

# Jupyter Notebook Written Assessment 2022-23

Identifying different types of cell nuclei in cancer samples using 2 different ConvNets

## Getting Started

The coursework is set as a Kaggle Competition. Kaggle is one of the leading data science competition sites and it is worthwhile getting familiar with it.

If you are unfamiliar with Kaggle then you should probably get started by doing the Titanic Tutorial: <https://www.kaggle.com/alexisbcook/titanic-tutorial>

You can use your current Kaggle Username if you already have one. Alternatively, you can use a pseudonym for yourself when registering an account if you want to remain anonymous in the competition (it is a private competition for this course ... but the leader board is still publicly available).

Once you are logged into Kaggle then you can join the private "Deep Learning for MSc 2022-23" competition using the link give on the Moodle site.

You will need to choose a Team Name to take part in the Kaggle competition. Often Kaggle competitions are done in Teams but this will be **individual coursework** so you will only have yourself in your Team !

**IMPORTANT: Your Team Name should be made of characters that are acceptable as a filename. (Ideally you would use the underscore character instead of a space since spaces in filenames can cause so many problems !) The Jupyter Notebook file you submit to Moodle will use your Team Name as it's filename.**

## Overall Goal

Colon cancer is the 3rd most common cancer in the world.

Pathologists can take microscope slides of a colon biopsy and when dyes with particular chemicals they can form digitized images as shown below in Fig. 1a) (the normal colon has projections known as villi which absorb food and appear as circles of cells on a microscope image of the tissue since the villi are cut through sideways).

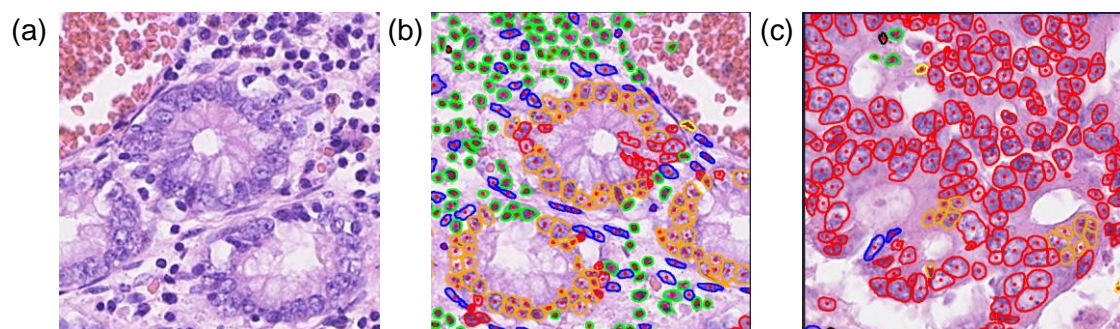


Figure 1: (a) Normal Colon Tissue. (b) Nuclei coloured by type with orange = normal, red = cancer, green = immune cells, blue = connective tissue. (c) Cancer sample.

The cell nuclei in these images can be segmented and coloured according to the different cell types as shown in Fig. 1b). In worse grades of colon cancer, the circle of normal cells disintegrate and the nuclei become larger and less well formed (due to DNA damage) as seen in Fig. 1c).

Digital pathology aims to apply machine learning to digital pathology images like these to determine if the tissue is normal or cancerous. One approach is to segment out the nuclei of cells from these images and classify them into different cell types including malignant cell nuclei (i.e. cancerous cells).

So your overall goal for this data science coursework is to train **two** deep neural network which can take a 100x100 pixel images with a cell nuclei in the centre of the image and classify it into one of the following types which are shown in the figures above:

1. Normal epithelial cell nuclei with label 0.
2. Cancer epithelial cell nuclei with label 1.
3. Muscle cell nuclei with label 2.
4. Immune leukocyte cell nuclei with label 3.

The “Data Tab” on the Kaggle site allows you to browse the available data. You should use this “Data Tab” to browse through the data so you understand what it is like. You will find a train.zip file which contains a large collection of 100x100 pixel images of cell nuclei (you can browse through these on the data tab). It also contains a train.csv file giving the image filename and its group truth label as given above. The test.zip file contains similar 100x100 pixel images of unlabelled cell nuclei. You need to predict the label of these test nuclei and submit these using a “submission.csv” file. (There is an example of the submission file format given as example.csv where all the labels are predicted class 0 - clearly your system needs to predict the correct labels!)

[How should I develop my code \(and where do I get the GPU/TPU power from ?\)](#)

So far you have mostly used the Google Colab Notebook for Labs but this would involve transferring fairly large data files and also you may find that you run out of GPU time on Google Colab (particularly if you are doing other coursework using it).

For this coursework we will use Kaggle Notebooks ! This not only gets you familiar with another Jupyter Notebook system – but it means you can directly access the data files for this competition without transferring them. It also allows you to keep your files in order since it has a Versioning system built in (it is important that you submit the same notebook as you used to generate predictions for your best attempt at Kaggle and this can be difficult unless you keep track of what versions of the notebook you used for different submissions).

**For this coursework you will be using Kaggle Notebooks. This has the advantage of direct access to the data but also means that everyone has the same access to GPUs and TPUs (making it fairer). Kaggle gives you 30 hours of GPU or TPU time each week which should be easily sufficient for this coursework given reasonable use.**

If you go onto the “Code Tab” then do “New Notebook” – this will create a competition notebook where you have direct access to the competition data. Please see the Kaggle Notebook documentation for more information about their notebook system:

<https://www.kaggle.com/docs/notebooks>

You use GPUs in the same manner as you would with normal PyTorch code (you need to turn them on though in a similar manner to Google Colab). If you want to experiment with using TPUs then a good getting started course is:


<https://www.kaggle.com/competitions/tpu-getting-started>


With a more specific tutorial on using TPUs with PyTorch given here:

<https://www.kaggle.com/code/tanlikesmath/the-ultimate-pytorch-tpu-tutorial-jigsaw-xlm-r/notebook>

### Steps to Success !

This is very much a “Capstone” project where you will bring together a lot of the material you have understood from different labs and lectures.

After getting familiar with the Kaggle infrastructure – the first stage of Notebook development would be to write a custom data loader for this nucleus and csv data in a similar manner to Lab 5. 

You first want to make sure you understand the key ideas in Lecture 4 – Machine Learning Workflow. A lot of these concepts will be essential in terms of splitting the data into suitable training and validation datasets (you should be evaluating your performance using the validation dataset and not relying on resubmission to Kaggle to assess your performance – you are only allowed 5 submissions per day – and doing more will result in overfitting to the test set). 

**Then the first stage would be writing a custom data loader for this particular data similar to Lab 5. What aspects of the data might you need to handle to get an accurate classifier?**

There is a cell nuclei in the middle of each image of a particular type. Your overall goal is to develop a deep learning method that can predict the cell type of the images in the test.zip file. You can make multiple submissions each day to the Kaggle site and they will be scored using a portion of the test data. (Testing against the final test data to get your overall score is only done at the end of the competition – this is when you find out if you have overtrained the model as your rank on the leaderboard goes down!)

**You need to develop **two models** in your notebook:**

1. “model1” should be your own small ConvNet with no more than 8 layers total. You should train this from scratch yourself. Lab 3 discusses how you should develop such a model.
2. “model2” should be an existing torchvision model which is pre-trained on ImageNet. You will need to modify this model for your particular task and then train it on the nucleus data – essentially using transfer learning. This was done for example in Interpretation Lab 6.

**For each model you should demonstrate carrying out some suitable hyperparameter optimization using Ray Tune as in Lab 5. Clearly you need to choose a sensible approach to hyperparameter optimization given your limited Kaggle GPU/TPU resources.**

For each model you should look at producing loss and accuracy curves to assess how your training is working and diagnosing any issues.

For each model you should produce a confusion matrix to understand what classes may be getting confused in terms of prediction and try to do something to resolve any confusion.

For each model you should try to interpret the model using Captum to understand how it is working as you did in Lab 6.

Clearly you should also submit your predictions to Kaggle for each model and determine which of them might be doing best (usually this is done by creating and submitting a submission.csv file). This can be done directly from the "Output" directory for Kaggle Notebooks.

### Submission

Please submit the results of your method (submission.csv) to the Kaggle site. This will be tested and the accuracy on the unseen test set will be given on the leader board. You can use a pseudonym on the Kaggle site if you do not wish to reveal your real name to be revealed in this competition. You can make up to 5 submissions per day to assess what might be the best approach. The leader board table will show the score of your best submission **and this will constitute 50% of the marks** (this will be based on getting a score better than particular thresholds rather than a direct conversion of the accuracy score !)

**IMPORTANT - also submit your final Jupyter Notebook for your results to Moodle (exported from Kaggle Notebooks system).**

Your Jupyter Notebook will be marked on a number of aspects such as showing the key components mentioned above (use of training and validation data, plotting and interpreting loss curves, hyperparameter tuning, showing a confusion matrix, visual interpretation of your networks and discussion of both your models in terms of these aspects). Your Jupyter Notebook file should be well annotated as a data science laboratory notebook – explaining what you are doing and why, interpreting your results and what they mean. **Your submitted notebook needs to have all the cells run so that they are all showing output to get you marks!** The submitted Notebook will constitute the other 50% of the coursework marks.

**Again - your submitted Jupyter Notebook should have all output visible so it can be read as a data science notebook.**

**IMPORTANT:** The notebook submission **filename** should be **identical** to your **Kaggle Team Name**. So if you are using a Kaggle Team Name "Dogs\_Versus\_Cats" - then you should submit your Jupyter notebook file to Moodle as "Dogs\_Versus\_Cats.ipynb" file. (There is no need to identify yourself in this file since Moodle will associate the file uploaded with your Moodle user account.)

***WARNING ! Failure to properly identify your filename as your Kaggle Team Name (or not to upload a Notebook File) will mean that you do not get any marks for your Kaggle submissions since I will not be able to identify your Team!!!***

PLEASE USE THE **TEAMS JUPYTER WRITTEN COURSEWORK CHANNEL** TO CLARIFY ANY INFORMATION ABOUT THE COURSEWORK.