Technical Project (Medical Imaging): Building an AI-based Brain Tumour Classification/Segmentation Models

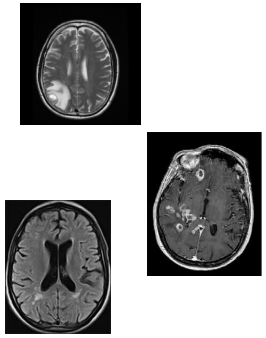
The project focuses on designing reliable AI-based brain tumour (gliomas) classification/segmentation models with the knowledge the students learnt from ‘The world of Artificial Intelligence’ lectures.

Gliomas are the most common malignant brain tumours causing significant mortality and morbidity. Accurate detection of brain tumours has always been a real-world challenge with great clinical importance.

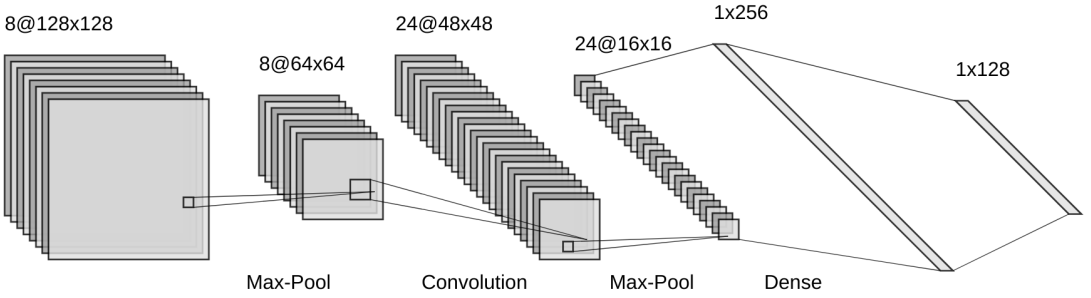
The students will work in small groups to build Convolutional Neural Network (CNN)-based AI models to detect gliomas on brain MR images.

CNN-based Framework for Brain Tumours Classification

- The students are expected to design a framework as given in the figure.



Input



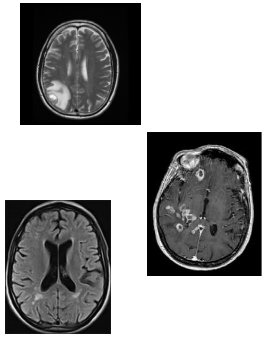
CNN-based model for feature extraction

With tumour or without tumour?

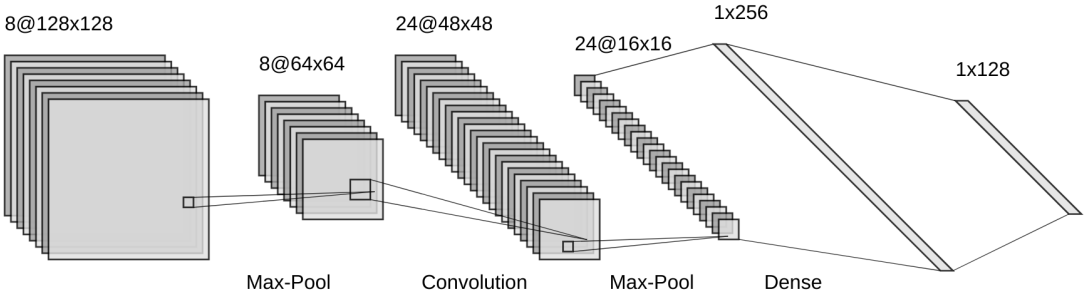
Output

CNN-based Framework for Brain Tumours Segmentation

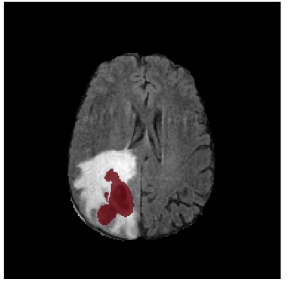
- The students are expected to design a framework as given in the figure.



Input



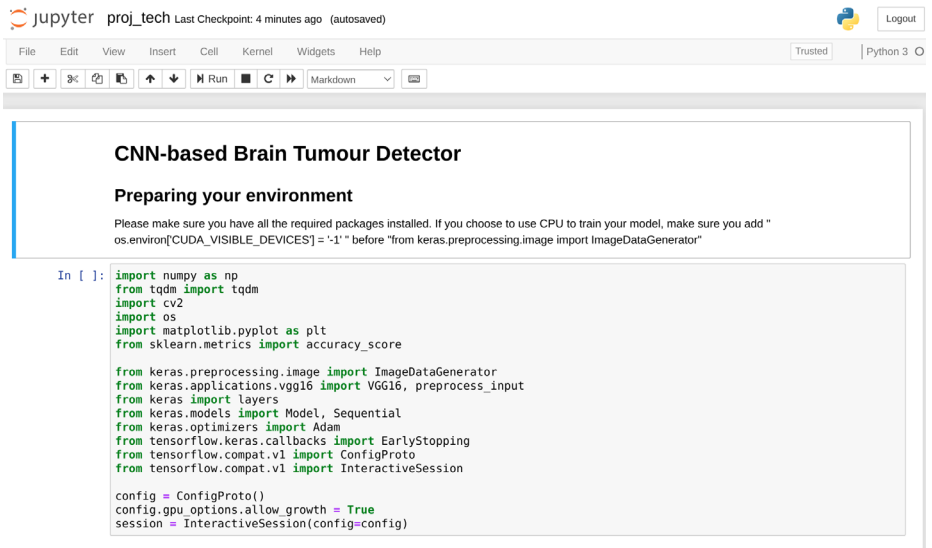
CNN-based model for feature extraction



Output

CNN-based Framework for Brain Tumours Classification

The students are provided with the dataset and sample code for realising the framework.



Sample code will be provided in Jupyter notebook

The students are able to modify the code to adopt different deep learning models, hyperparameters and data augmentation methods to achieve the best accuracy of classification.

CNN-based Framework for Brain Tumours Segmentation

Data preprocessing

Images in the original dataset are usually in different sizes, so sometimes we need to resize and normalise (z-score is commonly used in preprocessing the MRI images) them to fit the CNN model. Depending on the images you choose to use for training your model, some other preprocessing methods. If preprocessing methods like cropping is applied, remember to convert the segmentation result back to its original size.

In [ ]:

Train-time data augmentation

Generalizability is crucial to a deep learning model and it refers to the performance difference of a model when evaluated on the seen data (training data) versus the unseen data (testing data). Improving the generalizability of these models has always been a difficult challenge.

Data Augmentation is an effective way of improving the generalizability, because the augmented data will represent a more comprehensive set of possible data samples and minimizing the distance between the training and validation/testing sets.

There are many data augmentation methods you can choose in this projects including rotation, shifting, flipping, etc.

You are encouraged to try different augmentation method to get the best segmentation result.

Get the data generator ready

In [ ]:

Define a metric for the performance of the model

Dice score is used here to evaluate the performance of your model. More details about the Dice score and other metrics can be found at https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2

In [ ]:

Evaluation Criteria

- Medical imaging project: 40%
  - Accuracy of the final segmentation model on the test set: 25%
  - Rationale of the experimental design and final report: 15%
- Natural language processing project: 40%
  - Completeness of the project: 30%
  - Final report: 10%
- Presentation skills and team work: 20%

The students will demonstrate their frameworks on Monday 20th August.

Result Submission

On 19th August, you will be given the test set for brain tumour segmentation task. Use your trained model to generate the predicted segmentation map and upload your result in a .zip file on Teams by midnight.

We will let you know the performance of your model by noon 20th August.

Technical Report

Literature review on deep learning for medical image segmentation (Due by 12th August)

Experimental design

Results (Due by 19th August)

Setting Up Python Development Environment

Anaconda Installation (2020.02)
https://repo.anaconda.com/archive/

For using Nvidia GPU
https://github.com/antoniosehk/keras-tensorflow-windows-installation

Packages:
Tensorflow