分。而这里的误差函数就是每个像素在不同分类下预测得分情况和原标签图片每个像素分类之间交叉熵。
- 训练集的验证误差：训练集在不同epoch下的误差指标使用了源码自带的误差计算函数进行计算。对于每一个像素，我们取它得分最高的标签类别作为自己的分类预测，再和标签图片进行比较，以计算误差。
- 测试集的验证误差：我们调用源码自带的误差计算函数对测试集进行误差指标的计算，原理与训练集的误差指标计算相同。

## 指标可视化

1. 训练集在不同epoch下的平均loss



2. 训练集在不同epoch下的错误率指标

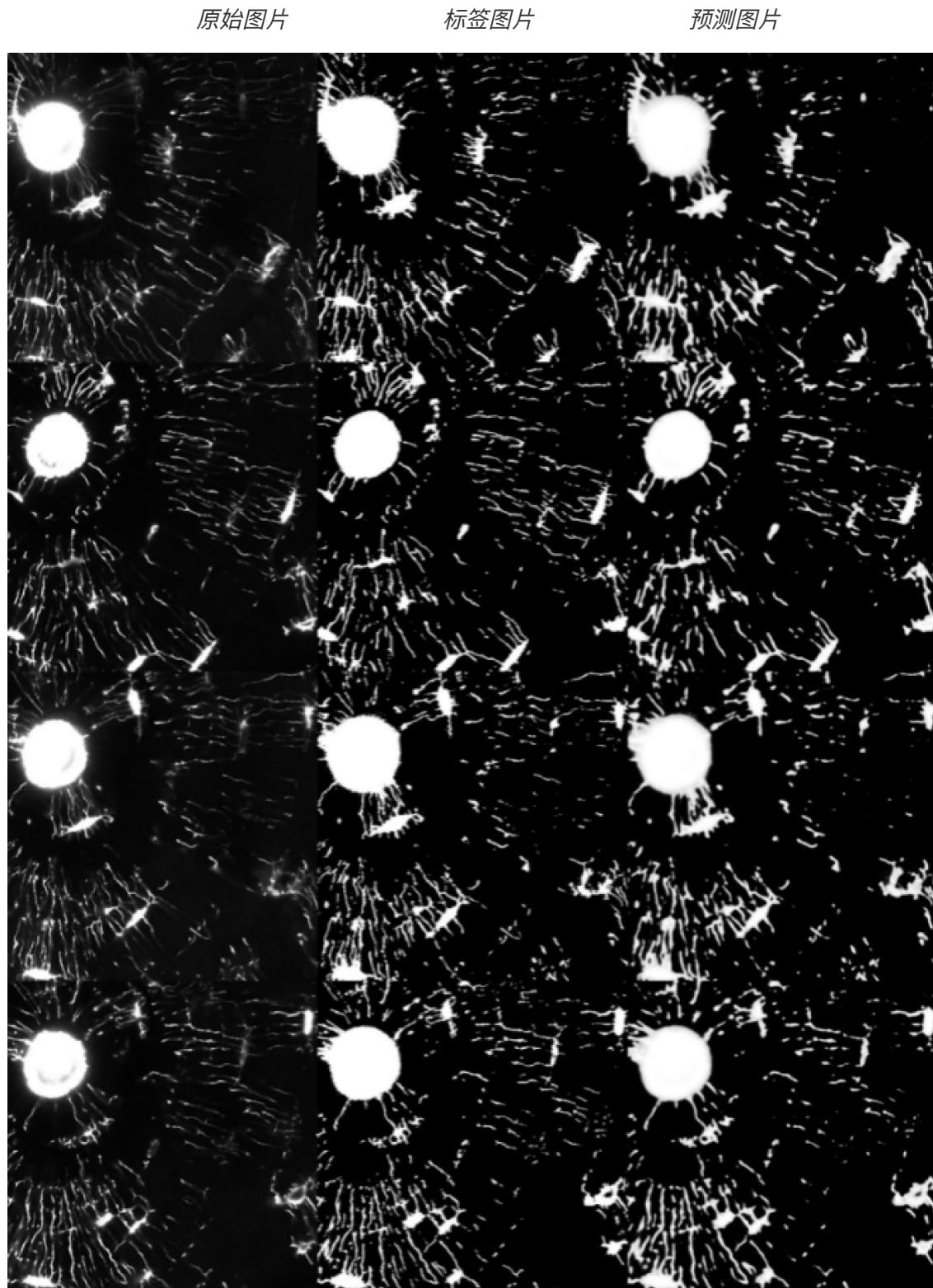


关于指标可视化图片的说明：

- 由于开始训练的第一个epoch的loss值与误差值相比较后面过大，故删除第一个epoch下的两个计算值再来作图。
- 训练中途有中断，恢复训练后相邻的两个epoch下会有十分相近的loss值和误差值，我们认为没有什么参考意义，所以删除了其中一个epoch下的这两个计算值再进行画图.

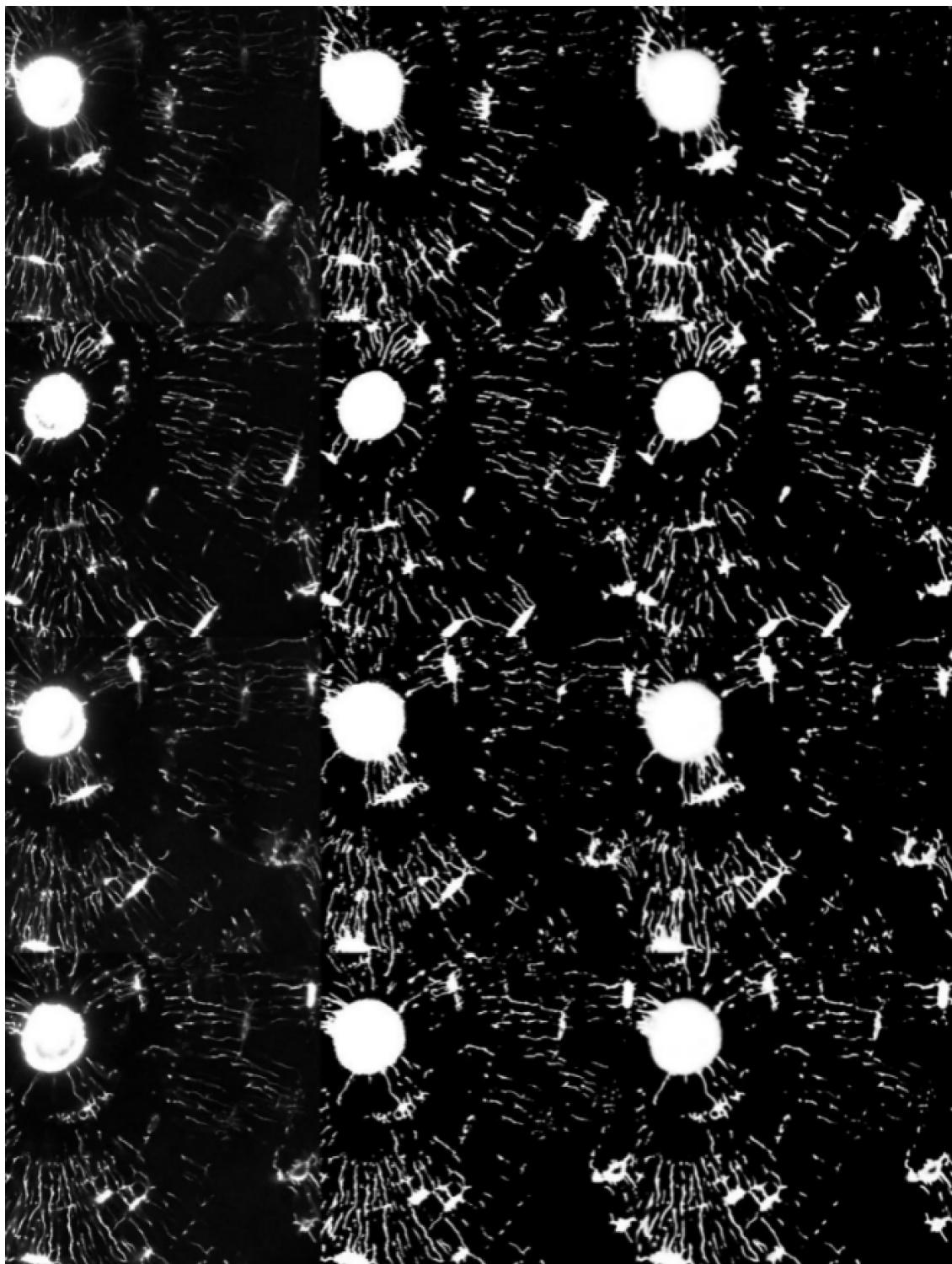
## 2.训练与测试集预测与标签图片的对比

1. 训练集预测与标签图片的对比 (Verification Batchsize = 4)



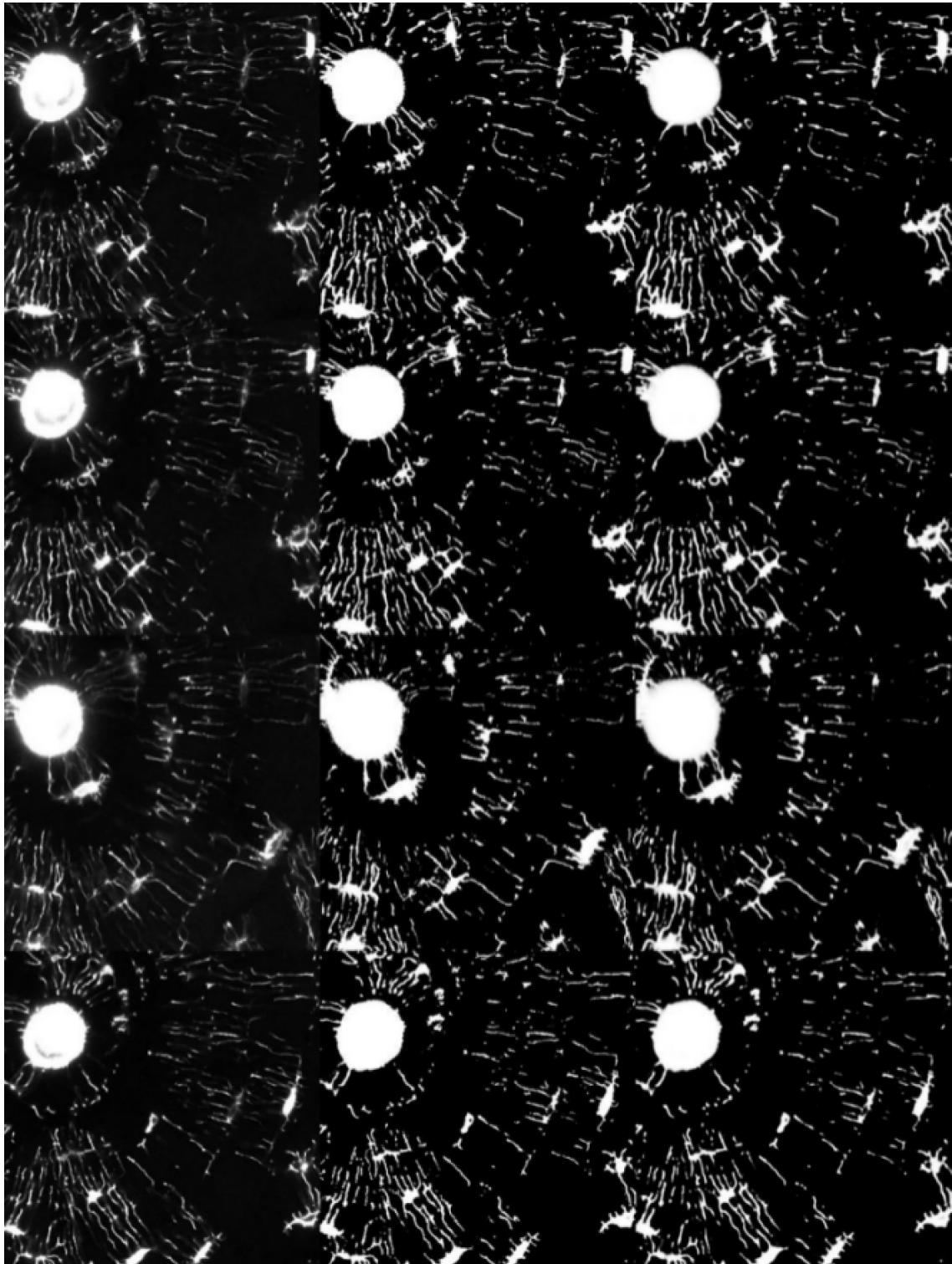
(epoch1)

---



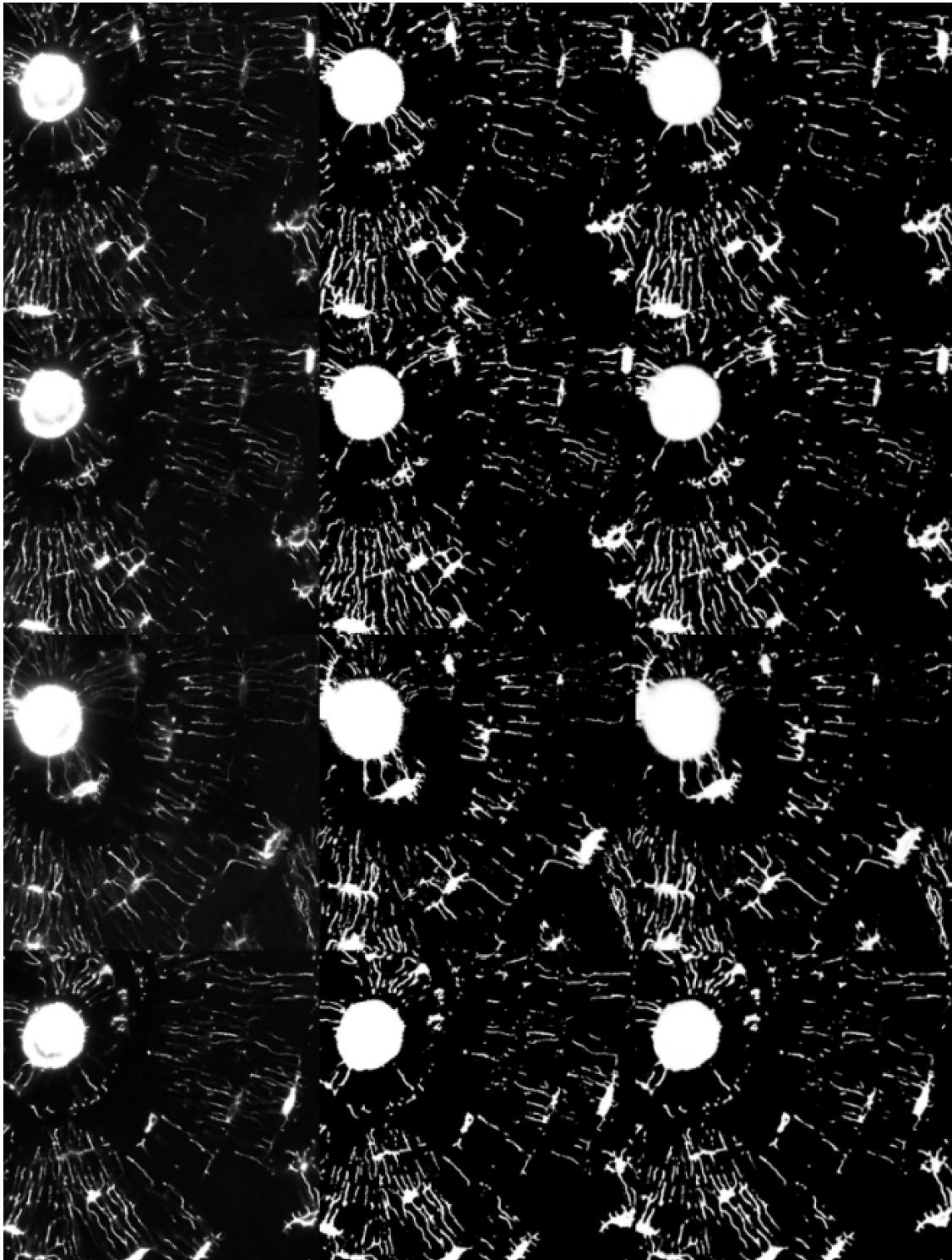
(epoch5)

---



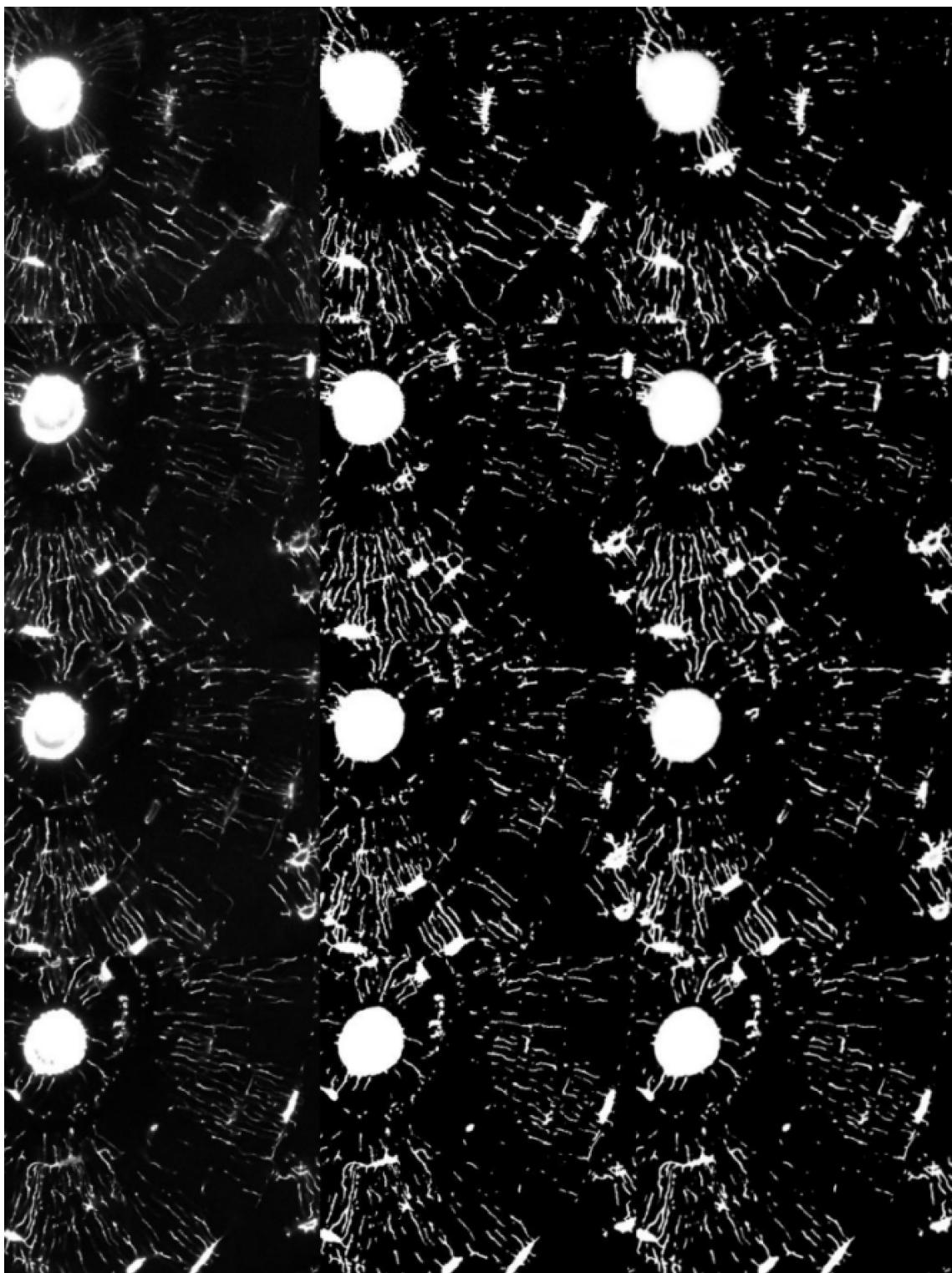
(epoch 10)

---



(epoch 15)

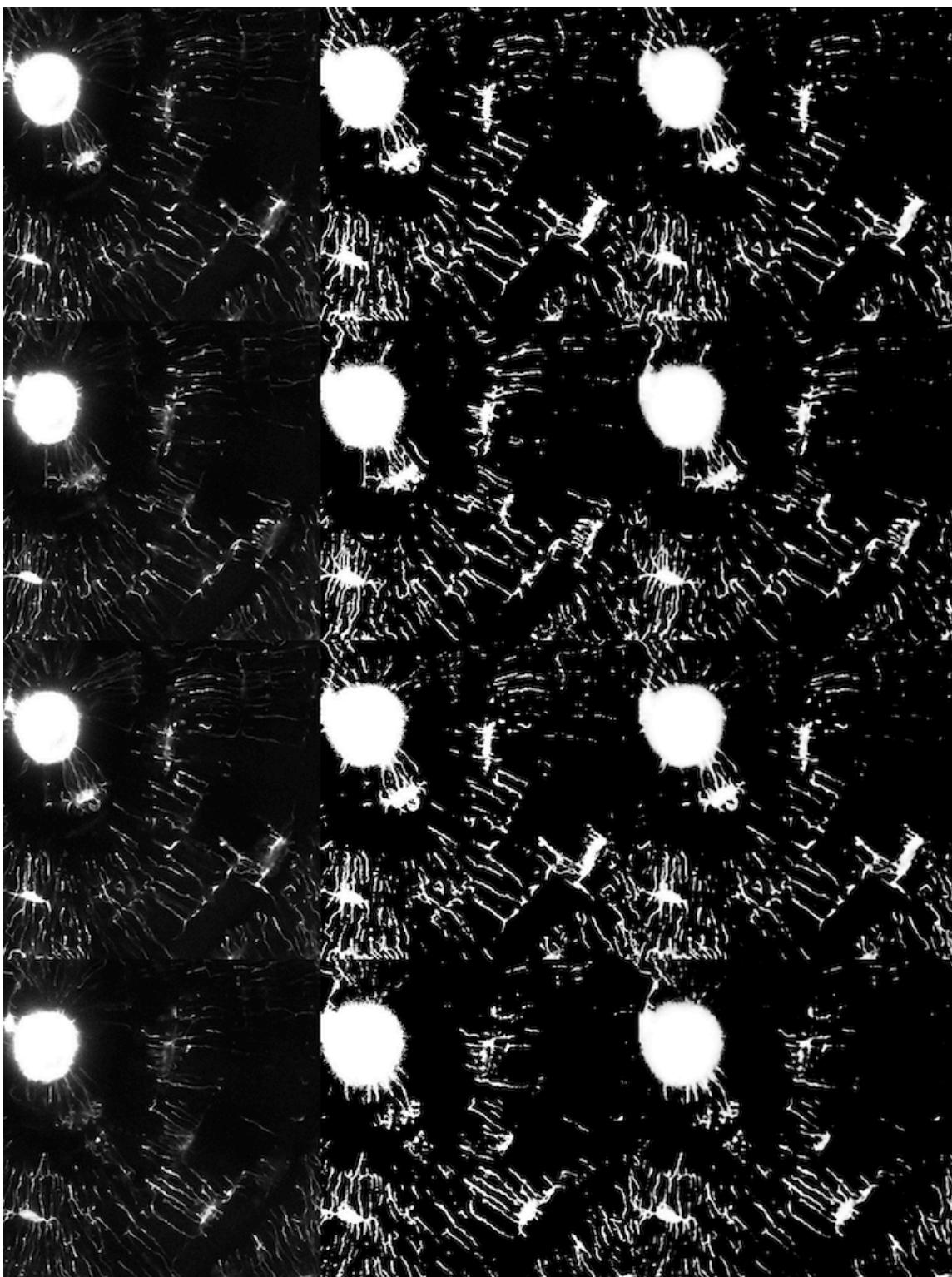
---

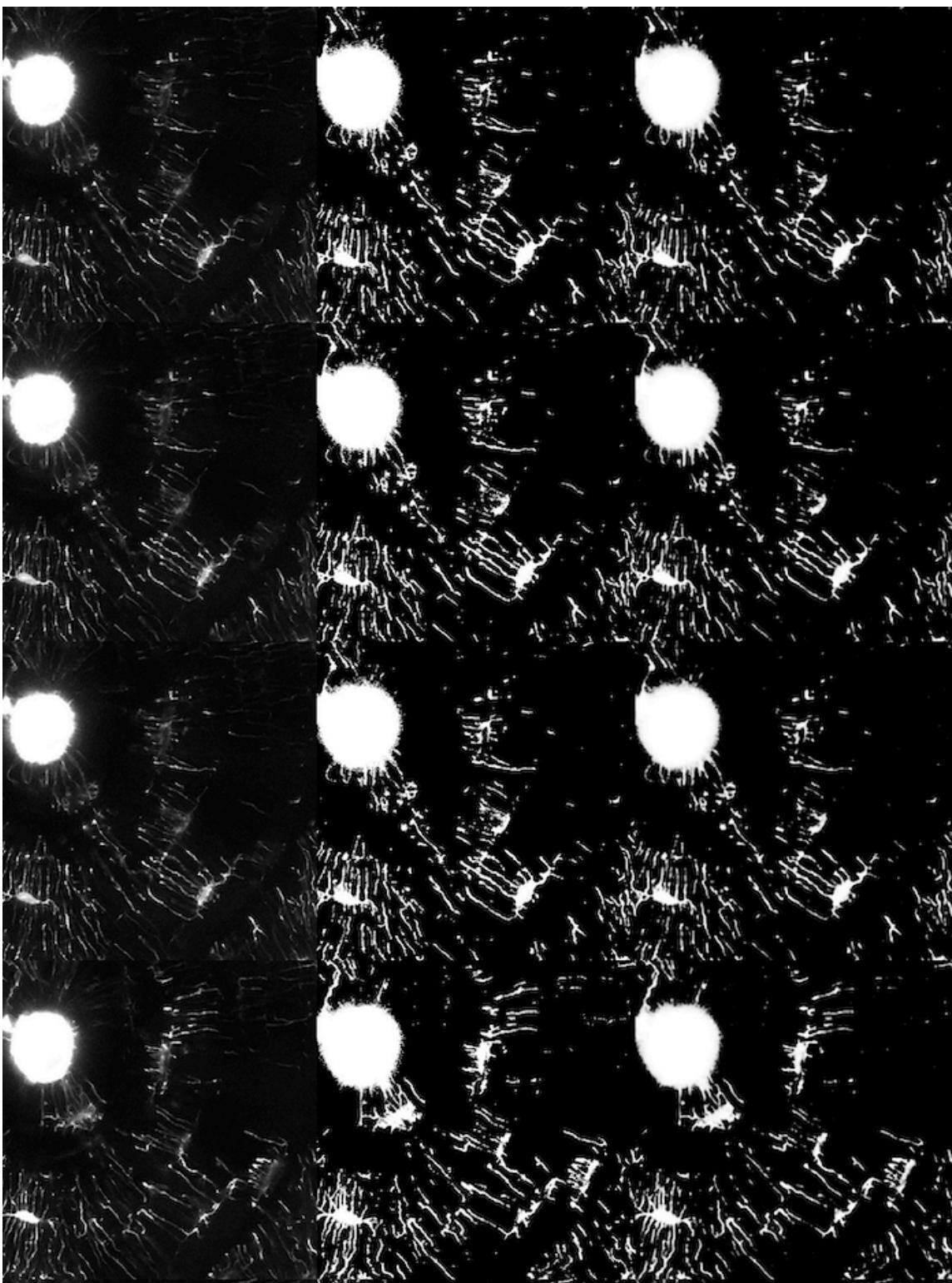


(epoch20)

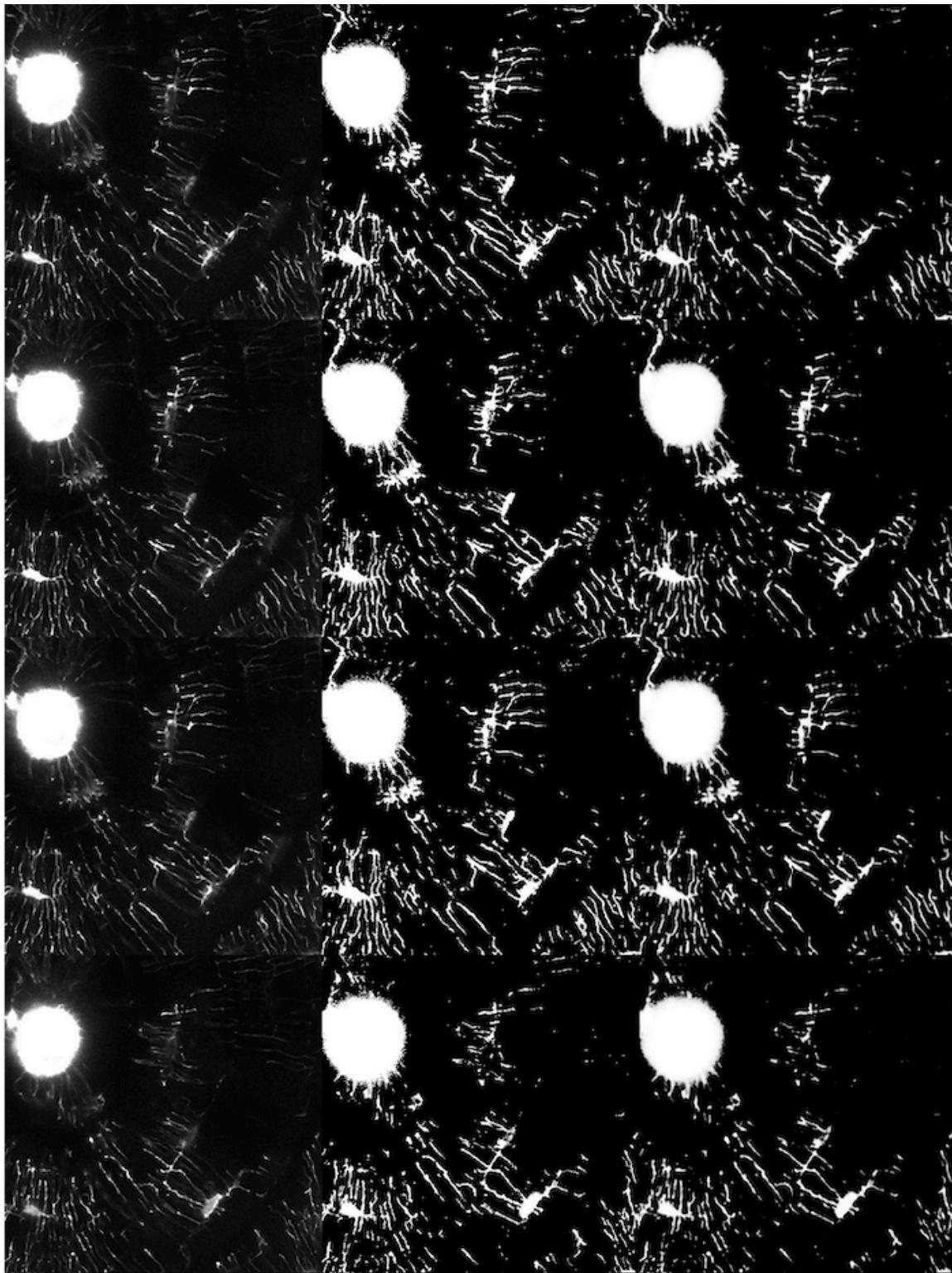
---

2. 测试集预测与标签图片的对比 (Verification Batchsize = 4)





---



### 3. 不足之处的思考

1. 从测试集和训练集的损失函数图像可以看出来，二者的loss都随着epoch的增加而有明显的下降。但是都没有收敛。经过学长的提醒，我们意识到可能继续训练下去能够得到更好的结果。
2. 关于测试集的误差计算，我们之前尝试设置合适的阈值将测试集的预测图片导出为二值图像。但是在计算时，训练集的平均误差会随着设置阈值的变大而减小。当阈值取得255时，平均误差会达到最小值 7.979%。然而，将此时的二值矩阵代表的图像显示出来，明显可以看出来它和标签图像不太吻合。反而经过较小阈值处理的图像能够较好地对应标签图像。这是我们没有弄懂的一个问题

例如：

*label picture*

*prediction 阈值 = 255(误差最小)*

*prediction 阈值 = 127*

