COMS4995: Applied Deep Learning

Final Project Presentation

Hanzhou Gu(hg2498)

Introduction: DL in medical imaging

Goals:

- Reduce the misdiagnosis rate for common diagnosis task
- Mostly assist pathologist rather than fully automatic
- Reduce wordload and error
- Reduce subjectivity
- deal with small tumor

Challenges:

- Mostly not techinical: collect data, data preprocessing, medical domain knowledge, deploy real-time model, morals about private information

Project desciption

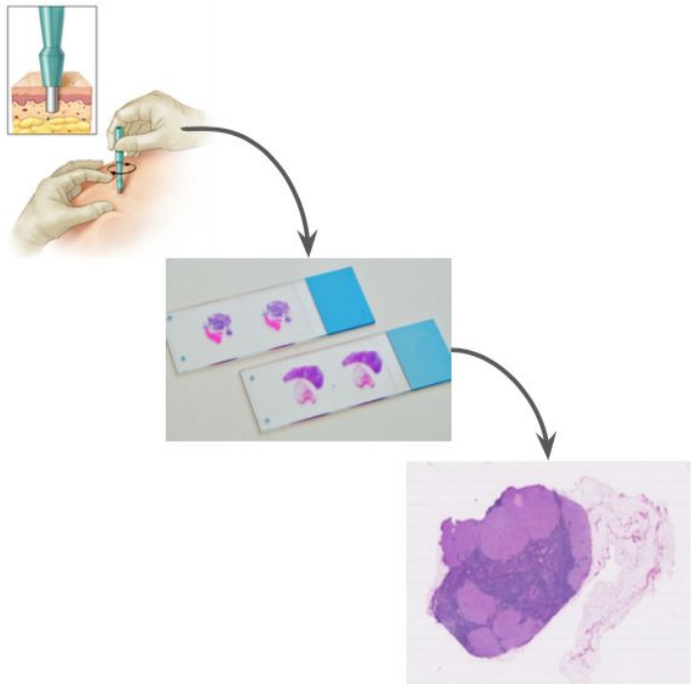
Goal: Detect cancer in gigapixel pathology stained whole-slide images of lymph node sections

Data:

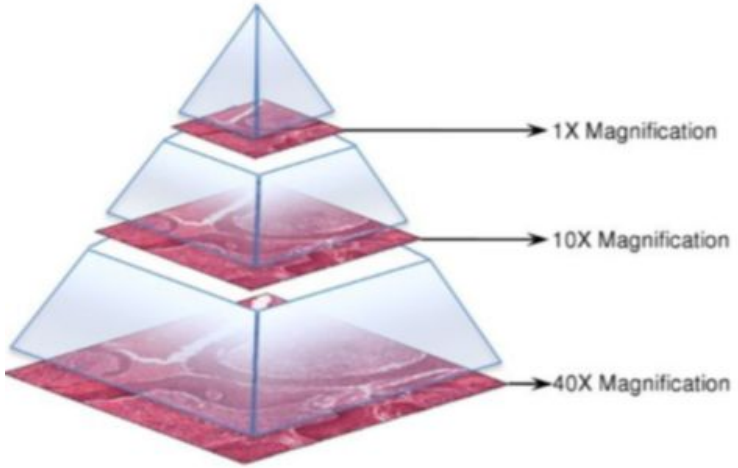
Whole-slide images of lymph node sections. Part of the dataset in the CAMELYON16 challenge which are Images from two medical center

Pipline:

Sample, process, learning and metrics



Data preprocessing



input to model: 299*299*3 for 2 level

label: whether center 128*128 contain cancer cell

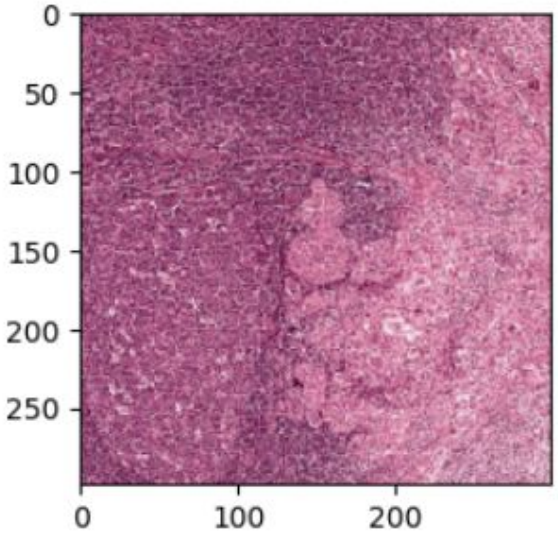
training and val data: random selected sliding window extracted across slide

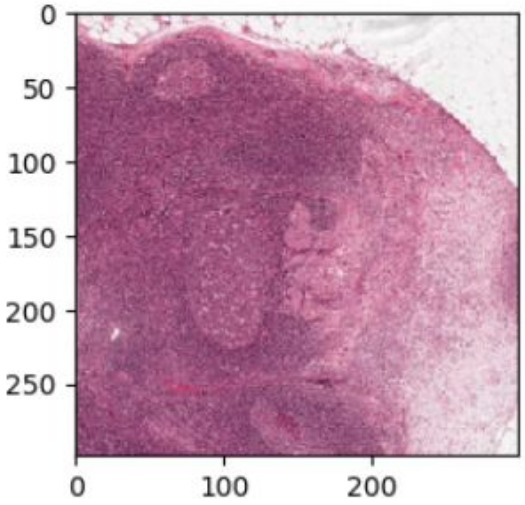
patch_size = 299 central_size = 128 stride = 100

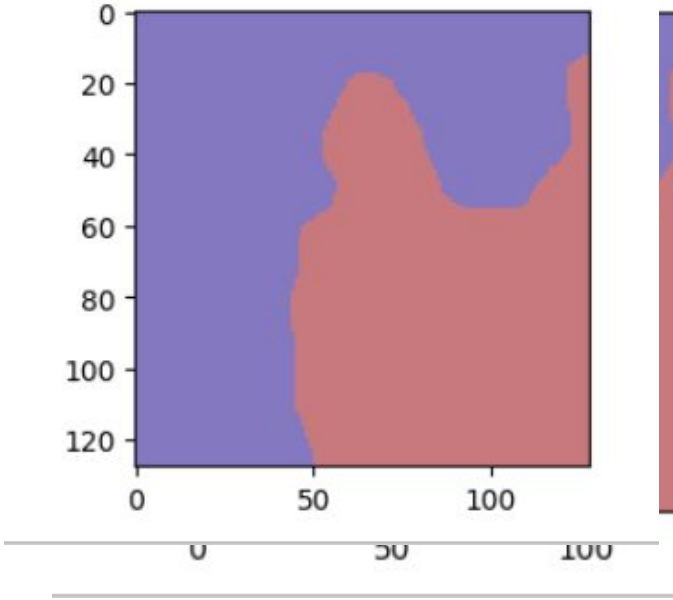
testing data: sequential sliding window extracted from slide

patch_size = 299 central_size = 128 stride = 128

64X ,32X, tumor mask







Reduce computation

--Threshold for gray value of image

removed background patches (gray value too high), only focus on tissue cell

containing tissue more than 0.7

--random select patch from slide for training set

but not all slides by sliding window

data downsampling for normal patch

Model

Transfer learning

InceptionV3 and fc layer

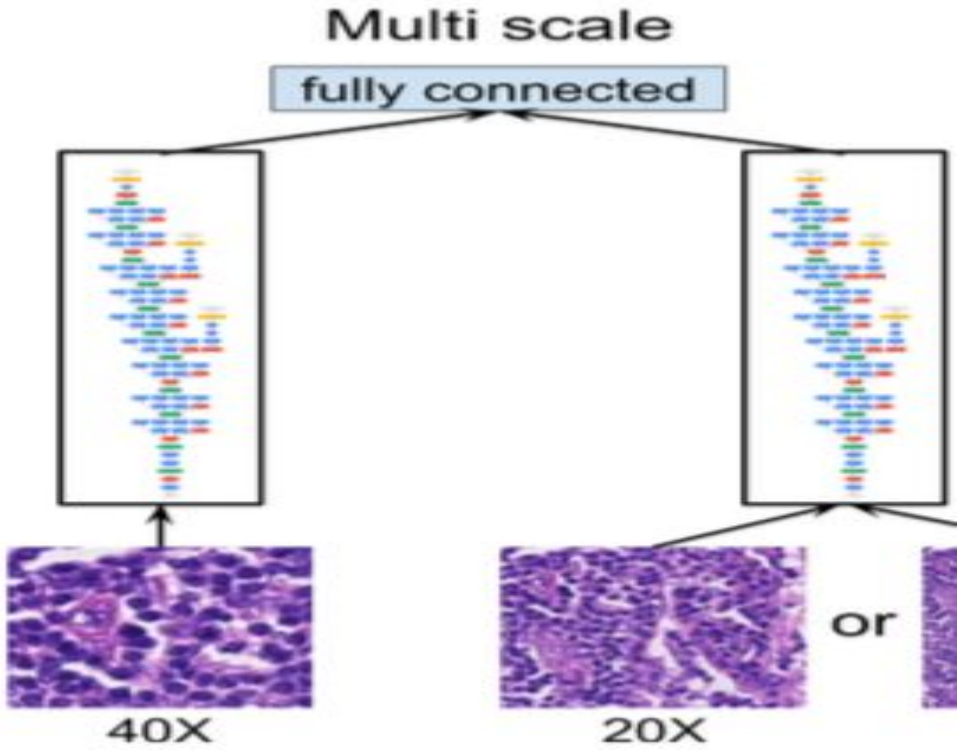
two scale

64X and 32X

validation split 0.1

sigmoid activation

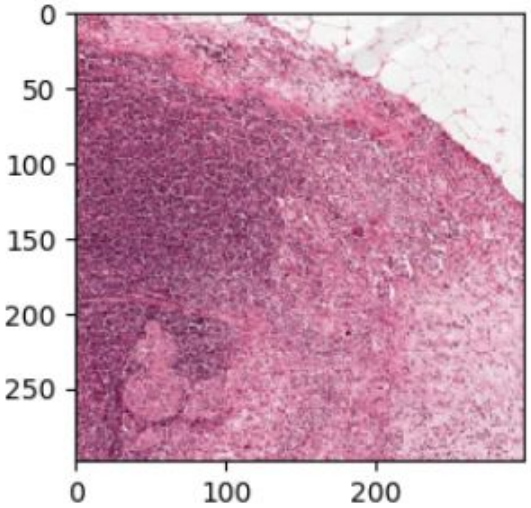
rms optimizor



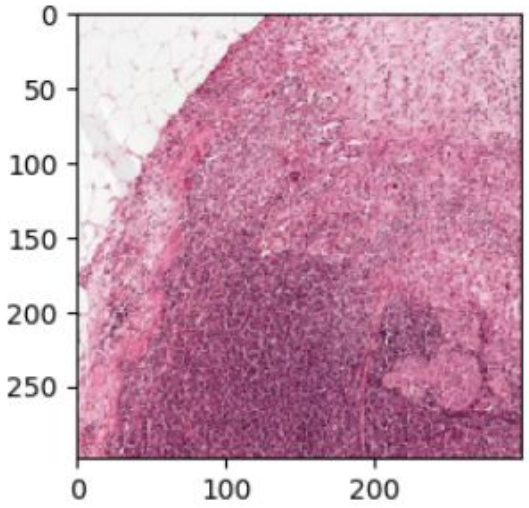
Data augmentation

only for training set

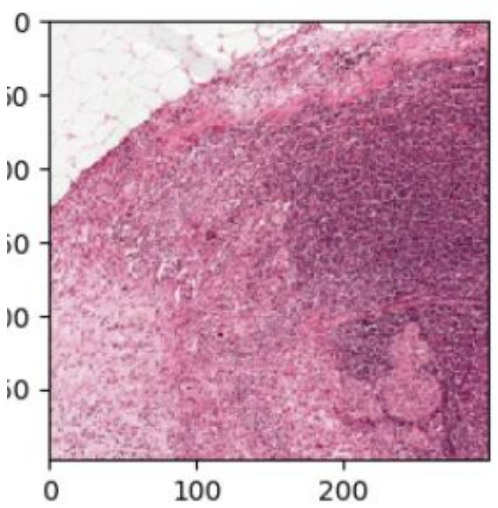
origin



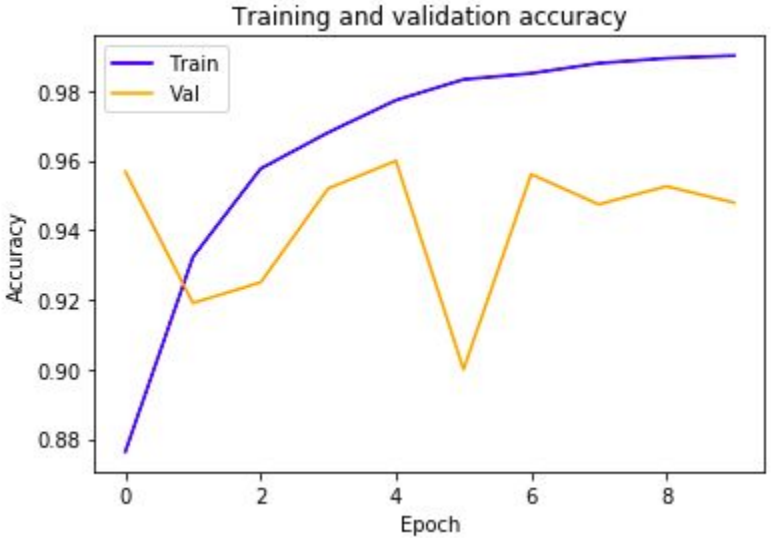
rotate 90 degree

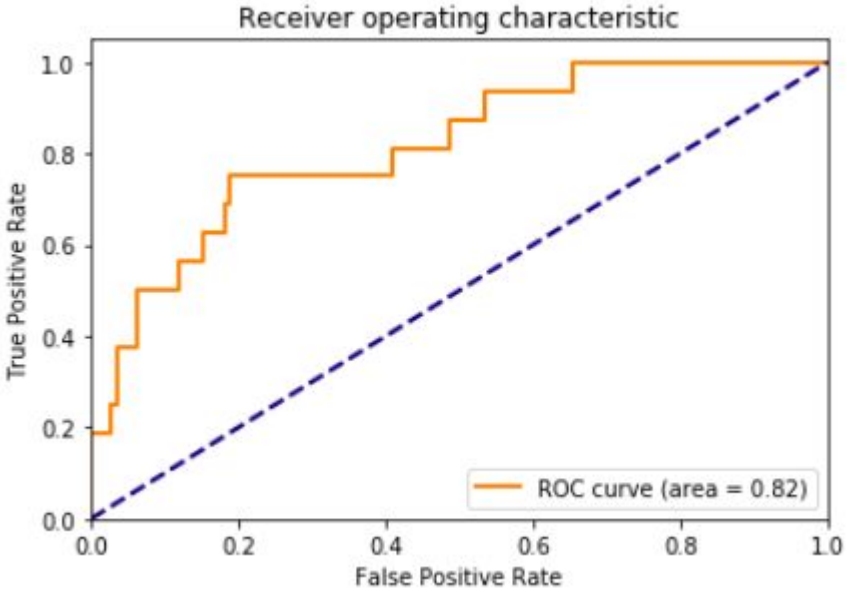


left-right flip



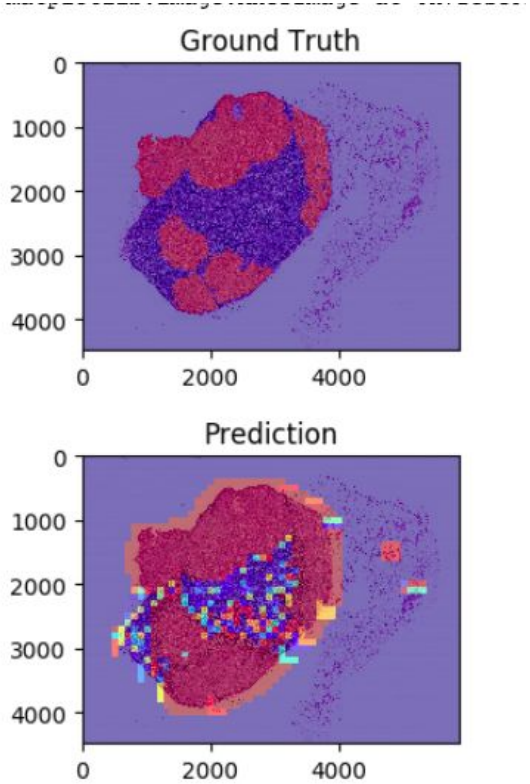
Result

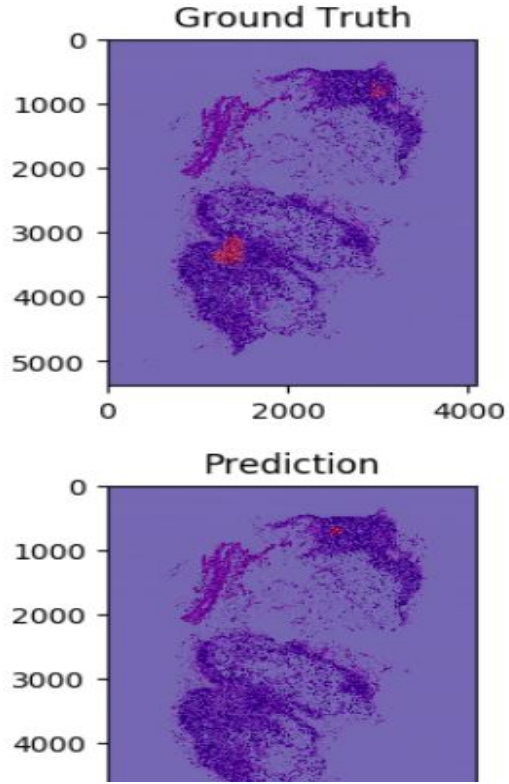





```
305/305 [==============================] - 3s 10ms/step
Test loss: 3.5401203531279726
Test accuracy: 0.780327868852459
```

Heatmap Result





Lesson

loss increase optimizer learning rate

imbalanced data oversampling downsampling data augmentation

overfitting dropout

store preprocessing data and model in time

begin from an complete deep learning pipe line and iterate quickly

small model for realtime prediction in practice

Future Work

shrink patch size or train on higher slide level with larger resolution

experiment small model such as mobilenet

experiment with weights trainable in InceptionV3

experiment with other data augmentation such as color normalization

Code walkthrough