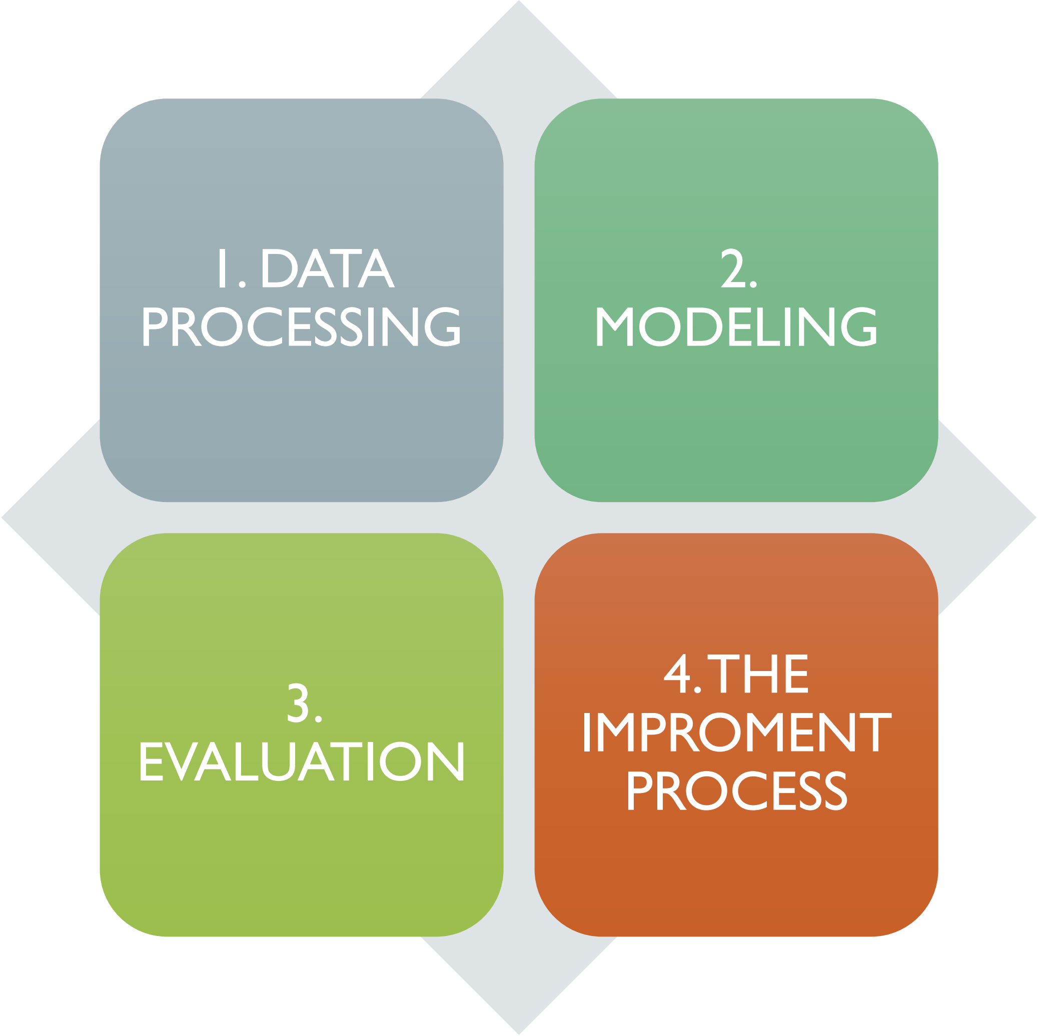
DETECTING CANCER METASTASES ON GIGAPIXEL PATHOLOGY IMAGE

XINLEI CAO UNI: xc2420

GOAL OF PROJECT

- Each year, the treatment decisions for more than 230,000 breast cancer patients in the U.S. hinge on whether the cancer has metastasized away from the breast. Metastasis detection is currently performed by pathologists reviewing large expanses of biological tissues. This process is labor intensive and error-prone. (copied from Paper: Detecting Cancer Metastases on Gigapixel Pathology Images)
- So we want to build a model to automatically detect and localize tumors.

INTRODUCTION TO MY METHOD



1. DATA PROCESSING

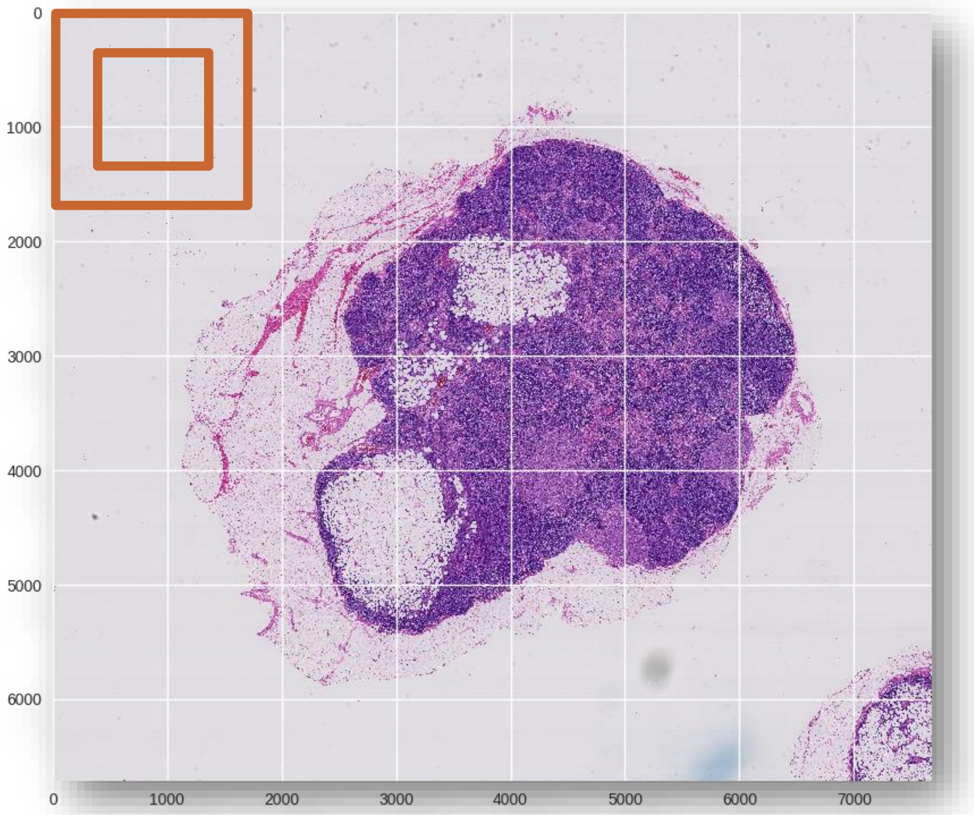
2. MODELING

3. EVALUATION

4. THE IMPROMENT PROCESS

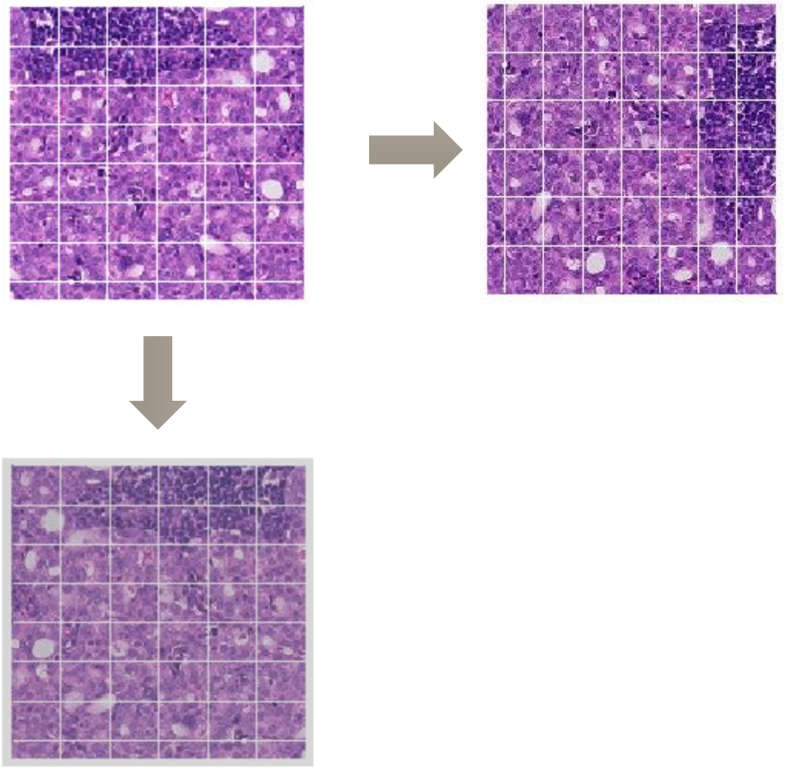
DATA PROCESSING

- **1. USING SLIDE WINSOW TO GENERATE IMAGE PATCH**
- Slide Level = 3
- Patch Size = 150, stride = 80, Center Size = 100



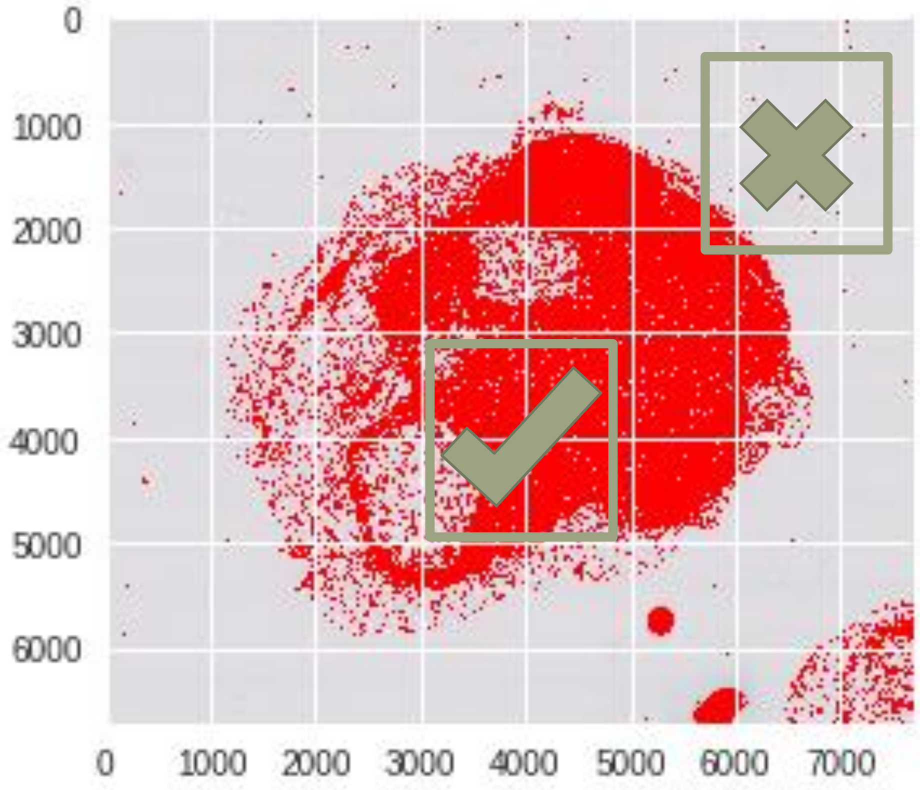
DATA PROCESSING

- **2. DATA AUGMENTATION**
- Flip and Rotation
- Only applied on the tumor patch
- 209 tumor images → 1672 tumor images



DATA PROCESSING

- **3. IGNORE NON-TISSUE AREAS**
- Ignore the patches whose tissue proportion is less than 0.2

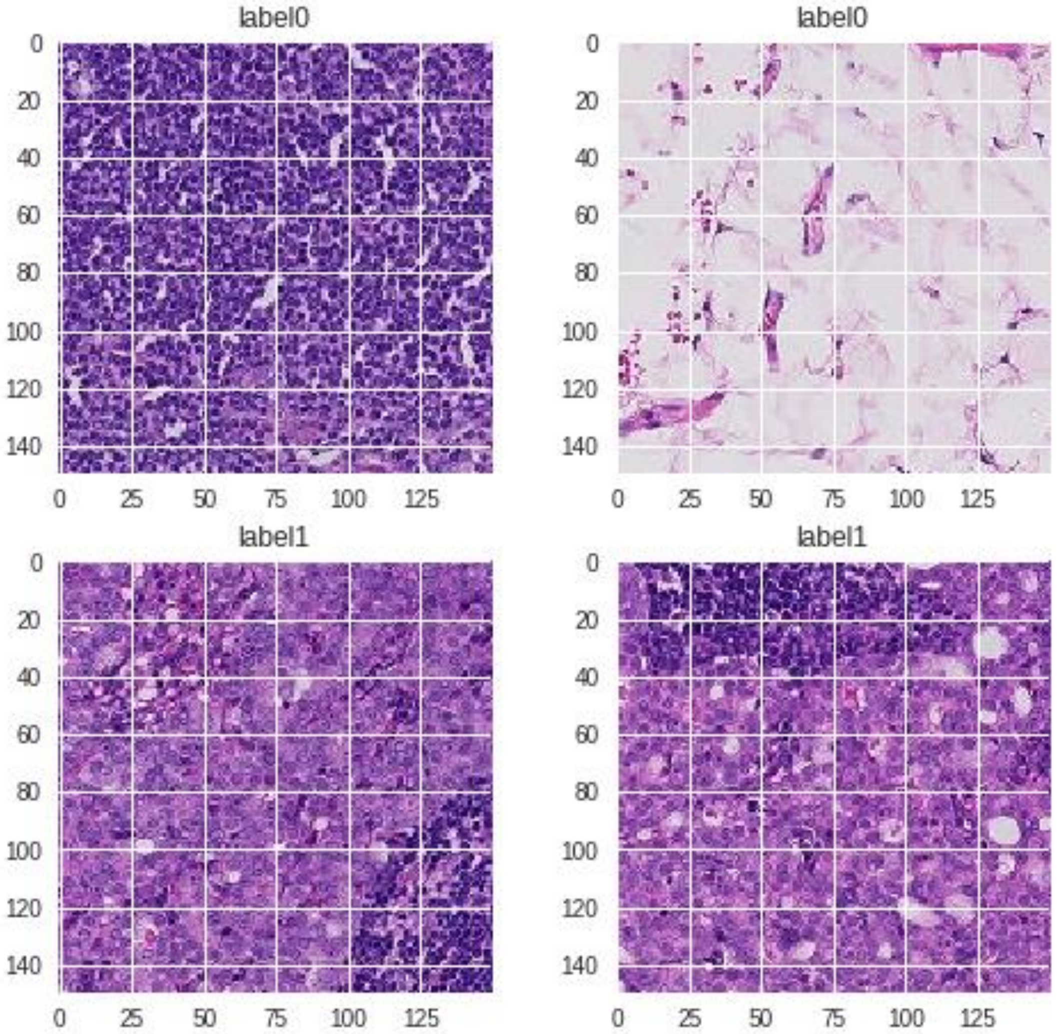


DATA PROCESSING

- **4. HANDEL UNBALANCED DATA**
- Resample the non-tumor patches to get a balanced data
- 1672 tumor images
- 1672 non-tumor images

DATA PROCESSING

- **LOOK AT THE FINAL DATA**



MODEL

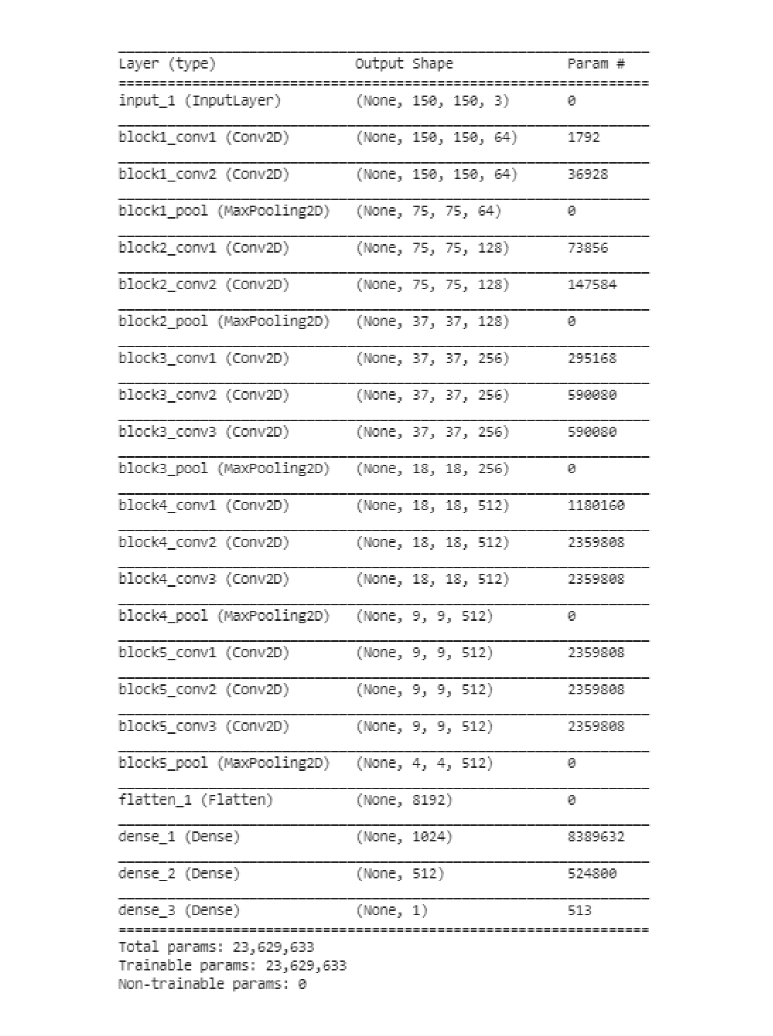
BASE MODEL: VGG16

Add three dense layers

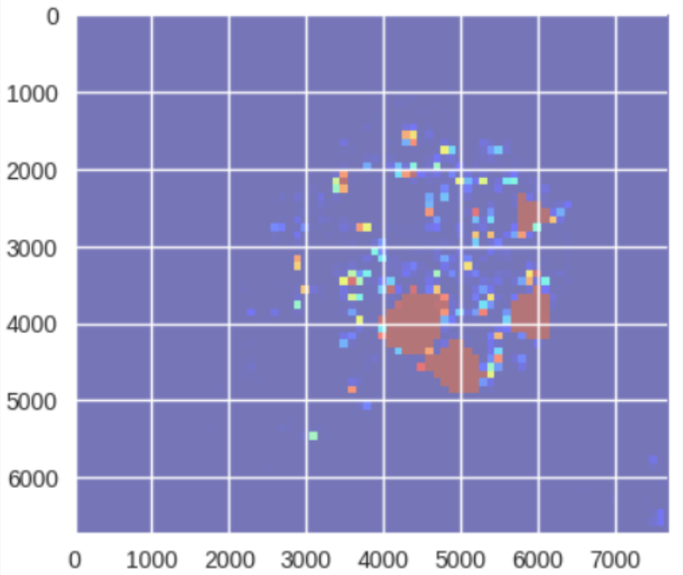
Not A Transfer Learning !

Because the data size is enough to train the network size of VGG.

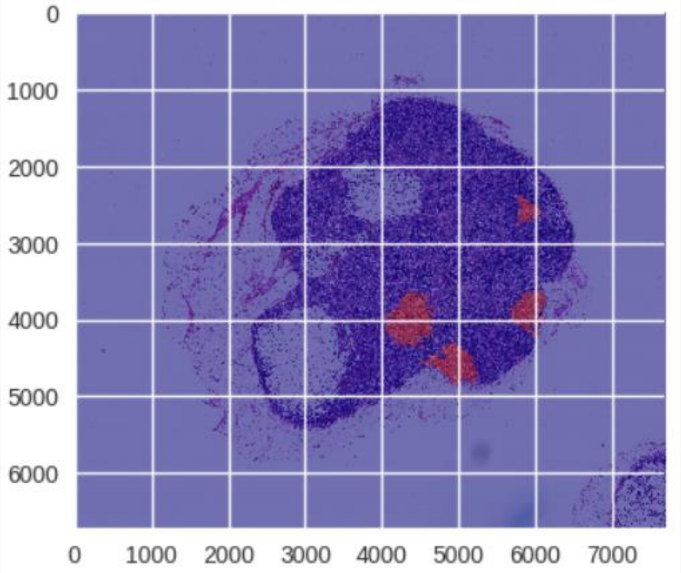
The categories of ImageNet and cell pictures are totally different.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         (None, 150, 150, 3)       0
_________________________________________________________________
block1_conv1 (Conv2D)        (None, 150, 150, 64)      1792
_________________________________________________________________
block1_conv2 (Conv2D)        (None, 150, 150, 64)      36928
_________________________________________________________________
block1_pool (MaxPooling2D)   (None, 75, 75, 64)        0
_________________________________________________________________
block2_conv1 (Conv2D)        (None, 75, 75, 128)       73856
_________________________________________________________________
block2_conv2 (Conv2D)        (None, 75, 75, 128)       147584
_________________________________________________________________
block2_pool (MaxPooling2D)   (None, 37, 37, 128)       0
_________________________________________________________________
block3_conv1 (Conv2D)        (None, 37, 37, 256)       295168
_________________________________________________________________
block3_conv2 (Conv2D)        (None, 37, 37, 256)       590080
_________________________________________________________________
block3_conv3 (Conv2D)        (None, 37, 37, 256)       590080
_________________________________________________________________
block3_pool (MaxPooling2D)   (None, 18, 18, 256)       0
_________________________________________________________________
block4_conv1 (Conv2D)        (None, 18, 18, 512)       1180160
_________________________________________________________________
block4_conv2 (Conv2D)        (None, 18, 18, 512)       2359808
_________________________________________________________________
block4_conv3 (Conv2D)        (None, 18, 18, 512)       2359808
_________________________________________________________________
block4_pool (MaxPooling2D)   (None, 9, 9, 512)         0
_________________________________________________________________
block5_conv1 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_conv2 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_conv3 (Conv2D)        (None, 9, 9, 512)         2359808
_________________________________________________________________
block5_pool (MaxPooling2D)   (None, 4, 4, 512)         0
_________________________________________________________________
flatten_1 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_1 (Dense)              (None, 1024)              8389632
_________________________________________________________________
dense_2 (Dense)              (None, 512)               524800
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 513
=================================================================
Total params: 23,629,633
Trainable params: 23,629,633
Non-trainable params: 0
```



- prediction

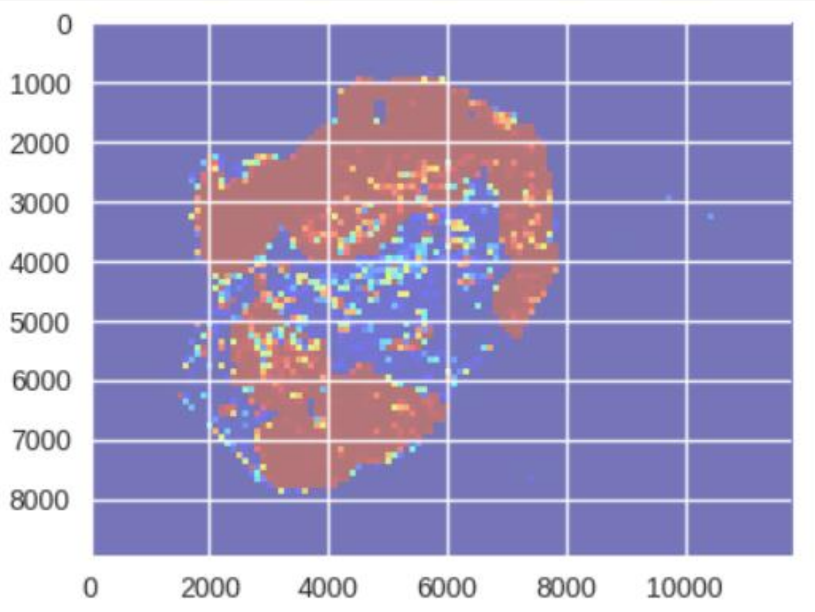


- Ground truth

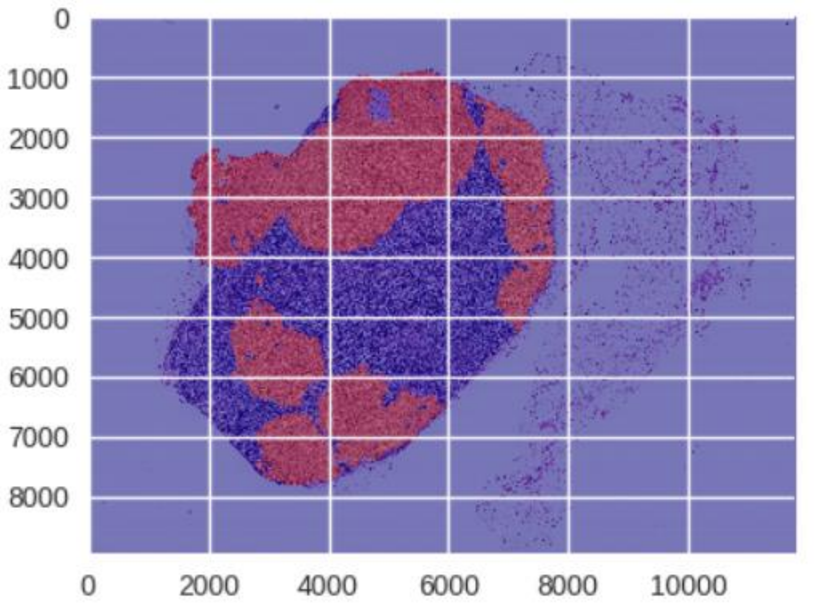
EVALUATION

Heatmap Compare on the Train Image

AUC = 0.998303400461 2835



- prediction



- Ground truth

EVALUATION TUMOR 110

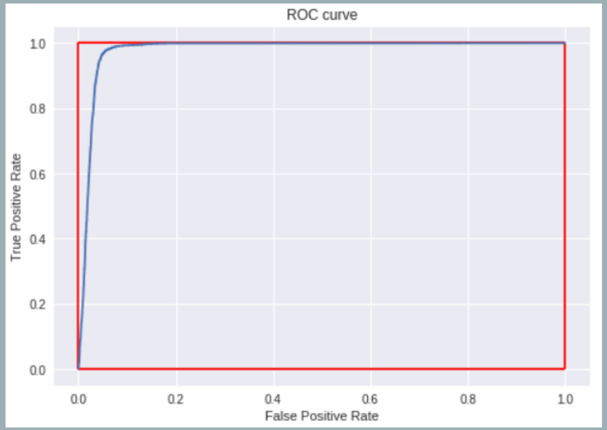
Confusion Matrix	Ground truth NO	Ground truth YES
Prediction NO	85251678	3812081
Prediction YES	901282	15547919

Precision = 0.989538583468287

Precision = 0.957198292068944

F1_score = 0.973099810184209

Test AUC = 0.97901706259915



MORE DETAILS

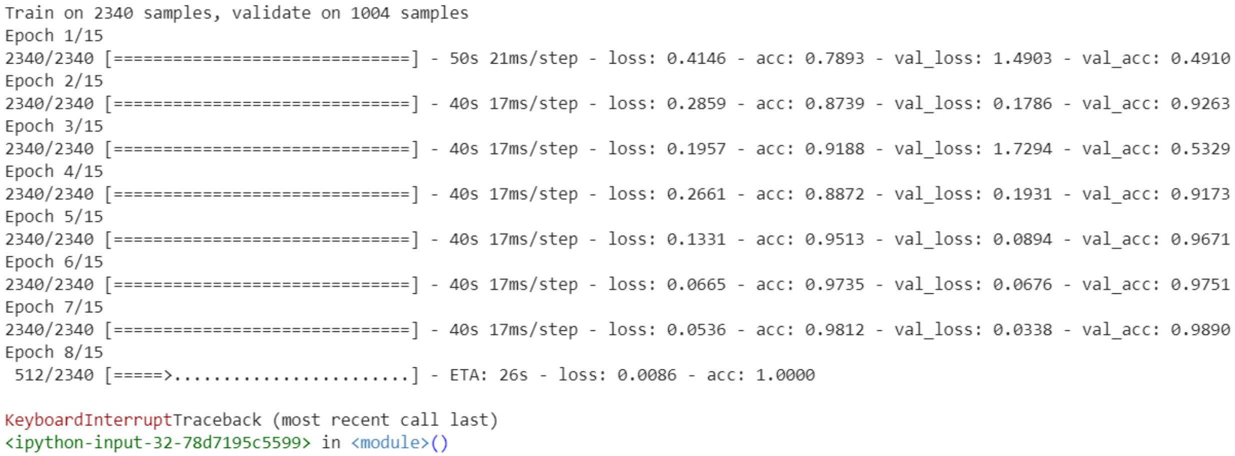
- **ABOUT InceptionV3**
- **Because the model is so large, it’s hard to train from scratch, then we must import the weights of ImageNet and freeze the model or fine tuning, but it doesn’t work well.**

MORE DETAILS

- **About Optimizer and Learning Rate in my model**
- **Found the Adam didn’t work well**
- **GradientDescent cannot converge well**
- **Used Momentum and tried many learning rate**

MORE DETAILS

- **Early Stop to prevent overfitting**
- **Pay attention to the training accuracy and validation accuracy**



```
Train on 2340 samples, validate on 1004 samples
Epoch 1/15
2340/2340 [==============================] - 50s 21ms/step - loss: 0.4146 - acc: 0.7893 - val_loss: 1.4903 - val_acc: 0.4910
Epoch 2/15
2340/2340 [==============================] - 40s 17ms/step - loss: 0.2859 - acc: 0.8739 - val_loss: 0.1786 - val_acc: 0.9263
Epoch 3/15
2340/2340 [==============================] - 40s 17ms/step - loss: 0.1957 - acc: 0.9188 - val_loss: 1.7294 - val_acc: 0.5329
Epoch 4/15
2340/2340 [==============================] - 40s 17ms/step - loss: 0.2661 - acc: 0.8872 - val_loss: 0.1931 - val_acc: 0.9173
Epoch 5/15
2340/2340 [==============================] - 40s 17ms/step - loss: 0.1331 - acc: 0.9513 - val_loss: 0.0894 - val_acc: 0.9671
Epoch 6/15
2340/2340 [==============================] - 40s 17ms/step - loss: 0.0665 - acc: 0.9735 - val_loss: 0.0676 - val_acc: 0.9751
Epoch 7/15
2340/2340 [==============================] - 40s 17ms/step - loss: 0.0536 - acc: 0.9812 - val_loss: 0.0338 - val_acc: 0.9890
Epoch 8/15
 512/2340 [=====>........................] - ETA: 26s - loss: 0.0086 - acc: 1.0000

KeyboardInterruptTraceback (most recent call last)
<ipython-input-32-78d7195c5599> in <module>()
```

MORE DETAILS

- **About Out of Memory**
- **I did lots of optimization in the code and release the large variable after use.**

THANK YOU!