* Data: <https://camelyon16.grand-challenge.org/Data/>

( whole dataset 600GB, can use a small subset to get an A )

* Start with inception model
* <https://github.com/MohamedAliHabib/Brain-Tumor-Detection>
* Paper: <https://arxiv.org/pdf/1703.02442.pdf>

<https://winterwindwang.github.io/2021/04/12/gigapixel_processing.html>

参考别人代码：<https://github.com/jeswanthyda/Cancer-Cell-Detection-ADL/blob/main/final_project_jy3012.ipynb>

<https://github.com/AHHOZP/Applied-Deep-Learning/blob/main/project.ipynb>

https://www.youtube.com/watch?v=3mdOnemrOUw&t=137s

Your project should include

● Train a model using a subset of the training data from CAMELYON16.

● Include a script to run it on a testing image and generate a heatmap (note:

you do not need to create a 100,000 x 100,000 heatmap -- use a much lower

resolution).

Results

● Design a thoughtful evaluation method, and include the results on at least

three images from the testing set.

● Include a saved, trained version of your model with your submission.

Tips from the paper

● “To reduce computation, we removed background patches (gray value > 0.8 and verified visually that lymph node tissue was not discarded.”

● “We surprisingly found that slimmed-down Inception architectures with only 3% of the parameters achieved similar performance to the full version”.

● Authors found that using a pretrained model on ImageNet ultimately hurt accuracy (Why? Pathology images do not match the natural images from ImageNet) -- but -- it improved convergence speed. This is a good place to start. Try transfer learning

and/or fine-tuning.

Tips from Josh:

1. Implement an end-to-end prototype (training data in -> heatmap on a test image out), then slowly scale up.

2. Start by using one zoom level, with low resolution, a simple model, and a small amount of data.

3. Begin with transfer learning from a model trained on ImageNet, then consider training a model from scratch. Only

after this is working end-to-end, consider data augmentation, using higher magnification, multiple levels, etc.

4. Is Inceptionv3 still the best architecture to use?

Training set : total of 270 slides, but in each slide, we have multiple patches in different magnifications and cover different regions ?????

Multi-scale approaches

Preprocess:

Delete gray area

Remove background patches

Create dataset

Data upsampling/ downsampling

Data augmentation(rotate, change colors)

Chooses of different levels, Patch?

Sliding window patches or? ( using smaller image patches, sliding window, generate probability heatmap)

Idea:

Experient

Level 2 with size 75 \* 75

and level 5 diff size of sliding window to construct dataset

Resample ratio:

No tumor : tumor :-> 5:1

Visualization:

Tumor on tissue pics

In different levels, patches

Training:

pretrained weights ( Inception V3)

Add customized layers at beginning at end?

Tuning: learning rates, batch size…

Validation:

Visualize the training process and the loss.

Present the precision or recall metrix

One of the prediction, heatmap

Evaluation:

1. AUC but hard to compute as the amount of patches is large
2. FROC ???

Future

Train more appropriate models to make one ensemble model to make it more robust.