Cancer detection in gigapixel pathology images

Kassiani Papasotiriou

Introduction

Goal

- **Medical goal**: pathologist want to determine whether cancer has spread beyond a primary tumor into the lymphatic system

- **ML goal:** Create a model that will offer an automatic **second opinion** for the pathologist to determine whether cancer has spread or not

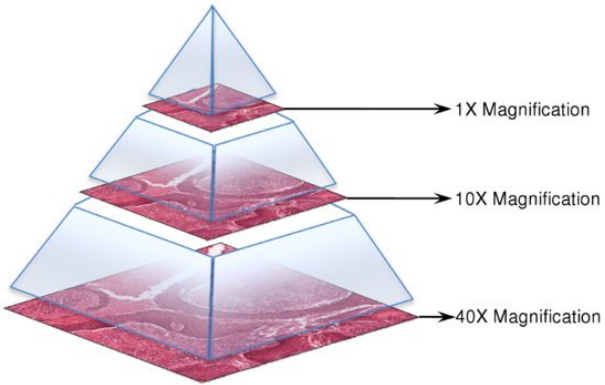
Previous work

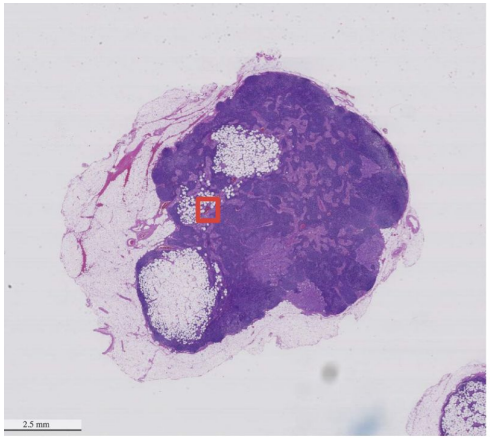
- Paper : Detecting Cancer Metastases on Gigapixel Pathology Images

- CNN architecture on the Camelyon16 dataset in the challenging lesion-level tumor detection task.

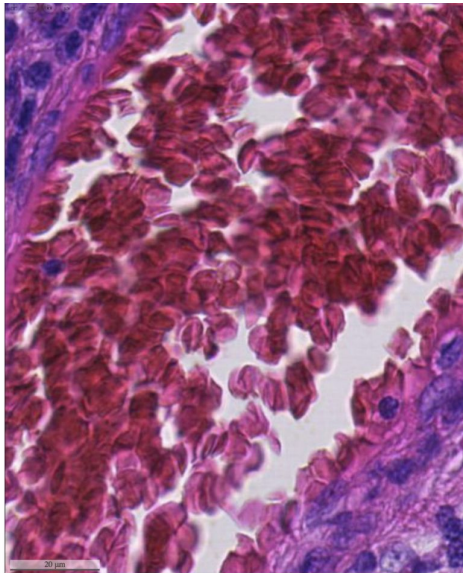
Data

Data - CAMELYON16:

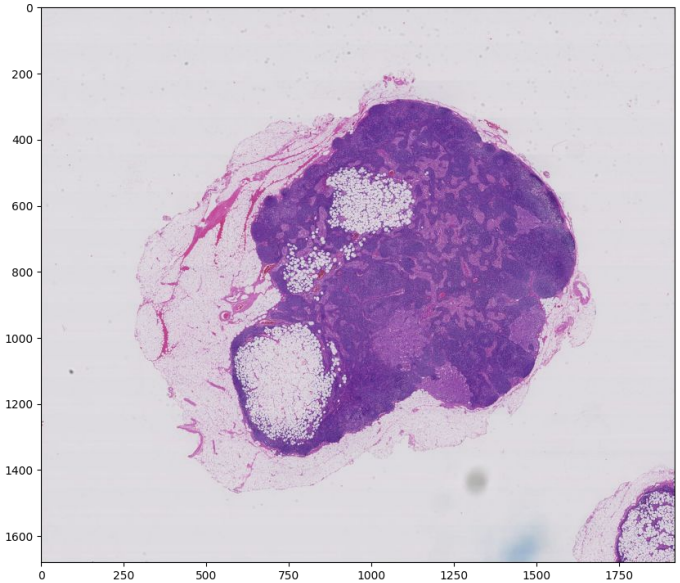


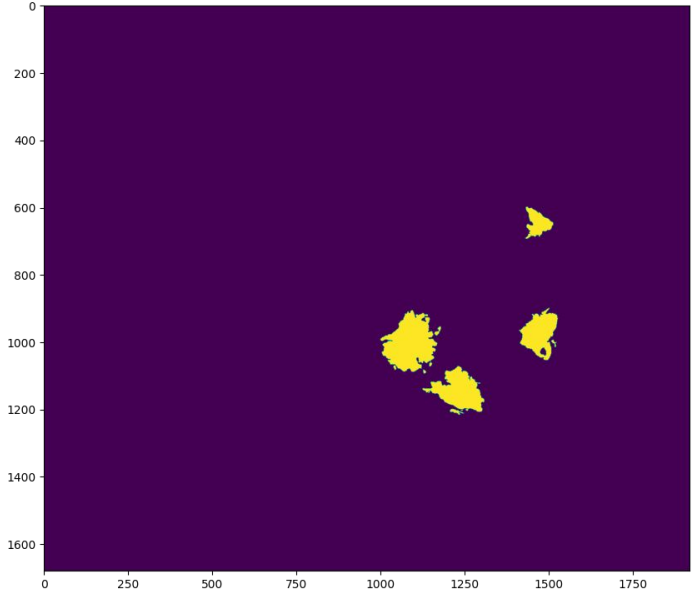


7 magnification levels available per slide, up to 128x.

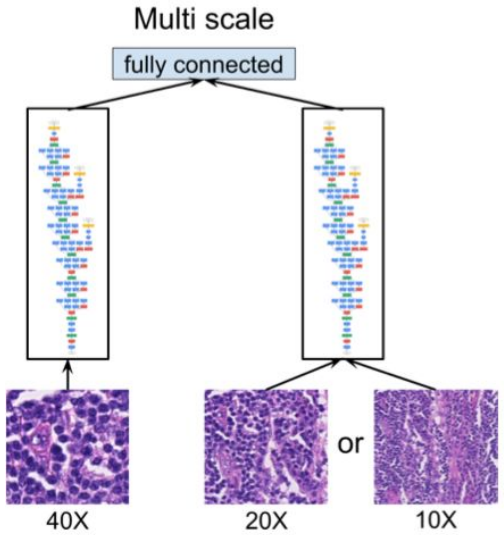


Data -Format


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

Model Overview



Patch generation

- Create two patches centered at the same part of the image. Use the mask of the most zoomed level to use as y label.

- **Challenge**: large number of patches of no important info. Very few pixels are marked as cancerous

- **Work around**: Find cancerous patches in a more targeted way

Data Augmentation

- Cancerous and non-cancerous patches don’t have a specific orientation in space
- We try two data augmentation methods

Models

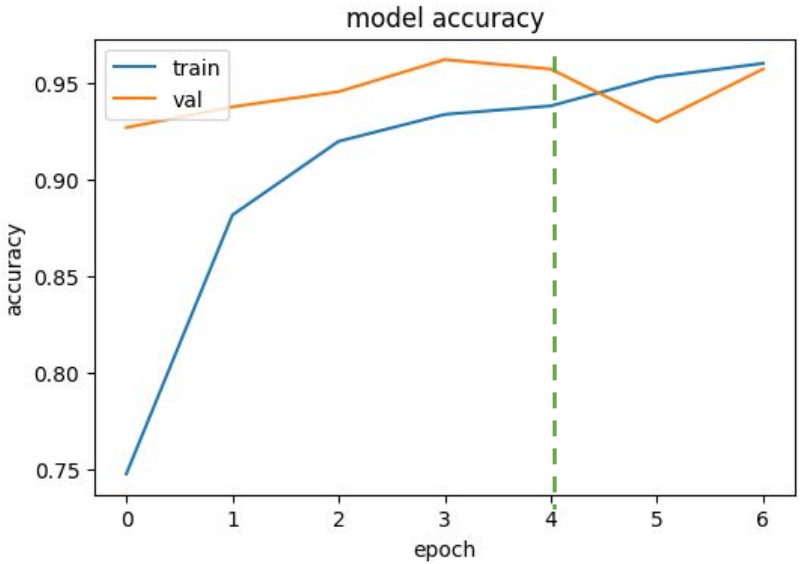
Model 1 - InceptionV3

- **Pretrained model** to have an initial sense of performance
- We don’t include the top/dense layer .
- **Two identical models** since we have two zoom levels
- We concatenate the output of the two models pass them through the final dense layer

Model 2 - Custom CNN

```
  tower1 = Sequential()
  tower1.add(Conv2D(128, kernel_size=(5, 5), input_shape=(input_size, input_size, 3),
padding='same'))
  tower1.add(BatchNormalization())
  tower1.add(Activation('relu'))
  tower1.add(AveragePooling2D(pool_size=(2, 2)))
  tower1.add(Conv2D(64, kernel_size=(5, 5), input_shape=(input_size, input_size, 3), padding='same'))
  tower1.add(BatchNormalization())
  tower1.add(Activation('relu'))
  tower1.add(AveragePooling2D(pool_size=(2, 2)))
  tower1.add(Conv2D(32, kernel_size=(5, 5), input_shape=(input_size, input_size, 3), padding='same'))
  tower1.add(BatchNormalization())
  tower1.add(Activation('relu'))
  tower1.add(AveragePooling2D(pool_size=(2, 2)))
```

InceptionV3 - Metrics Metrics

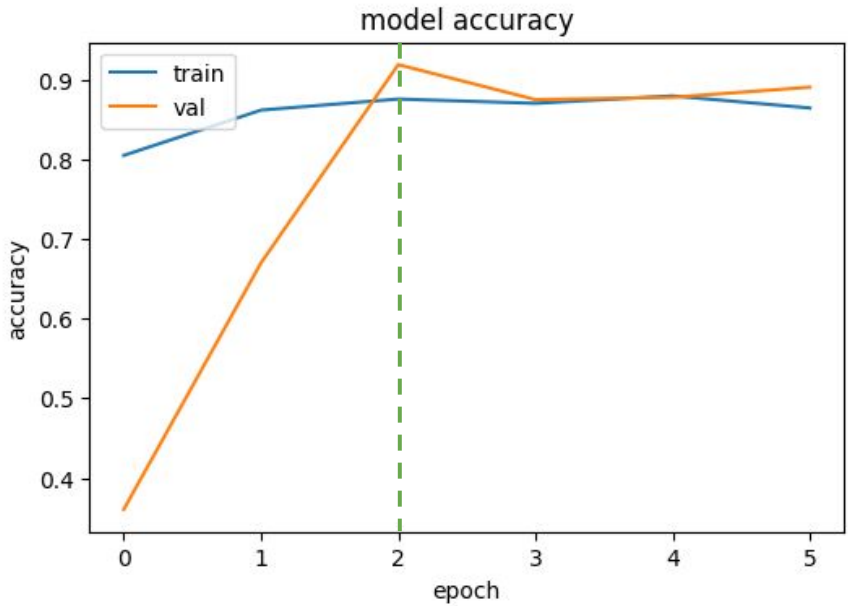


Train Accuracy: 0.9265

Validation Accuracy: 0.9482

Test Accuracy: 0.9001

CNN Metrics

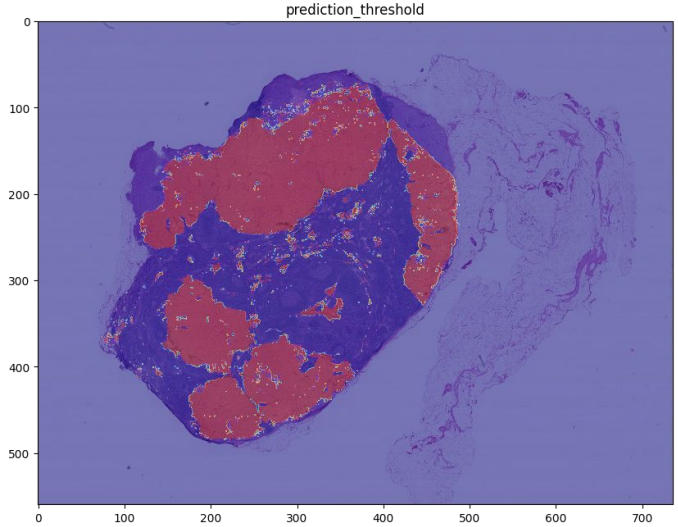


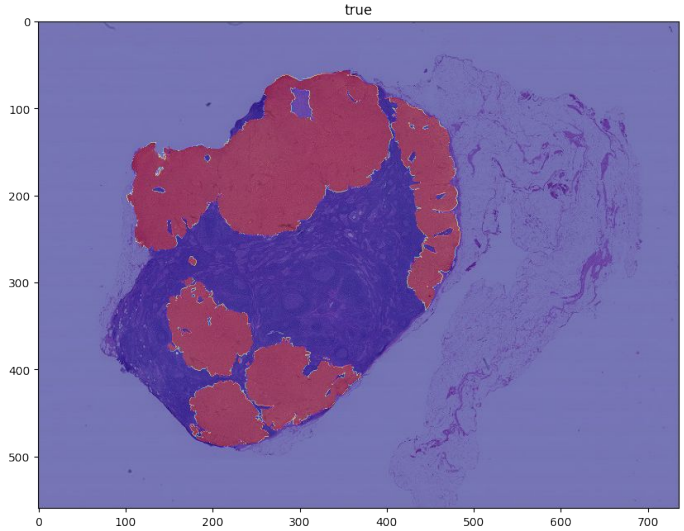
Train Accuracy: 0.8757

Validation Accuracy: 0.9189

Test Accuracy: 0.7992

InceptionV3- Mask Prediction





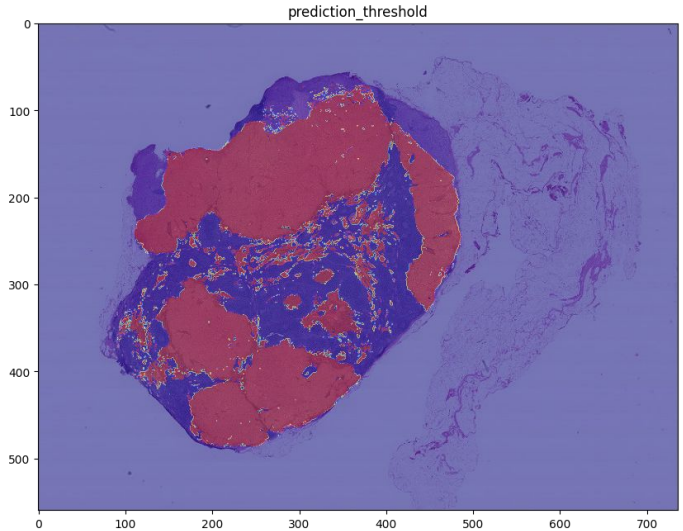
Accuracy: 0.958402

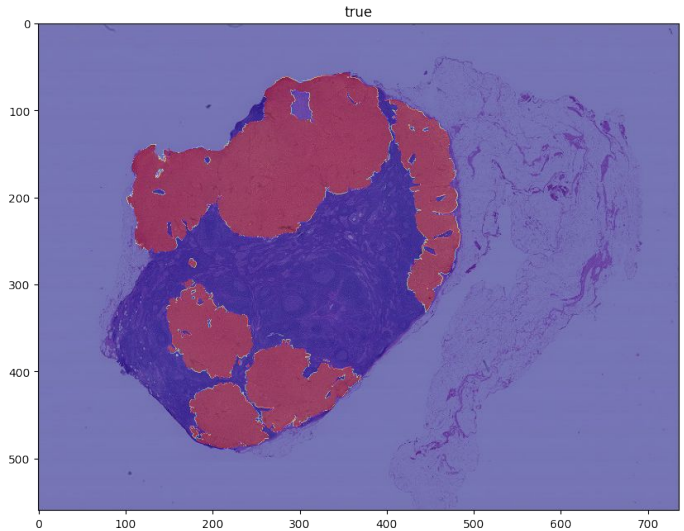
Precision: 0.914761

Recall: 0.808561

F1 score: 0.858389

CNN Mask Prediction





Accuracy: 0.939635

Precision: 0.777062

Recall: 0.859428

F1 score: 0.816172

Sources

Sources

https://camelyon16.grand-challenge.org/Data/

https://arxiv.org/abs/1703.02442

https://paperswithcode.com/paper/detecting-cancer-metastases-on-gigapixel#code

https://github.com/kira-95/adl_cancer_detection

https://arxiv.org/pdf/1712.04621.pdf

https://github.com/olahosa/adl_cancer_detection

https://www.pyimagesearch.com/2019/02/04/keras-multiple-inputs-and-mixed-data/