

NestedUNet のテスト結果を出力

これまで訓練した `Nested UNet` が入力画像に対してセグメンテーションをできるかを確認するために、セグメンテーションした結果を画像として保存する関数を作成する。

方法は簡単で、`Validation` で使用したコードを流用しつつ、結果を `OpenCV` で保存するようにした。

作成したコードの一部を以下に示す。

```
avg_meter = AverageMeter()

for c in range(config['num_classes']):
    os.makedirs(os.path.join(results_dir_path, config['name'], str(c)), exist_ok=True)
    with torch.no_grad():
        for input, target, meta in tqdm(val_loader, total=len(val_loader)):
            input = input.cuda()
            target = target.cuda()

            # compute output
            if config['deep_supervision']:
                output = model(input)[-1]
            else:
                output = model(input)

            iou = iou_score(output, target)
            avg_meter.update(iou, input.size(0))
            output = torch.sigmoid(output).cpu().numpy()

            for i in range(len(output)):
                for c in range(config['num_classes']):
                    cv2.imwrite(os.path.join(results_dir_path, config['name'], str(c), meta['img_id'][i] + '.jpg'), (output[i, c] * 255).astype('uint8'))

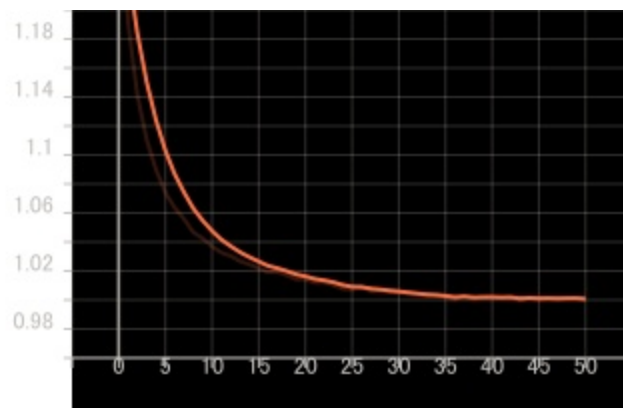
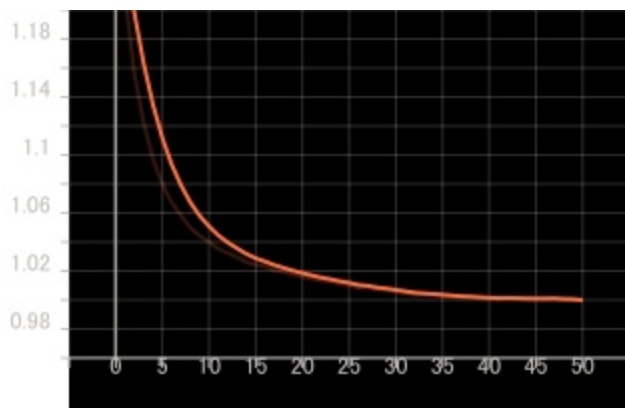
print('IoU: %.4f' % avg_meter.avg)

torch.cuda.empty_cache()
```

50 Epoch 回して得られた結果

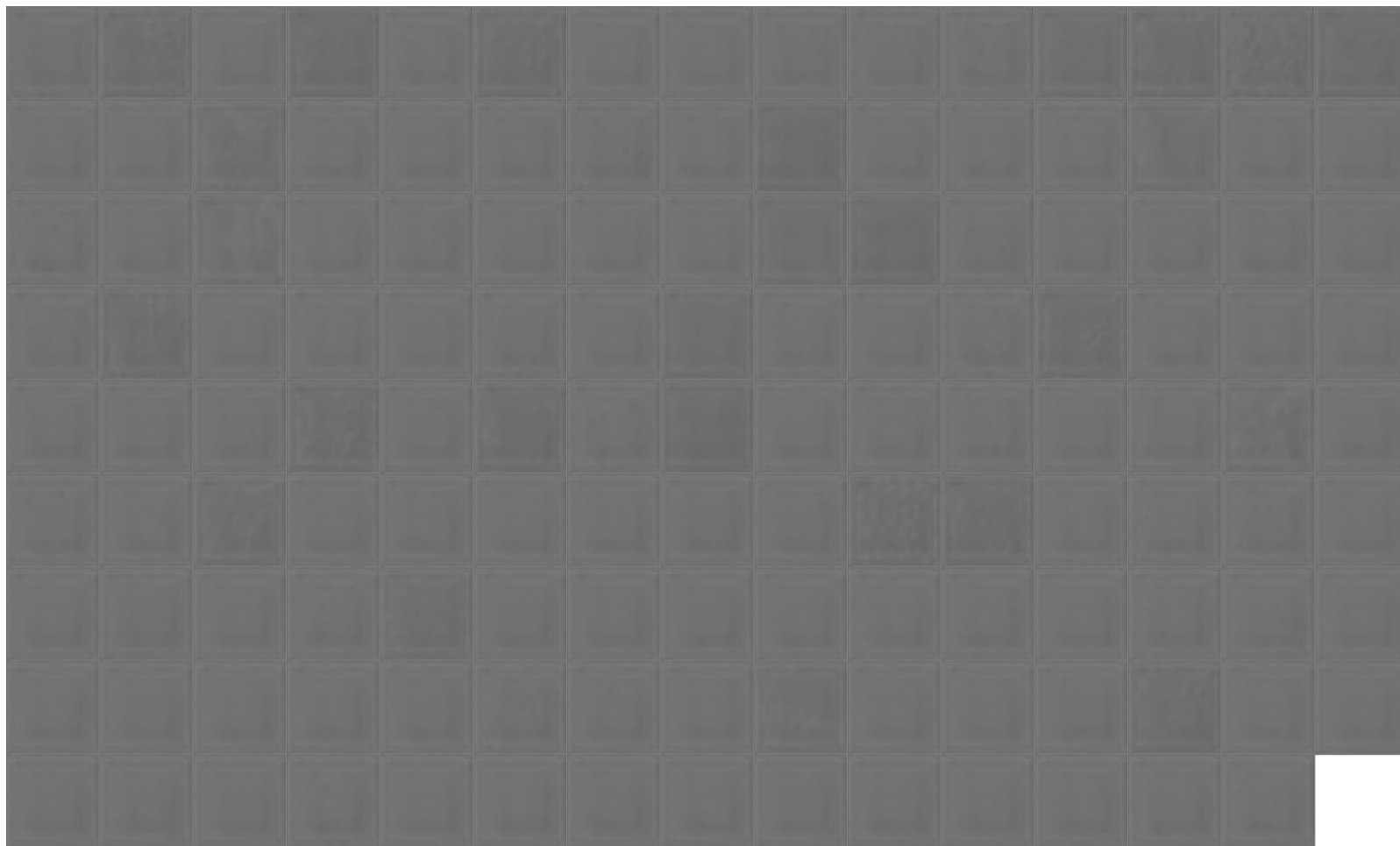
次に、50 Epoch 訓練した NestedUNet の訓練結果とテスト結果を、それぞれ以下に示す。

訓練誤差と検証誤差の減衰傾向



このグラフより、50 Epoch 程度では誤差が0.98程度までしか減少していないことが確認できる。

テスト画像に対してセグメンテーションを行った結果



画像ごとに微少な変化があることは確認できるが、何かしらを生成途中という印象を受ける。これは、誤差がまだ十分に減少していないことが原因であると考えられる。