Detecting Cancer Metastases on Gigapixel Pathology Images

Zhirui (Alex) Yang

Major: Data Science

Email: zy2494@columbia.edu

COLUMBIA UNIVERSITY
Data Science Institute

1 Background

2 Methods and Models

3 Results and Heatmaps

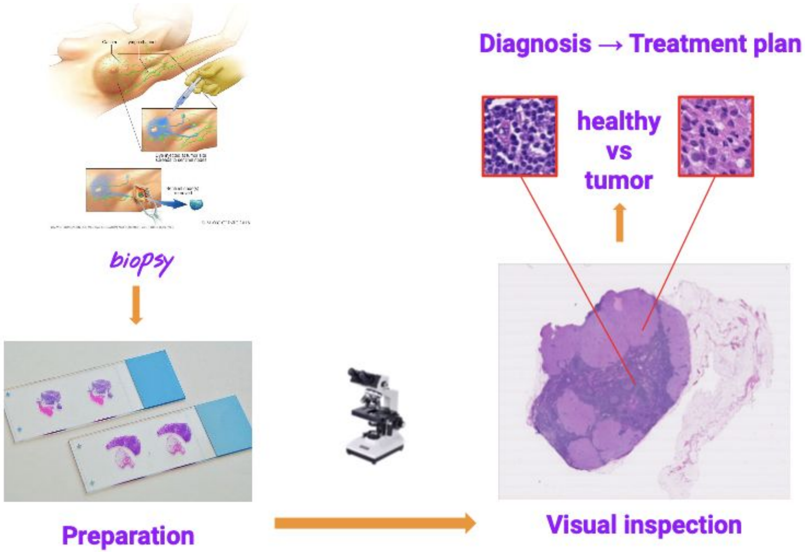
4 Conclusion and Future Work

Background

- Each year, huge number of breast cancer patients hinge on whether the cancer has metastasized away from the breast.[1]

- Microscopic examination requires highly skilled pathologists and is fairly time-consuming and error-prone.[1]

- Computer assisted detection could increase the sensitivity, speed, and consistency of metastasis detection.[2]



[1] Detecting Cancer Metastases on Gigapixel Pathology Images

[2] Litjens, G., et al.: Deep learning as a tool for increased accuracy and efficiency of histopathological diagnosis. Sci. Reports 6 (2016)

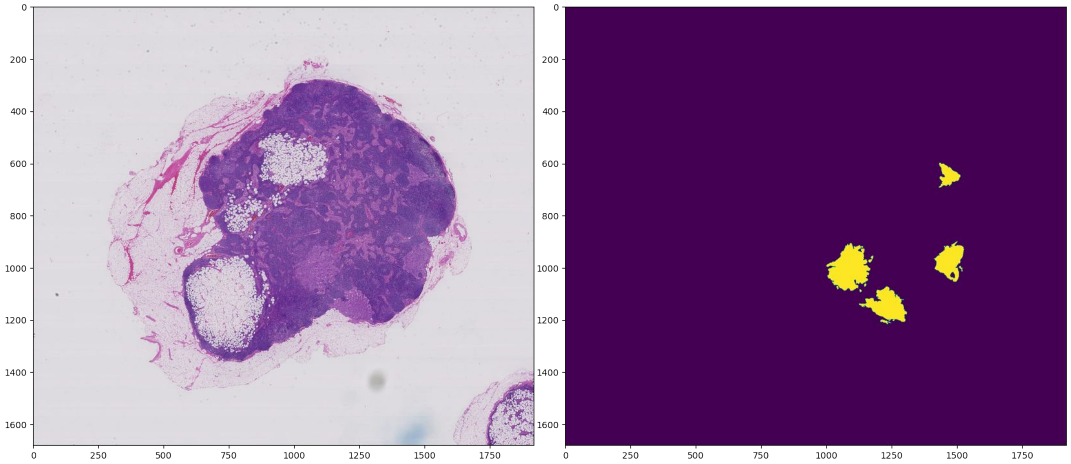
Data Processing - Raw Data Description

Raw Data:

- Get some slides from database
- Have some function to load image of certain level

Image used in my project

- "101", "094", "110", "016", "078", "031", "064", "091", "075", "094", "084"



- Left is the original slide, and right is the mask.
- Cancer tumor cells are marked as 1 shown yellow in the image, and other area is marked as 0 shown purple in the image.

Data Processing - Train-Test Dataset Split

- Ensure model only learn the information from training set

- Better generalization ability for future use

  - If this tools will be applied in the real treatment process, model need to be able to deal with new patients.

- Training set : "101", "094", "110", "016", "078", "031", "064"

- Test set : "091", "075", "094", "084"

Data Processing - Data Augmentation

- Use the methods mentioned in the paper [1]

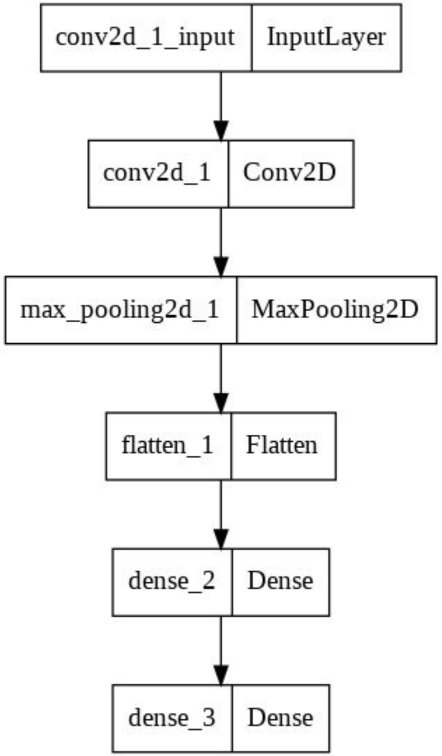
- Apply data augmentation randomly on the pitches

- Rotate the input patch by 4 multiples of 90◦, apply a left-right flip and repeat the rotations.
- Perturb color:
    - brightness with a maximum delta of 64/255
    - saturation with a maximum delta of 0.25
    - hue with a maximum delta of 0.04
    - contrast with a maximum delta of 0.75

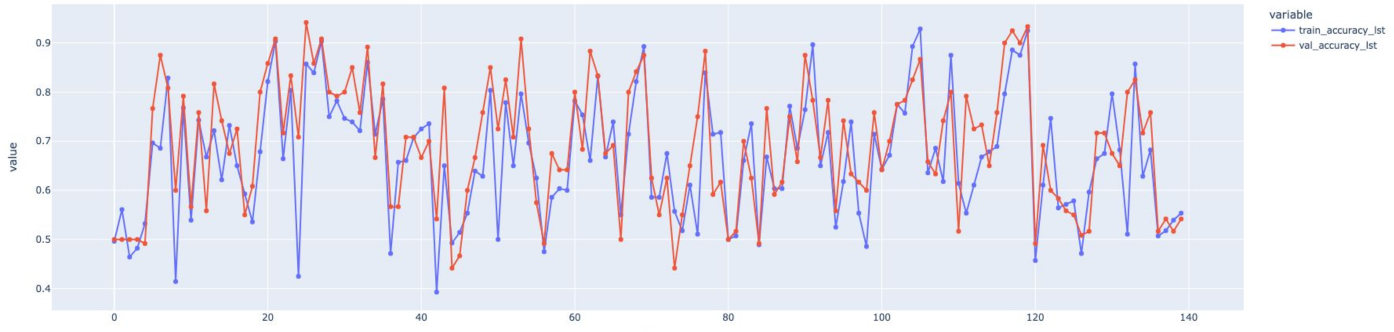
[1] Detecting Cancer Metastases on Gigapixel Pathology Images

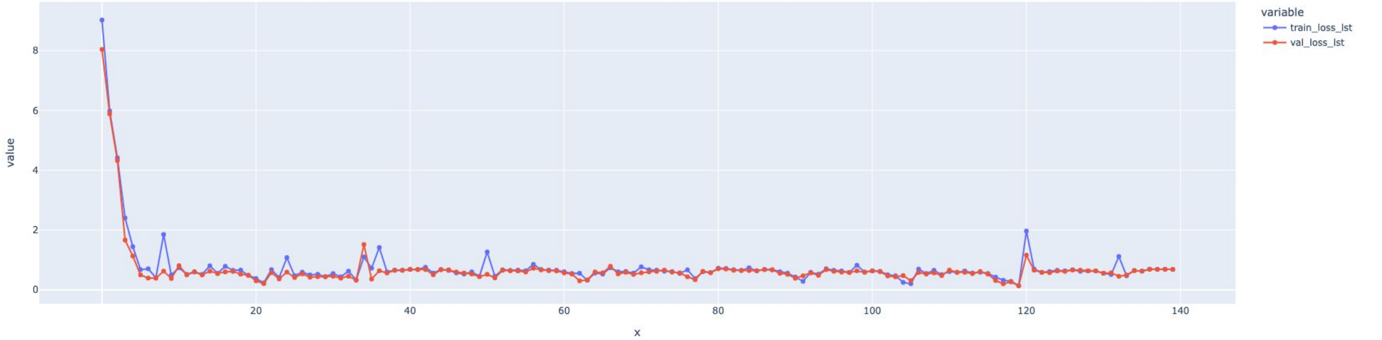
Models - Base Model

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 297, 297, 16)      448

max_pooling2d_1 (MaxPooling  (None, 148, 148, 16)      0
2D)

flatten_1 (Flatten)          (None, 350464)            0

dense_2 (Dense)              (None, 32)                11214880

dense_3 (Dense)              (None, 1)                 33

=================================================================
Total params: 11,215,361
Trainable params: 11,215,361
Non-trainable params: 0
_________________________________________________________________
```



Models - Training process



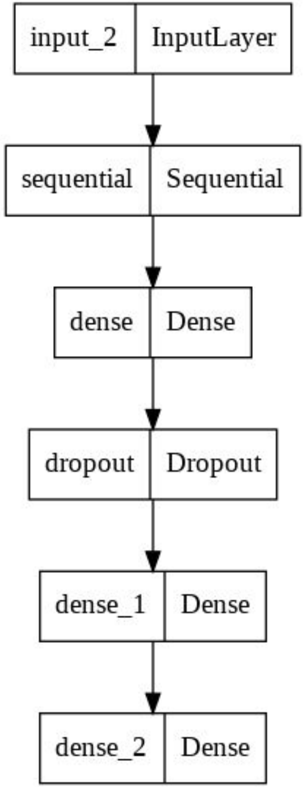


Models - One-zoom (5) Model

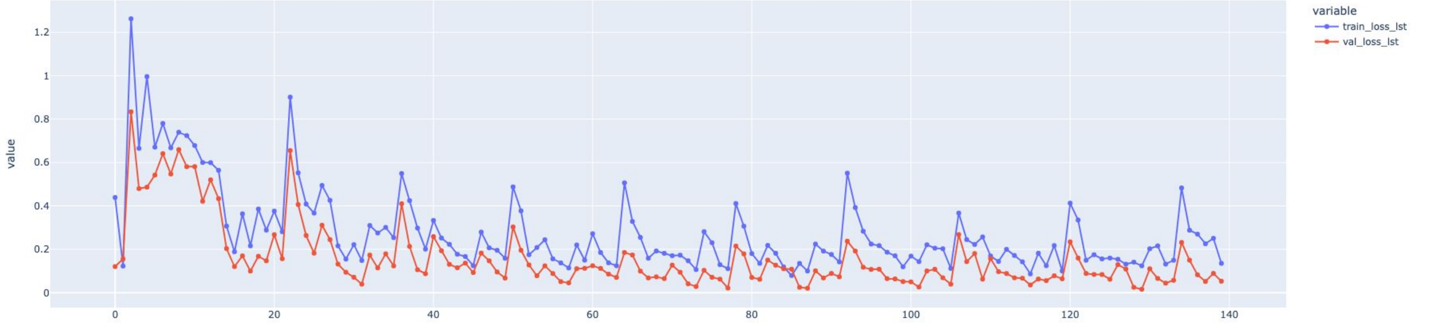
```
inception_zoom1 = InceptionV3(
    weights='imagenet',
    include_top=False,
    input_shape=(patch_len, patch_len, 3))

# freeze the inception model to increase training speed
inception_zoom1.trainable = False

model_zoom1 = models.Sequential()
model_zoom1.add(inception_zoom1)
model_zoom1.add(layers.GlobalAveragePooling2D())

input_zoom1 = layers.Input(shape=(patch_len, patch_len, 3))

encoded_zoom1 = model_zoom1(input_zoom1)

dense1 = layers.Dense(128, activation='relu')(encoded_zoom1)
drop_layer = layers.Dropout(0.5)(dense1)
dense2 = layers.Dense(32, activation='relu')(drop_layer)

output = layers.Dense(1, activation='sigmoid')(dense2)
model = models.Model(inputs=[input_zoom1], outputs=output)
```

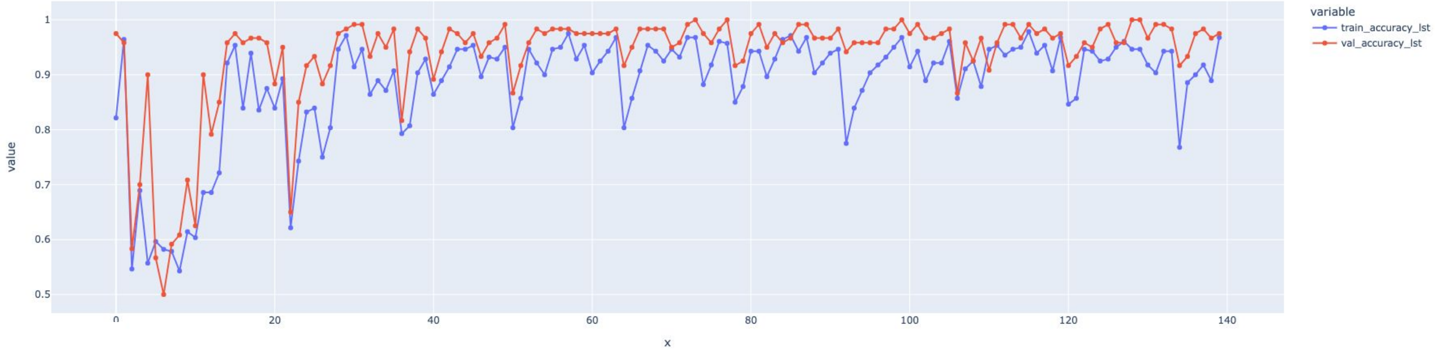
Layer (type)	Output Shape	Param #
input_2 (InputLayer)	[(None, 299, 299, 3)]	0
sequential (Sequential)	(None, 2048)	21802784
dense (Dense)	(None, 128)	262272
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 32)	4128
dense_2 (Dense)	(None, 1)	33

Total params: 22,069,217
Trainable params: 266,433
Non-trainable params: 21,802,784



Models - Training process

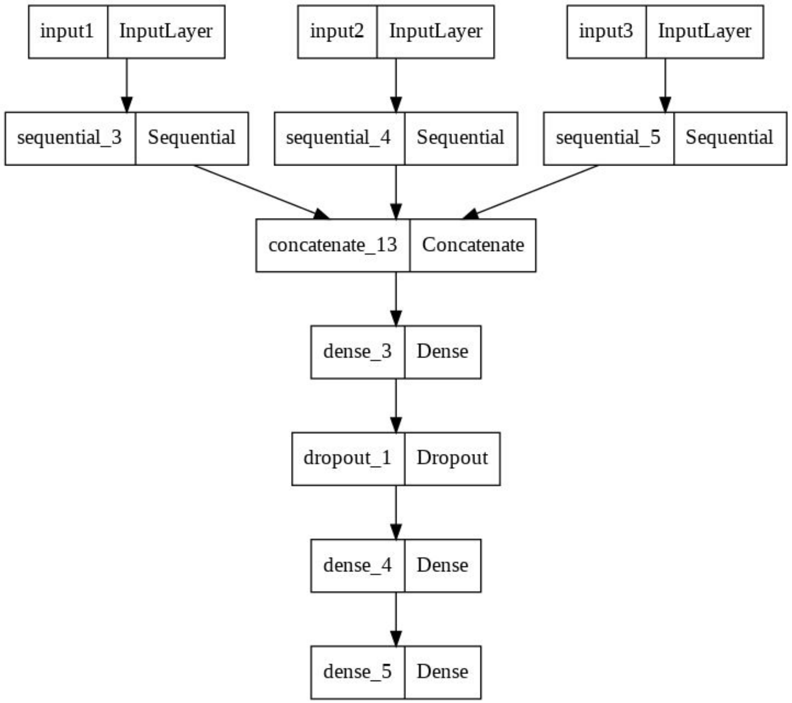




Models Three-zooms (567) Model

```
inception_zoom1 = InceptionV3(
    weights='imagenet',
    include_top=False,
    input_shape=(patch_len, patch_len, 3))

inception_zoom2 = InceptionV3(
    weights='imagenet',
    include_top=False,
    input_shape=(patch_len, patch_len, 3))

inception_zoom3 = InceptionV3(
    weights='imagenet',
    include_top=False,
    input_shape=(patch_len, patch_len, 3))
```

```
model_zoom1 = models.Sequential()
model_zoom1.add(inception_zoom1)
model_zoom1.add(layers.GlobalAveragePooling2D())

model_zoom2 = models.Sequential()
model_zoom2.add(inception_zoom2)
model_zoom2.add(layers.GlobalAveragePooling2D())

model_zoom3 = models.Sequential()
model_zoom3.add(inception_zoom3)
model_zoom3.add(layers.GlobalAveragePooling2D())
```



Three-zooms (567) Model

Layer (type)	Output Shape	Param #	Connected to
input1 (InputLayer)	[(None, 299, 299, 3)]	0	[]
input2 (InputLayer)	[(None, 299, 299, 3)]	0	[]
input3 (InputLayer)	[(None, 299, 299, 3)]	0	[]
sequential (Sequential)	(None, 2048)	21802784	['input1[0][0]']
sequential_1 (Sequential)	(None, 2048)	21802784	['input2[0][0]']
sequential_2 (Sequential)	(None, 2048)	21802784	['input3[0][0]']
concatenate_6 (Concatenate)	(None, 6144)	0	['sequential[0][0]', 'sequential_1[0][0]', 'sequential_2[0][0]']
dense (Dense)	(None, 256)	1573120	['concatenate_6[0][0]']
dropout (Dropout)	(None, 256)	0	['dense[0][0]']
dense_1 (Dense)	(None, 126)	32382	['dropout[0][0]']
dense_2 (Dense)	(None, 1)	127	['dense_1[0][0]']

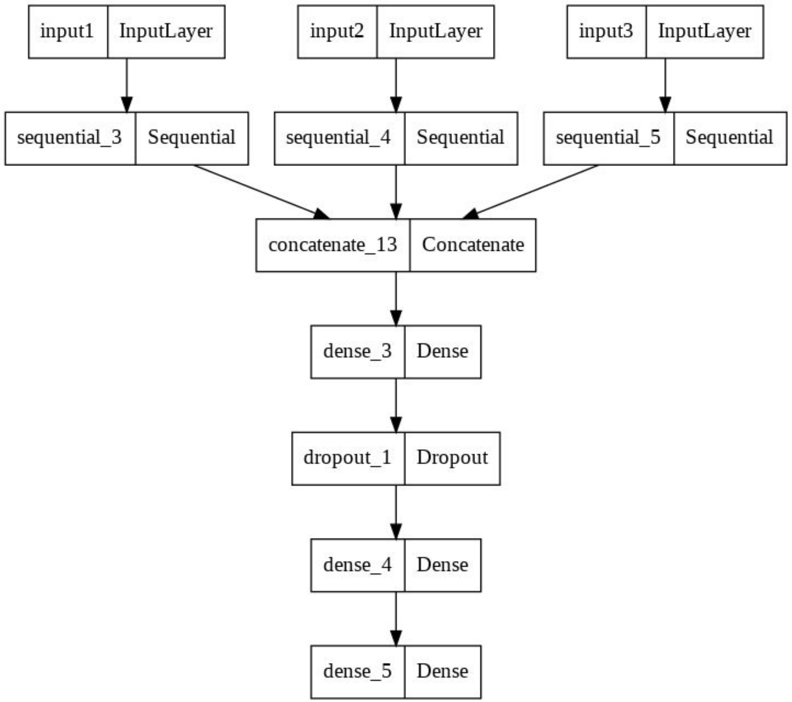
Total params: 67,013,981
Trainable params: 1,605,629
Non-trainable params: 65,408,352

Models - Training process

Models Three-zooms (357) Model

```
inception_zoom1 = InceptionV3(
    weights='imagenet',
    include_top=False,
    input_shape=(patch_len, patch_len, 3))

inception_zoom2 = InceptionV3(
    weights='imagenet',
    include_top=False,
    input_shape=(patch_len, patch_len, 3))

inception_zoom3 = InceptionV3(
    weights='imagenet',
    include_top=False,
    input_shape=(patch_len, patch_len, 3))
```

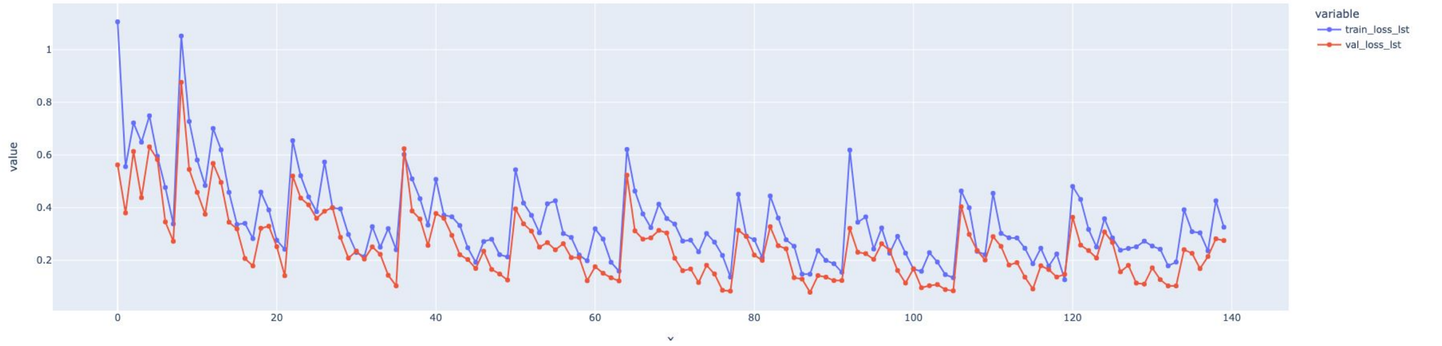
```
model_zoom1 = models.Sequential()
model_zoom1.add(inception_zoom1)
model_zoom1.add(layers.GlobalAveragePooling2D())

model_zoom2 = models.Sequential()
model_zoom2.add(inception_zoom2)
model_zoom2.add(layers.GlobalAveragePooling2D())

model_zoom3 = models.Sequential()
model_zoom3.add(inception_zoom3)
model_zoom3.add(layers.GlobalAveragePooling2D())
```



Models - Three-zooms (357) Model

```
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input1 (InputLayer)            [(None, 299, 299, 3  0           []
                                )]

 input2 (InputLayer)            [(None, 299, 299, 3  0           []
                                )]

 input3 (InputLayer)            [(None, 299, 299, 3  0           []
                                )]

 sequential (Sequential)        (None, 2048)         21802784    ['input1[0][0]']

 sequential_1 (Sequential)      (None, 2048)         21802784    ['input2[0][0]']

 sequential_2 (Sequential)      (None, 2048)         21802784    ['input3[0][0]']

 concatenate_6 (Concatenate)    (None, 6144)         0           ['sequential[0][0]',
                                                                  'sequential_1[0][0]',
                                                                  'sequential_2[0][0]']

 dense (Dense)                  (None, 256)          1573120     ['concatenate_6[0][0]']

 dropout (Dropout)              (None, 256)          0           ['dense[0][0]']

 dense_1 (Dense)                (None, 126)          32382       ['dropout[0][0]']

 dense_2 (Dense)                (None, 1)            127         ['dense_1[0][0]']

==================================================================================================
Total params: 67,013,981
Trainable params: 1,605,629
Non-trainable params: 65,408,352
__________________________________________________________________________________________________
```

Models - Training process



COLUMBIA UNIVERSITY
Data Science Institute

Results - Model Performance (classification report)

Base Model

	precision	recall	f1-score	support
normal	0.61	0.63	0.62	4000
cancer	0.62	0.60	0.61	4000
accuracy			0.62	8000
macro avg	0.62	0.62	0.62	8000
weighted avg	0.62	0.62	0.62	8000

One-zoom (5) Model

	precision	recall	f1-score	support
normal	0.58	1.00	0.73	4000
cancer	0.99	0.26	0.42	4000
accuracy			0.63	8000
macro avg	0.78	0.63	0.57	8000
weighted avg	0.78	0.63	0.57	8000

Three-zooms (567) Model

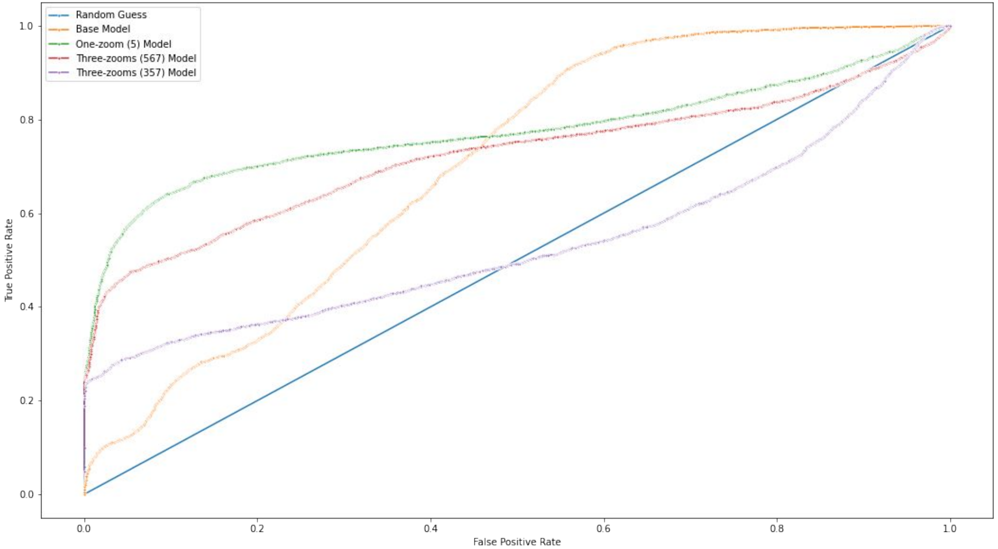
	precision	recall	f1-score	support
normal	0.64	0.96	0.77	4000
cancer	0.92	0.45	0.61	4000
accuracy			0.71	8000
macro avg	0.78	0.71	0.69	8000
weighted avg	0.78	0.71	0.69	8000

Three-zooms (357) Model

	precision	recall	f1-score	support
normal	0.57	0.98	0.72	4000
cancer	0.94	0.25	0.40	4000
accuracy			0.62	8000
macro avg	0.76	0.62	0.56	8000
weighted avg	0.76	0.62	0.56	8000

Results - Model Performance (ROC and AUC)

Model	AUC
Random Guess	0.5
Base Model	0.698
One-zoom (5) Model	0.775
Three-zooms (567) Model	0.722
Three-zooms (357) Model	0.533

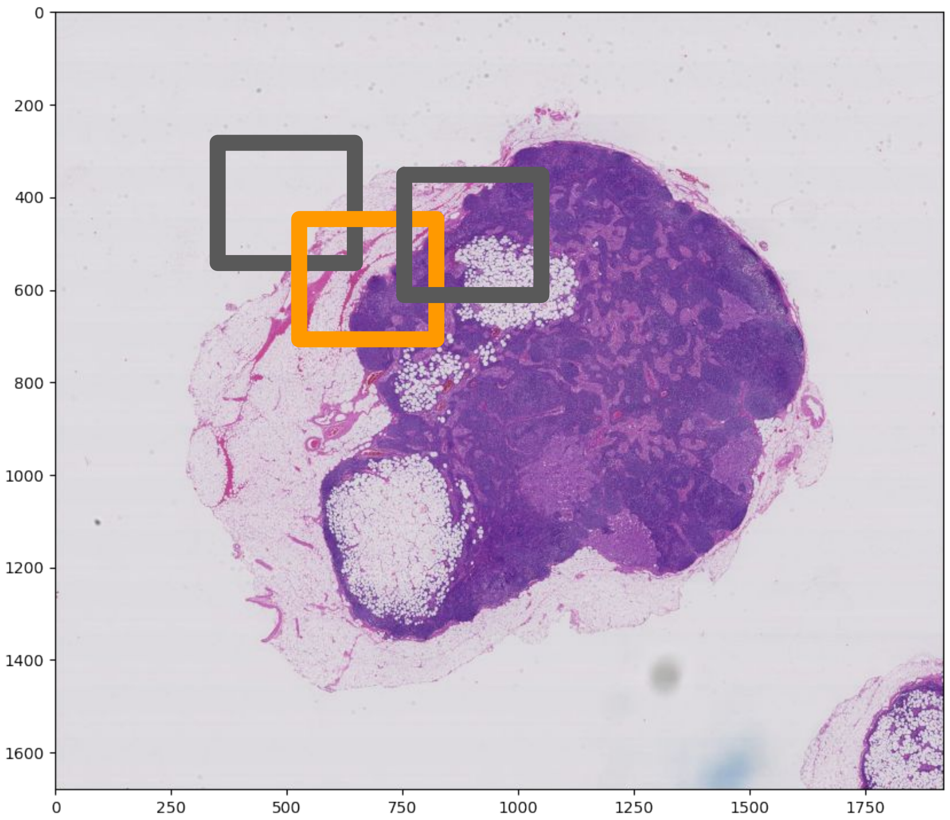


Generate Heatmaps

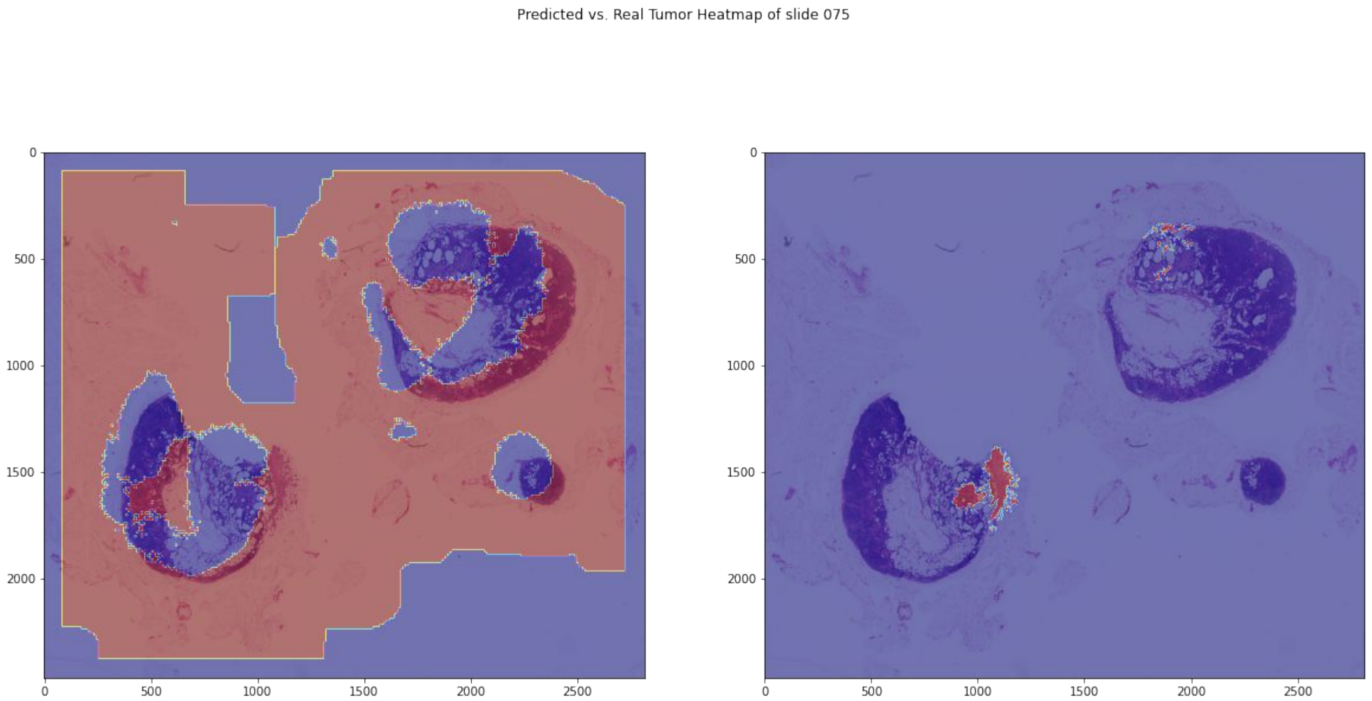
Use sliding window to generate samples for model

- Move x steps and y steps to generate new samples and make predictions

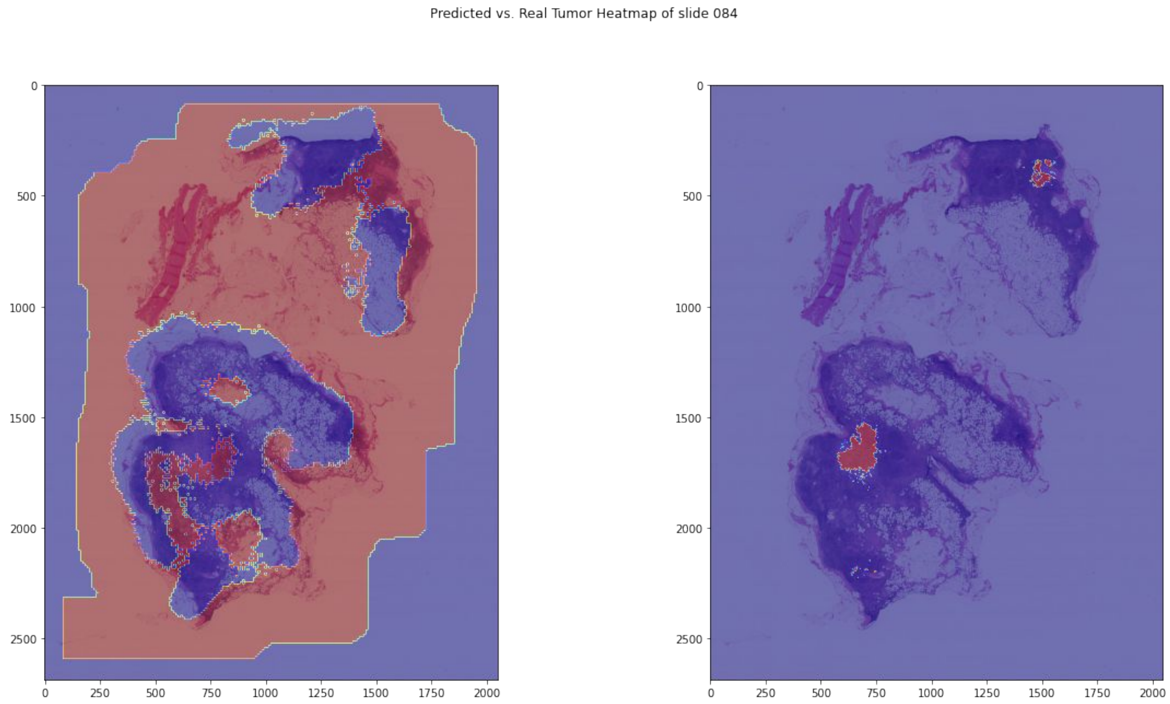
- If model prediction is 1, then mark all 128*128 pixels as 1, otherwise, mark pixels as 0



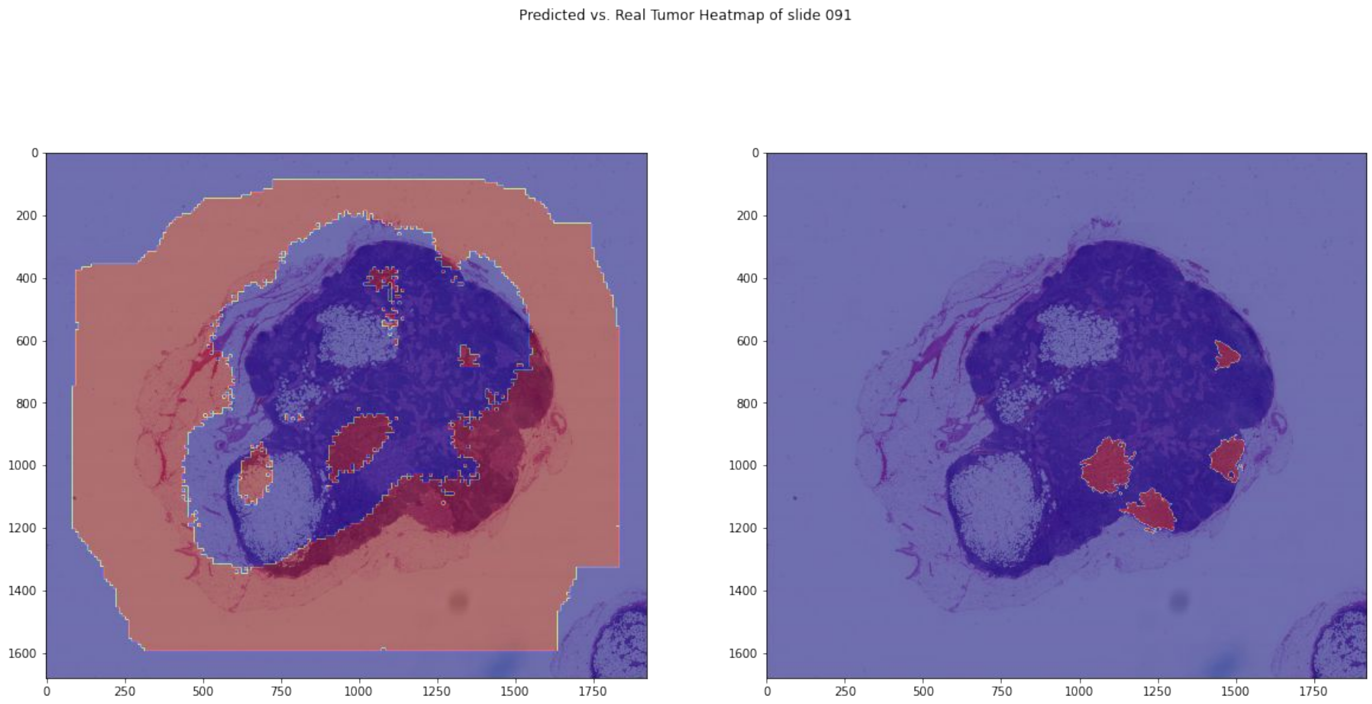
Base Model


Predicted vs. Real Tumor Heatmap of slide 075
0
500
1000
1500
2000
0
500
1000
1500
2000
2500

Base Model

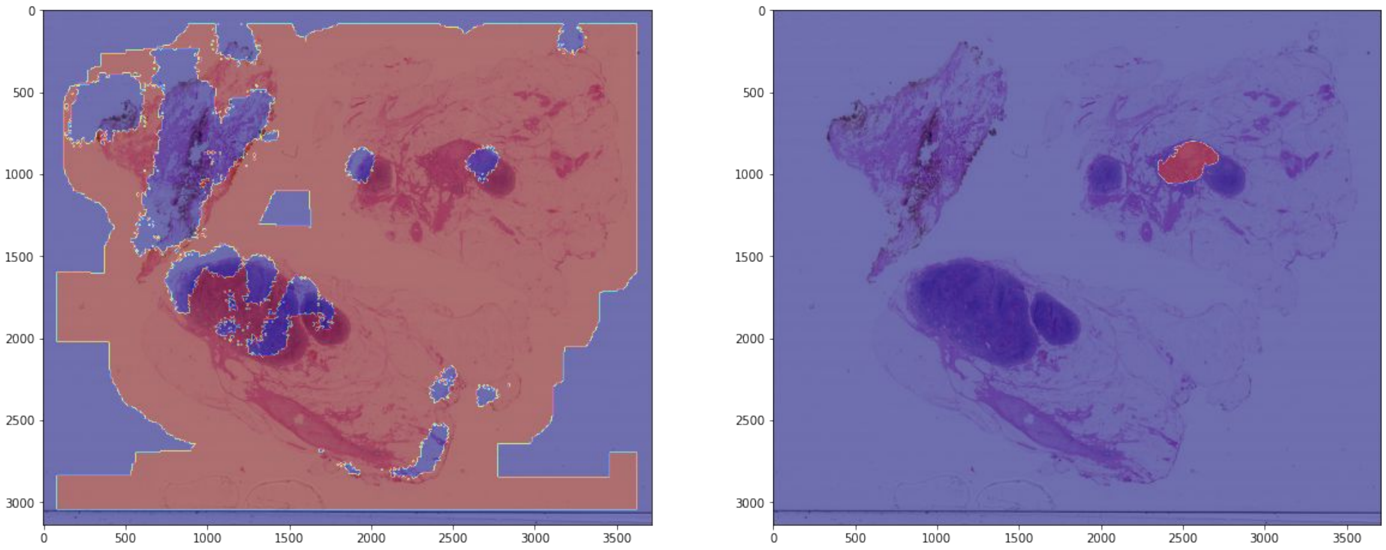

Predicted vs. Real Tumor Heatmap of slide 084
0
500
1000
1500
2000
2500
0
250
500
750
1000
1250
1500
1750
2000
0
500
1000
1500
2000
2500
0
250
500
750
1000
1250
1500
1750
2000

Base Model

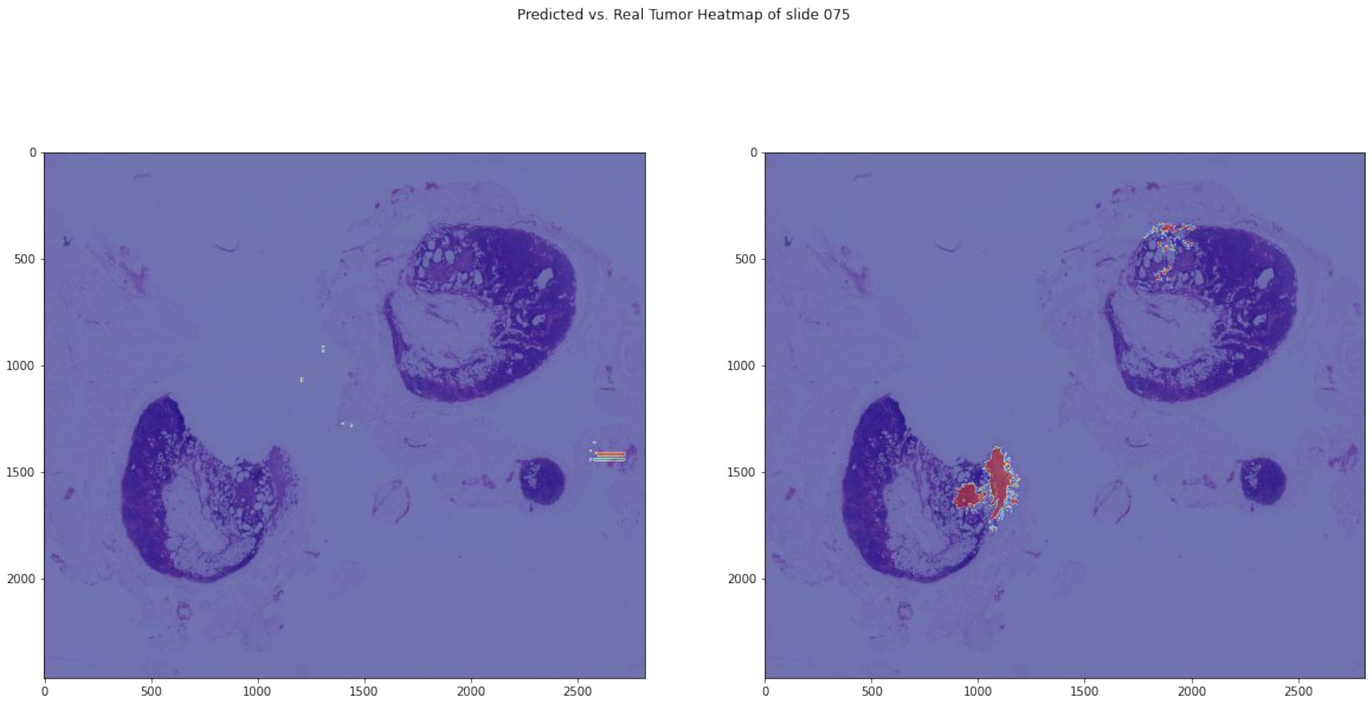

Predicted vs. Real Tumor Heatmap of slide 091
0
200
400
600
800
1000
1200
1400
1600
0
250
500
750
1000
1250
1500
1750
0
200
400
600
800
1000
1200
1400
1600
0
250
500
750
1000
1250
1500
1750

Base Model

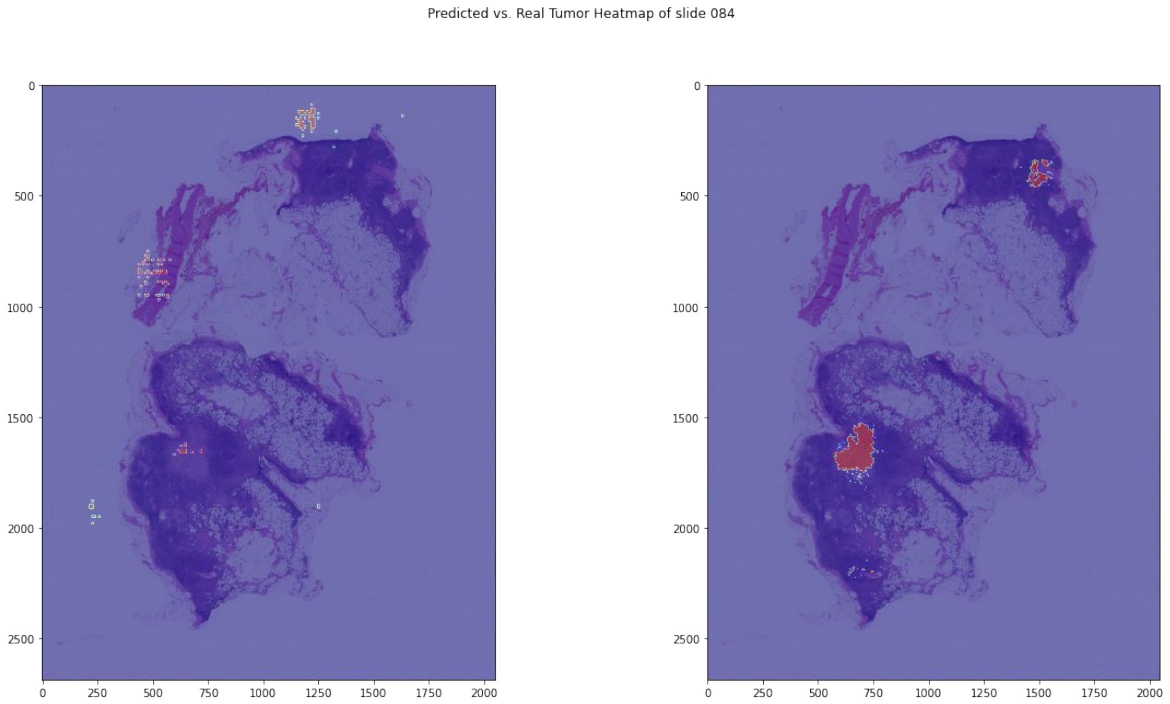
Predicted vs. Real Tumor Heatmap of slide 094



One-zoom (5) Model

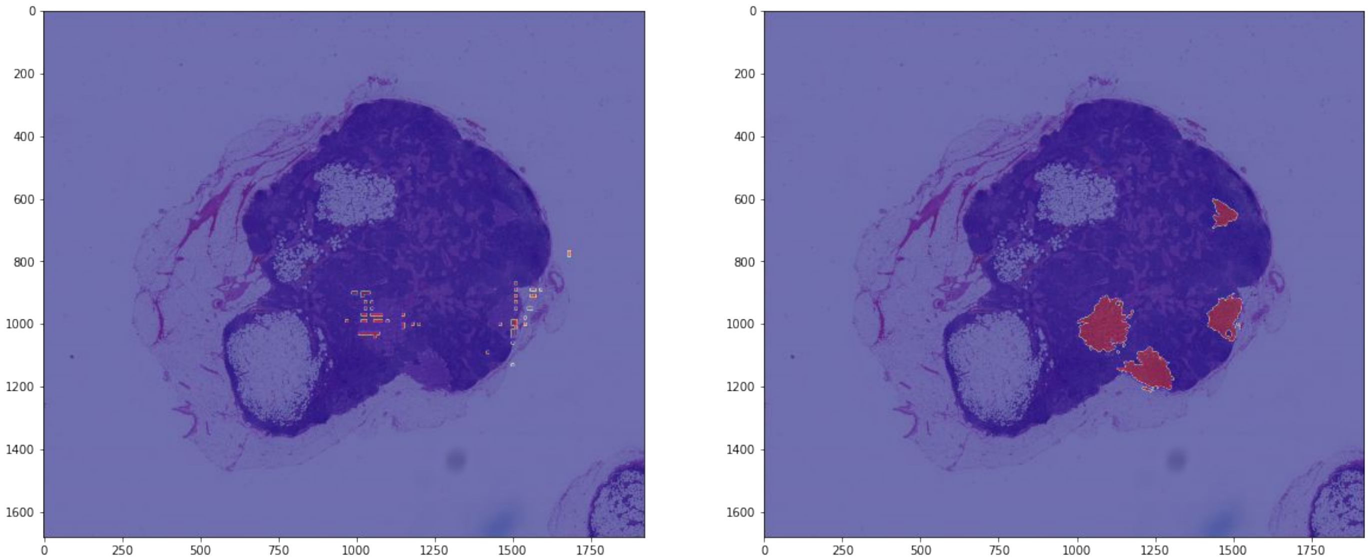


One-zoom (5) Model



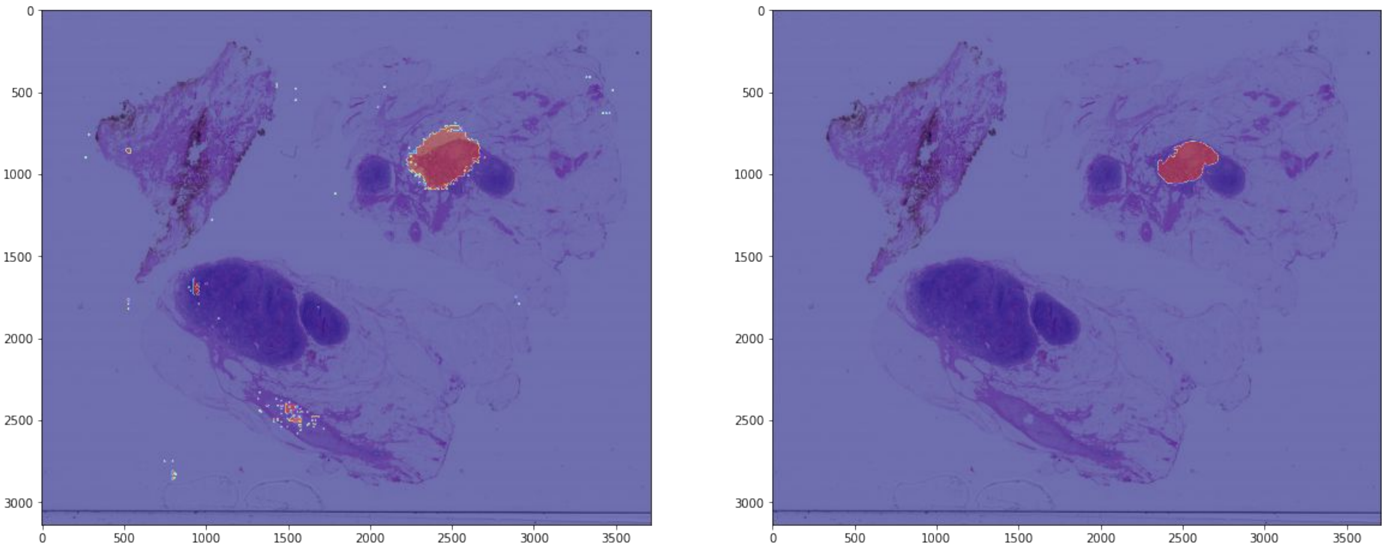
One-zoom (5) Model

Predicted vs. Real Tumor Heatmap of slide 091

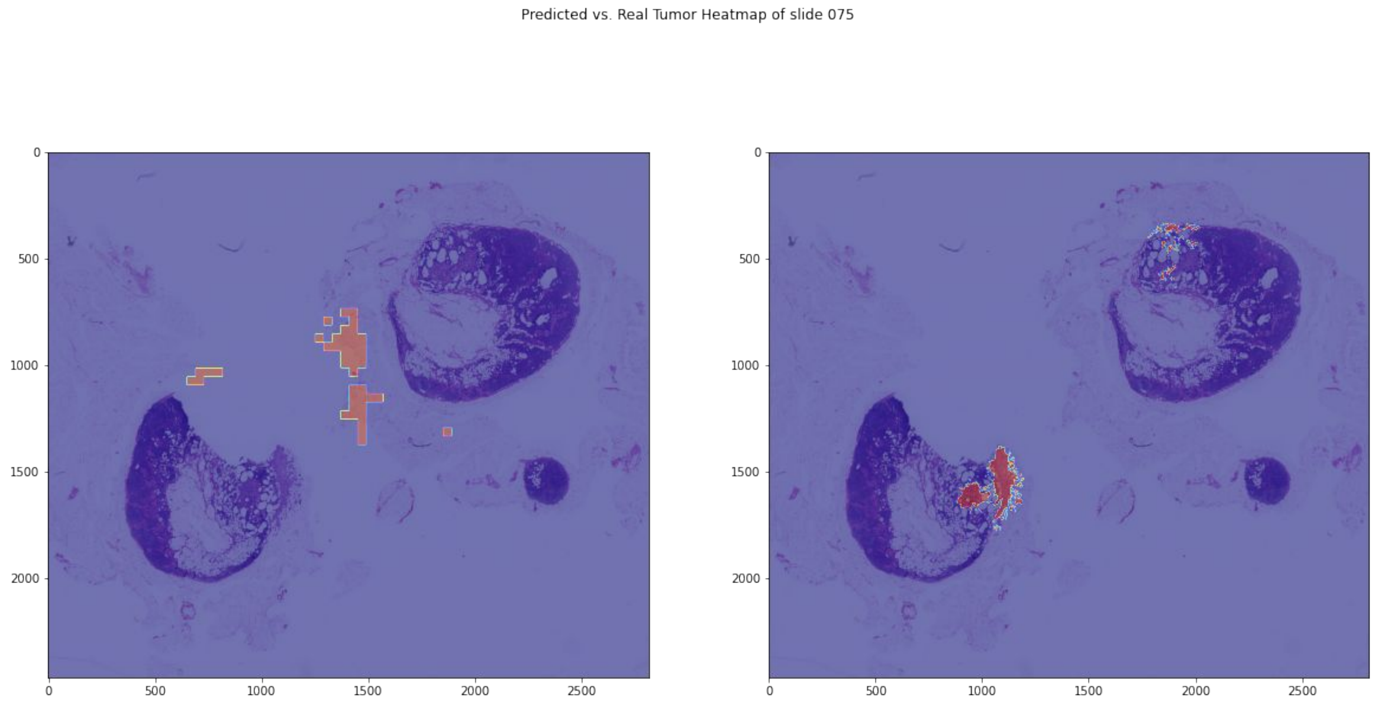


One-zoom (5) Model

Predicted vs. Real Tumor Heatmap of slide 094

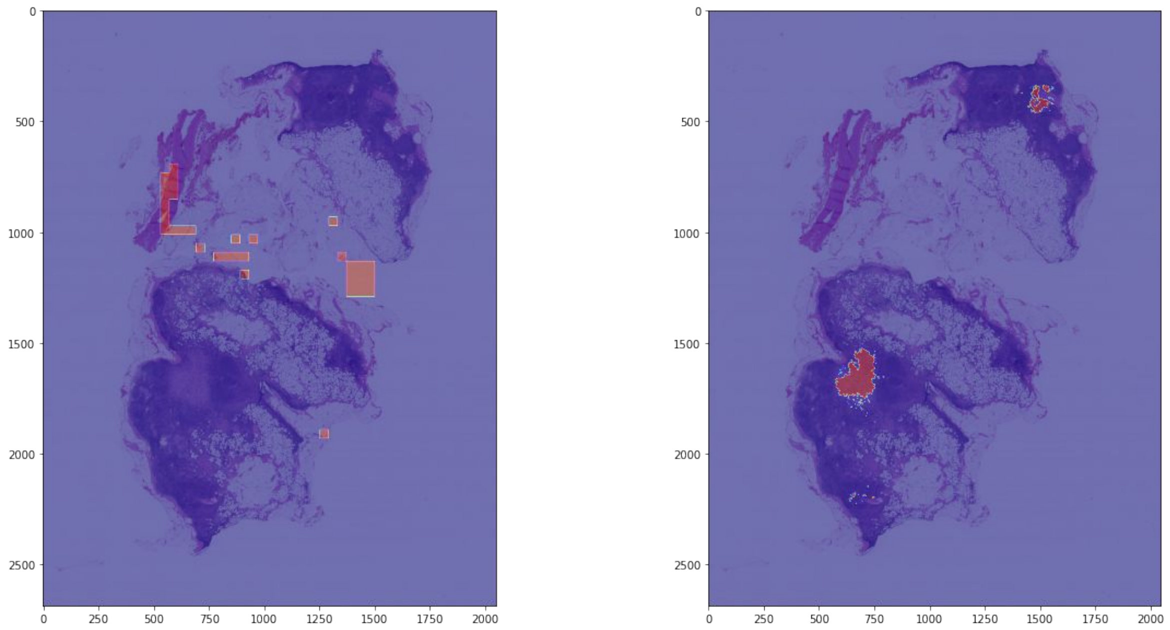


Three-zooms (567) Model

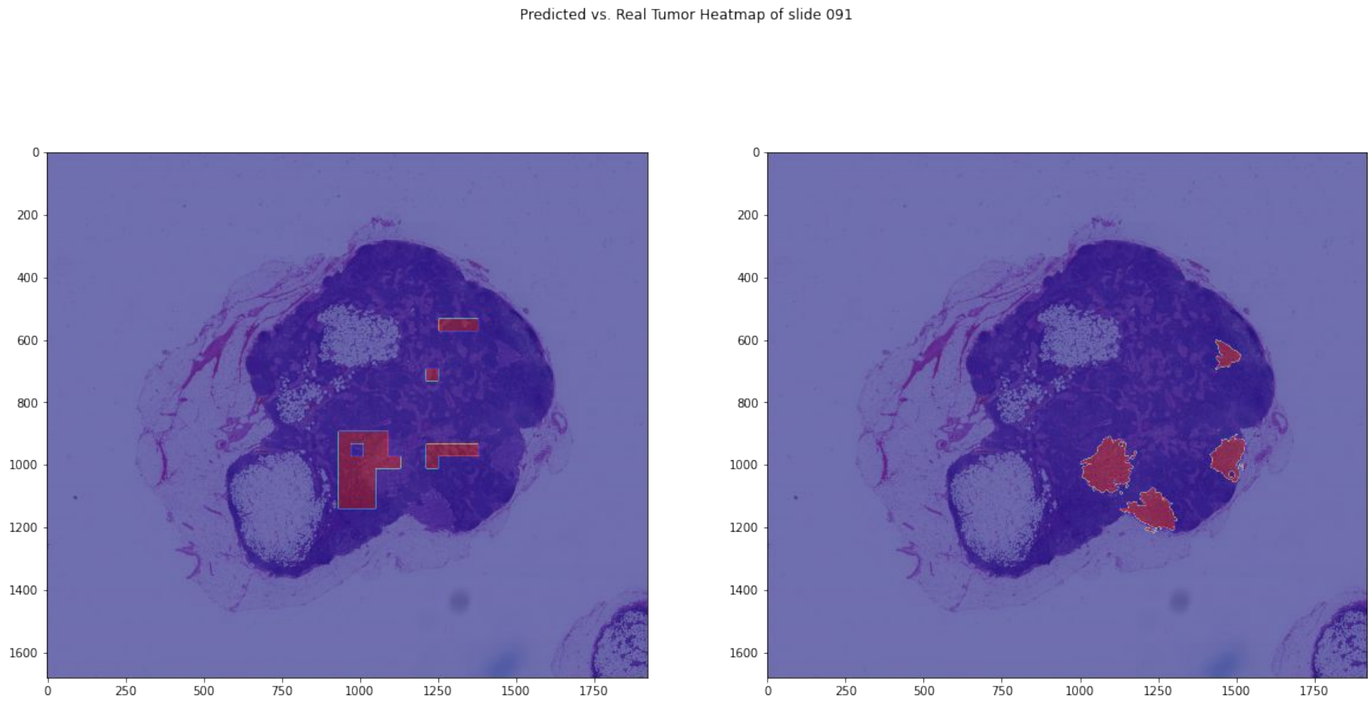


Three-zooms (567) Model

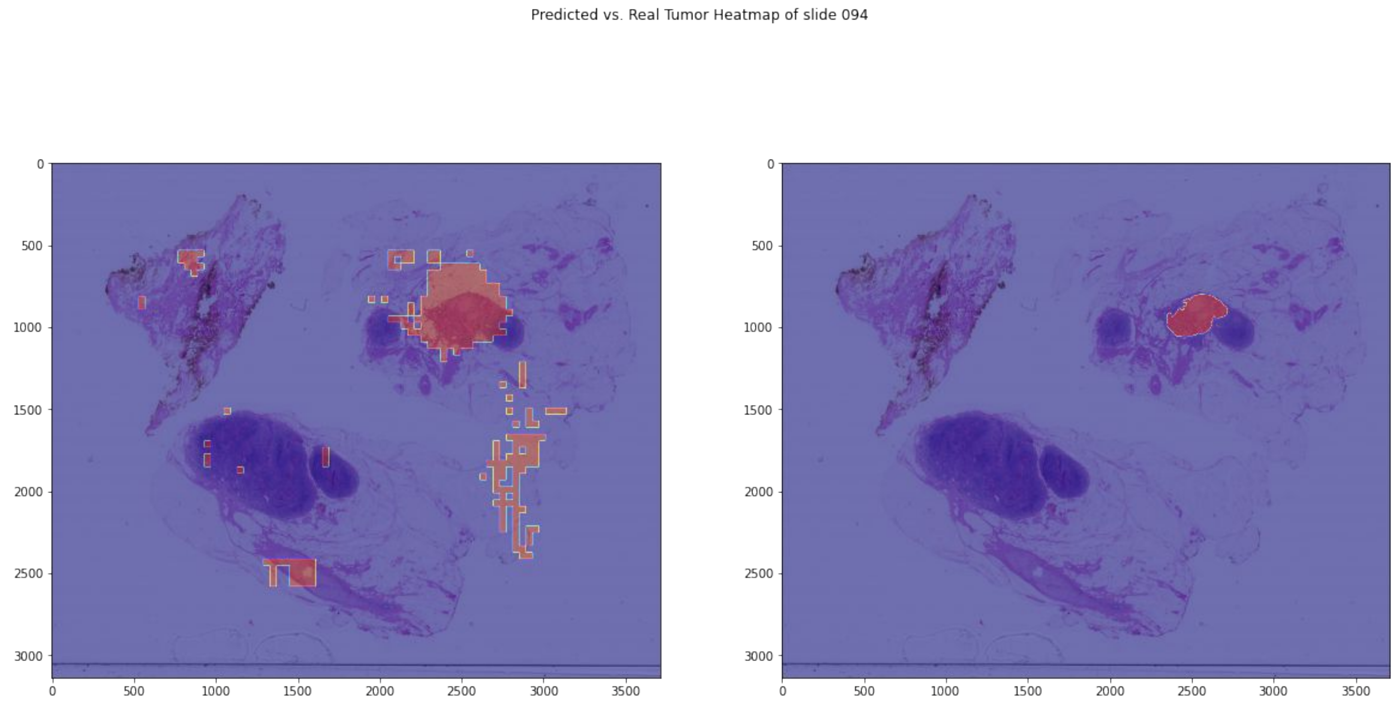
Predicted vs. Real Tumor Heatmap of slide 084



Three-zooms (567) Model

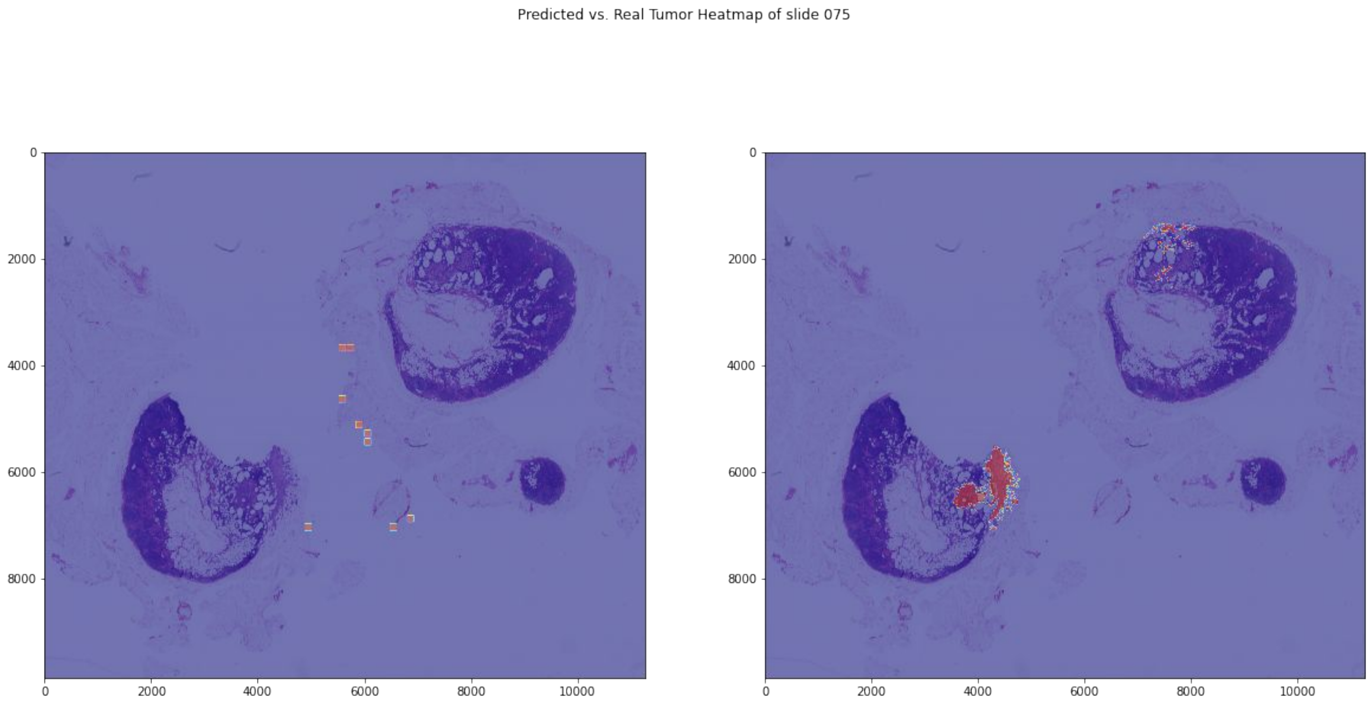


Three-zooms (567) Model



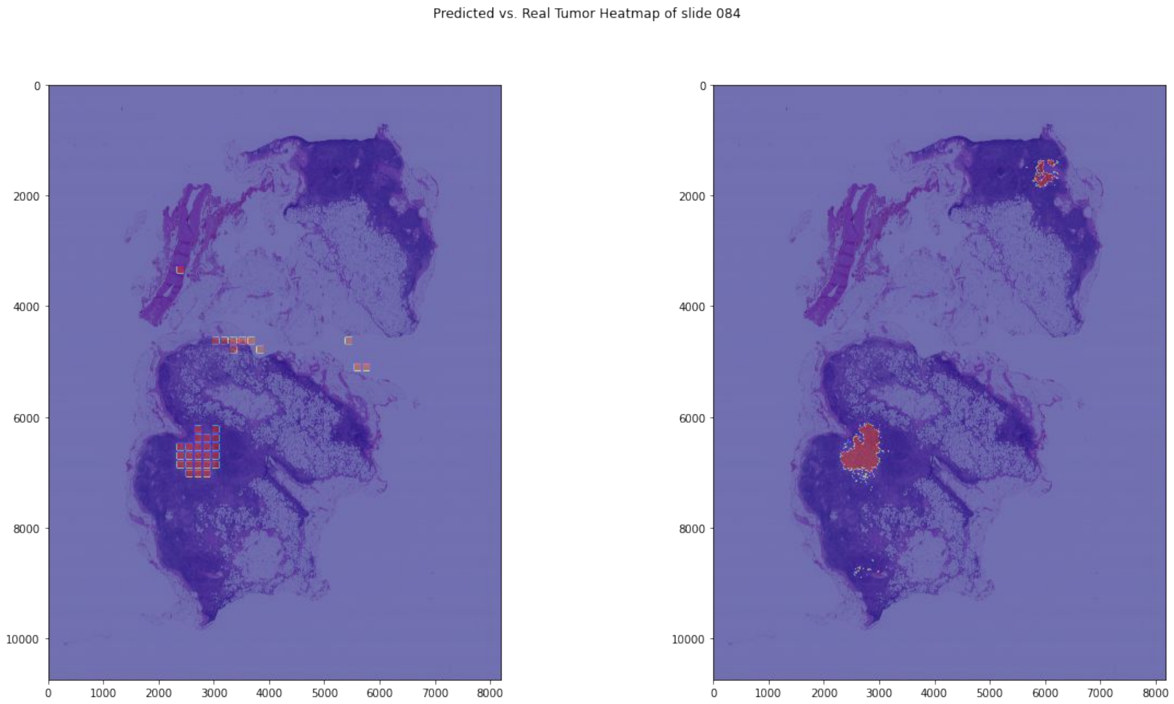
Columbia University
Data Science Institute

Three-zooms (357) Model



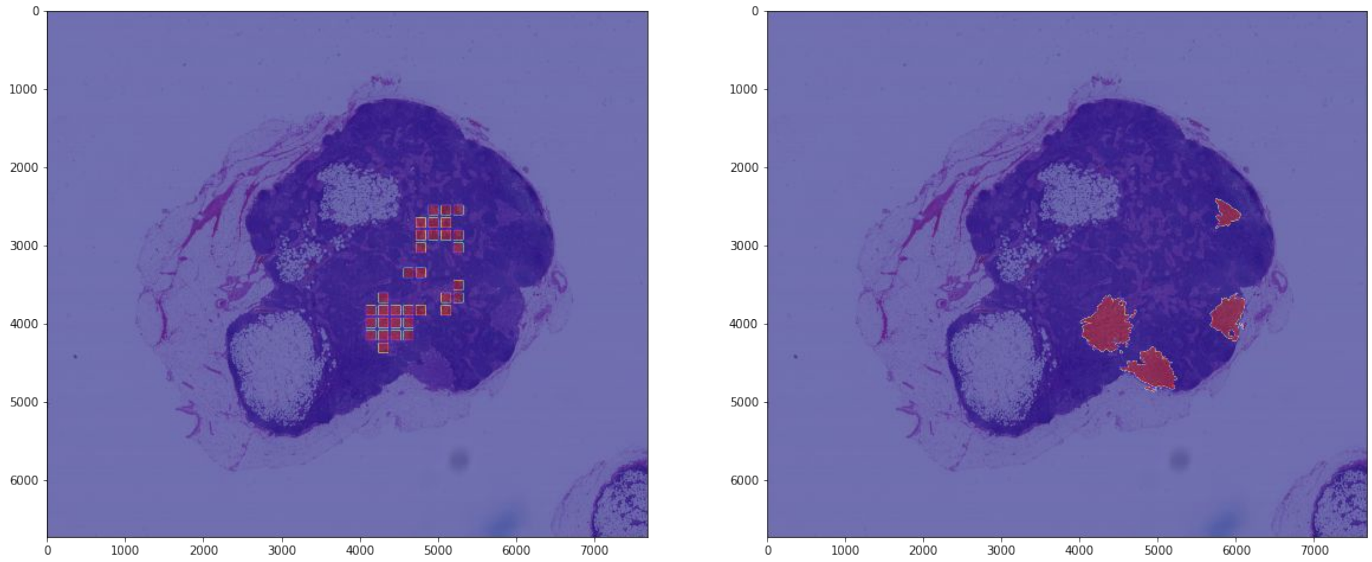
COLUMBIA UNIVERSITY
Data Science Institute

Three-zooms (357) Model

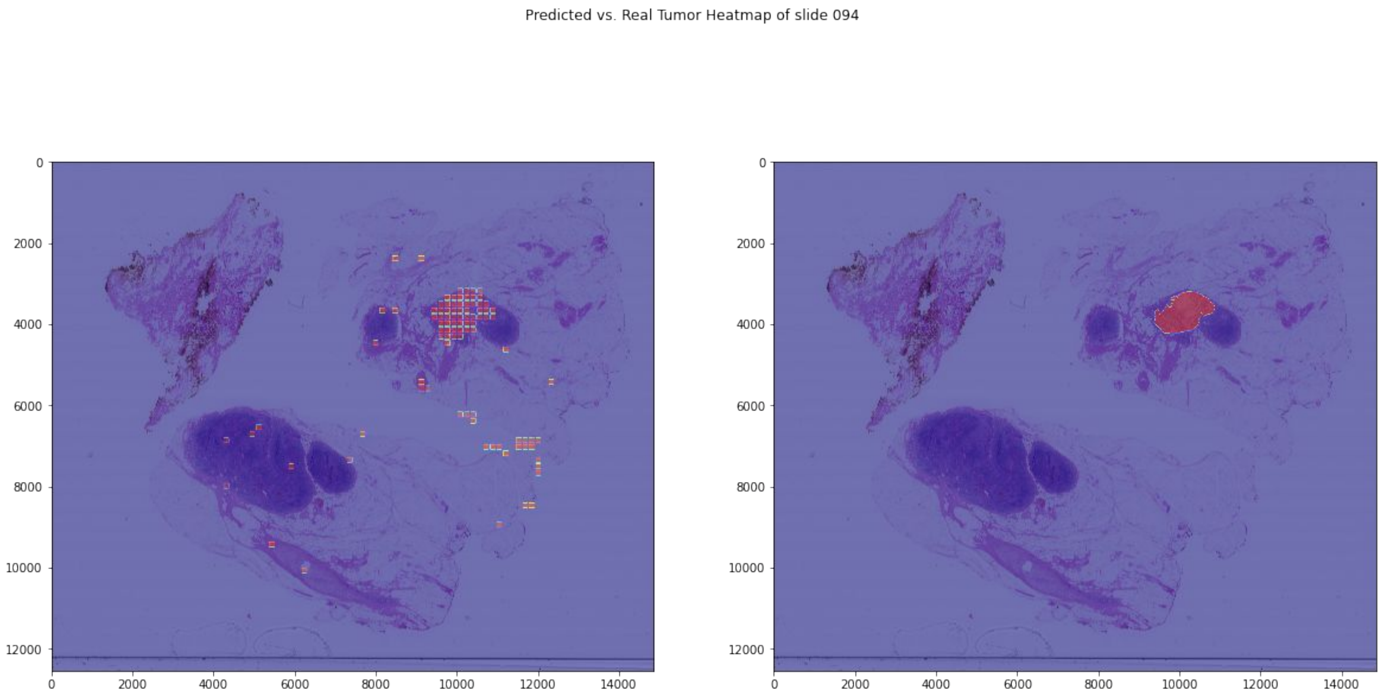


Three-zooms (357) Model

Predicted vs. Real Tumor Heatmap of slide 091



Three-zooms (357) Model



Conclusions

- Build entire pipeline for cancer tumor detection.

- Package common functions to make experiments much easier.[1]

- Combination of different zoom levels are helpful to detect the cancer tumor.

- The interval of levels is significant to the model performance.

- Transfer learning greatly improve the efficiency and accuracy of image recognition.

[1] All code are in the Github repository. https://github.com/Alex2Yang97/Cancer_Tumor_Detection_on_Gigapixel_Pathology_Images

Future Work

- Better selection of samples.

- Tumor area is much smaller than the normal area, so selecting positive samples randomly is much easier to get similar pitches than selecting negative samples

- Try more combination

- Try different number of levels and different level intervals

- Improve heatmap generation process

- For each level, I need give model the full pitch of cells, so I need to depend on the max zoom level to get pitches but model will predict on the minimum level. It will make the heatmap has some gaps.