



PROGRESS REVIEW

HUBMAP + HPA - HACKING THE HUMAN BODY

GROUP B

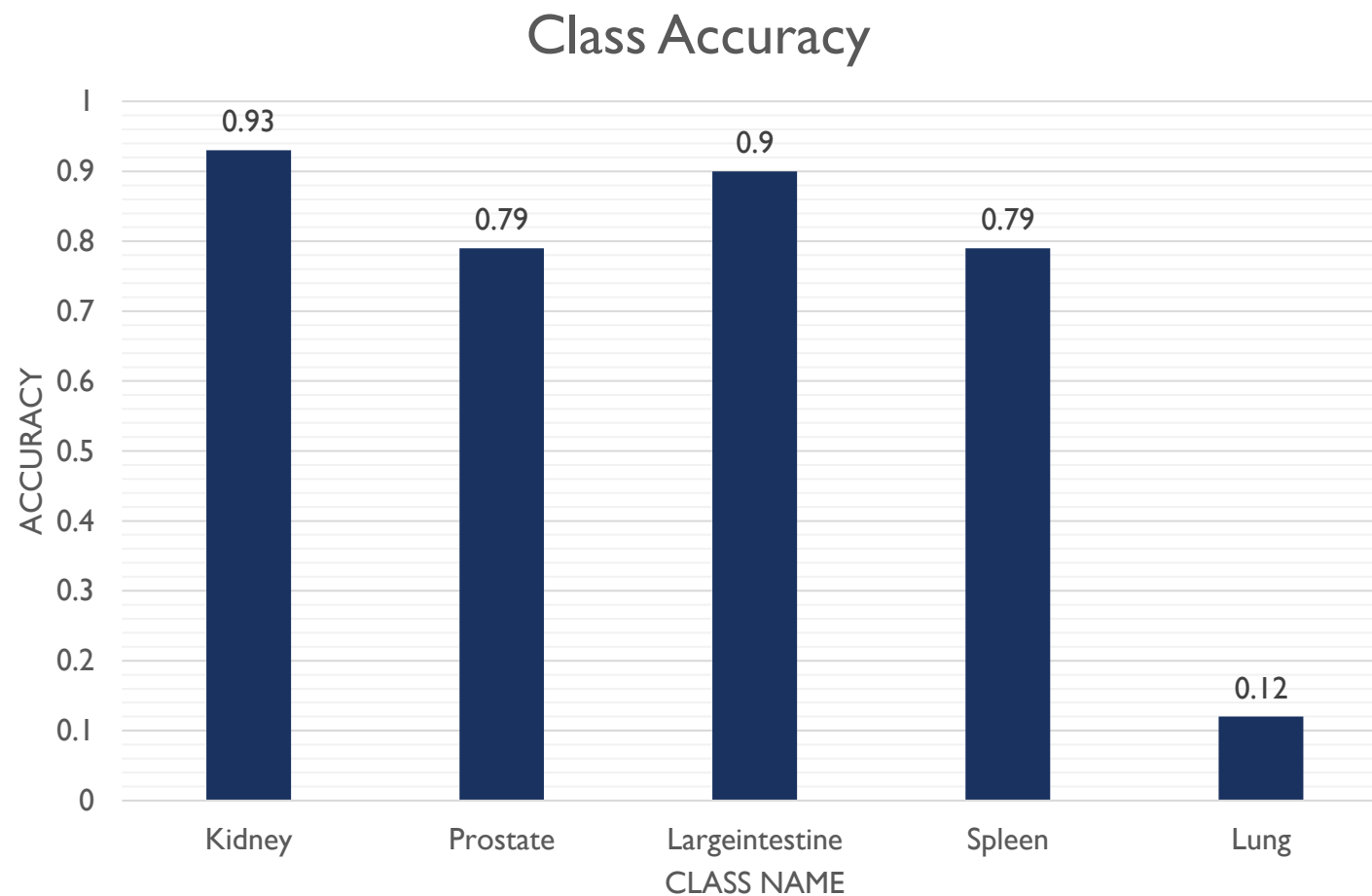


CONTENT

- Organs' accuracy
- Simple Swin-Transformer
- More reasonable Swin-Transformer

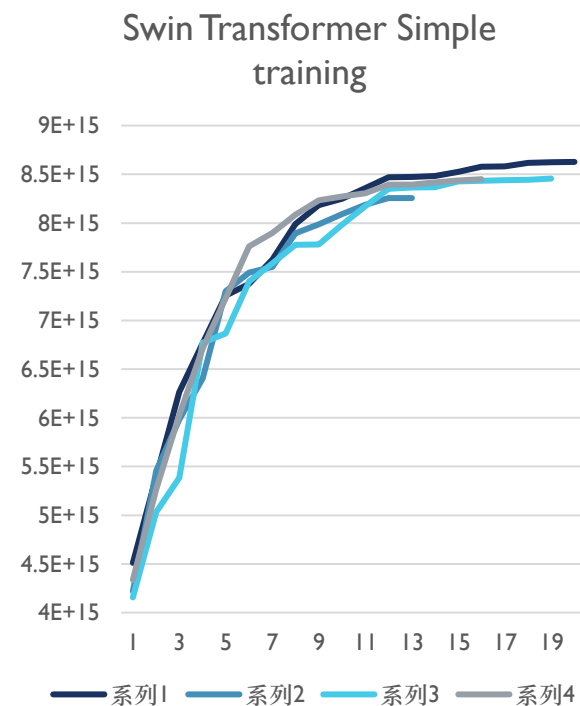
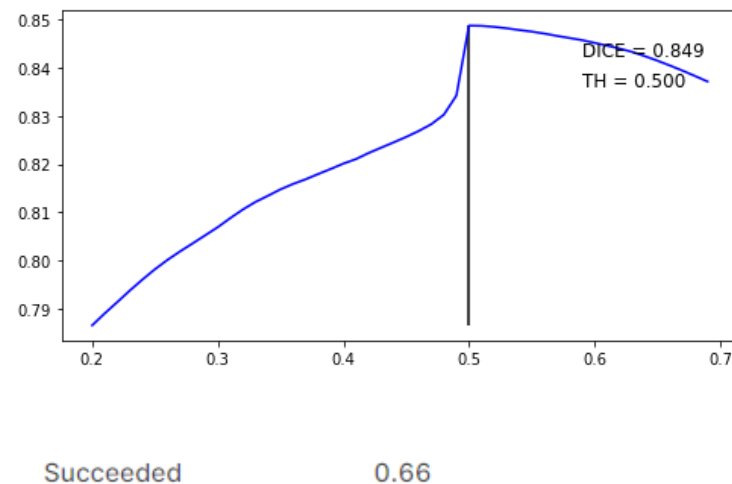
DURING REGULAR TRAINING PROCESS

- During Regular training process, we have find that accuracy varies for different organ samples:
- Thus, organ embeddings are essential to make difference for network to consider such variance.



SIMPLE SWIN-TRANSFORMER FAILED

- After we come out the inference notebooks for swin-transformer like models, we can finally get some feedbacks on it.
- If we simply implement swin-transformer to a tiled and balanced dataset without multi-modal implementation, it seems no better than FPN models which can make use of expansion tricks. Even though the loss and threshold curve looks quite promising.
- i.e. some overfitting!



RELATED READINGS

- [A review: Deep learning for medical image segmentation using multi-modality fusion – ScienceDirect](#)
- [Deep Learning-Based Image Segmentation on Multimodal Medical Imaging | IEEE Journals & Magazine | IEEE Xplore](#)
- [A Review of Multimodal Medical Image Fusion Techniques \(hindawi.com\)](#)

WE HAVE TO CONSIDER MORE INTRICATE WAYS.

- Instead of using 256*256 tiles in training, we decided to use resized whole image in training to make use of slide window mechanisms which should be of the size divisible to 256, in our implementations: 768*768.
- Also, we embedded our inputs by loading extra information of organ class as multi-modal mechanism of our model.

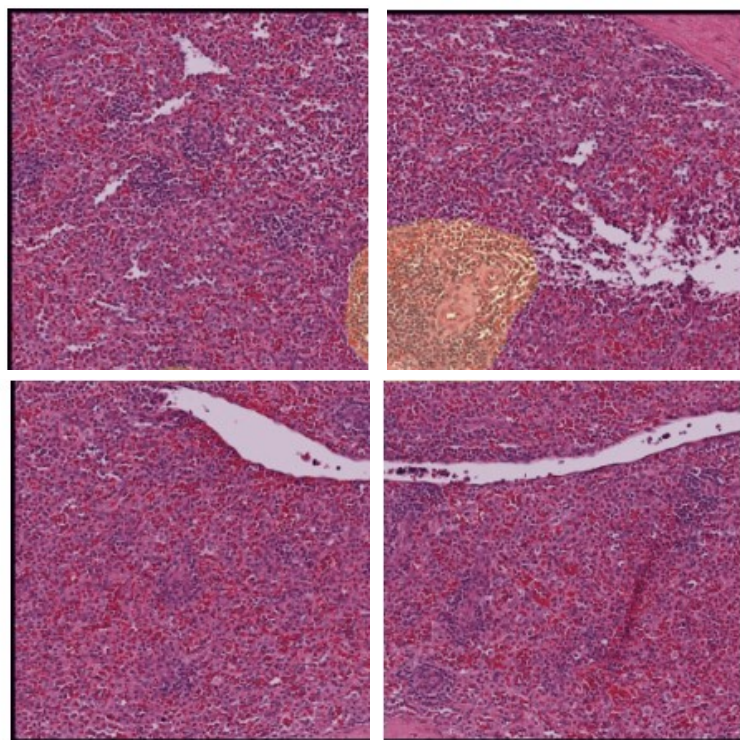
```
image_size = 768
```

```
image = cv2.resize(image, dsize=(image_size, image_size))  
mask = cv2.resize(mask, dsize=(image_size, image_size))
```

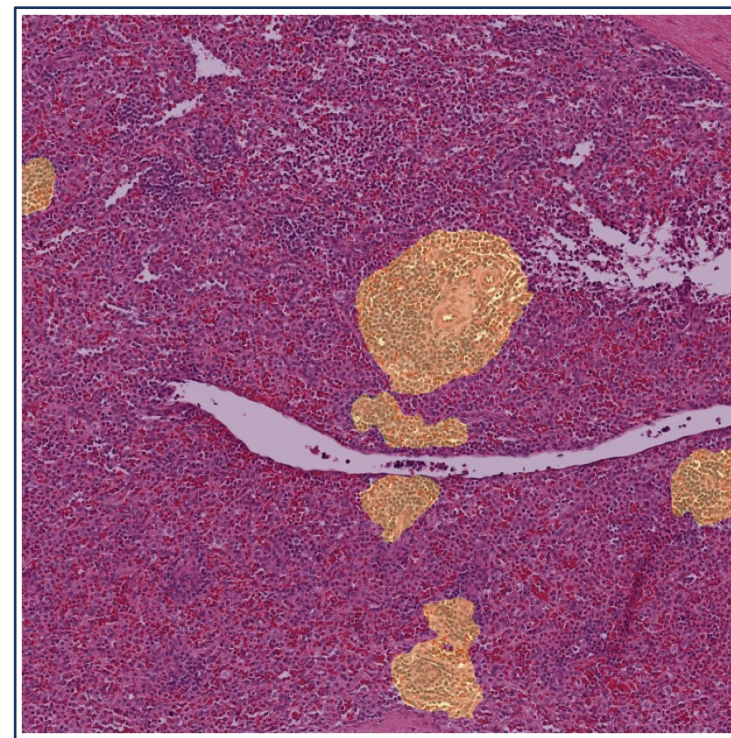
```
class HubmapDataset(Dataset):  
    def __init__(self, df, augment=None):  
  
        self.df = df  
        self.augment = augment  
        self.length = len(self.df)  
        ids = pd.read_csv(LABELS).id.astype(str).values  
        self.fnames = [fname for fname in os.listdir(TRAIN) if fname.split('_')[0] in ids]  
        # print(self.fnames)  
        self.organ_to_label = {'kidney' : 0,  
                                'prostate' : 1,  
                                'largeintestine' : 2,  
                                'spleen' : 3,  
                                'lung' : 4}
```

WE HAVE GOT QUITE GOOD RESULTS ON IT.

- By visualization we can see that the effect is much better than efficientnetb7 baseline we have made before.



Efficientb7-0.7



Swin-Transformer

However, there are some submission issues that we have to debug our inference notebook to solve it.

FUTURE PLANS

- Find external data for some organs and try to produce some annotation.
- Debugging on what we have for now.



Thank you!