**Artificial Intelligence Course Project - PatchCamelyon**

Hao Weining 1002195

Du Li 1002187

Wang Shunqi 1002191

Yin Yuanzhi 100 2206

**Data Pre-processing:**

Since the tumor will always appear at the center 32x32 region, we decided to crop out that part and pass it to the neural network. However, we decided to set the crop size to 40 in order to include a bit of surrounding context.

After the crop, we resized the image to 224 to match the input size of our selected model(resnet18). And finally we normalized the image for model input.

**Data augmentation:**

Since the pre-trained models on PyTorch are all relatively large, we decided to apply multiple data augmentation methods to reduce the level of overfitting. The detailed augmentation methods are as follows:

Random horizontally flip the image with probability 0.2

Random vertically flip the image with probability 0.2

Random rotate the image with the maximum degree being 20.

Random shift the image with the maximum distance being 10% of the original image size.

Randomly change the color of the image, including brightness, contrast, saturation and hue.

**Model Selection:**

Since the number of images that we have for training is relatively small, it is best to use a light weight model for this task to reduce overfitting. At first, we used the pre-trained model Resnet18 and changed the last layer of the model to 2 outputs. After that, we trained all the parameters for at most 50 epochs.

The training process will be terminated if there is enough sign of overfitting. E.g. when training accuracy is significantly higher than validation accuracy.

**Hyperparameters:**

After several rounds of tuning the hyperparameters based on the results of validation accuracy and loss, we finally decided to use these hyperparameters as below:

Loss function: Cross entropy loss

Learning rate: 0.0001

Optimizer: Adam

Weight decay: 0.0001

Number of Epochs: 50

**Performance:**

Our best validation accuracy is 0.8466796875. And our accuracy and loss curves of training and validation are shown below:

A screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generated

Using our model which we use to get the highest validation accuracy, we achieved a testing accuracy of 0.8277587890625.

**Graphics User Interface:**

We used the PyQt5 python package as our tool to build the GUI. By this GUI, user can select image file from the user’s computer, and it will then show the image selected as well as give the prediction based on the image, either ‘No Tumor’ or ‘Contains Tumor!’. Here is a sample screenshot of our GUI:

A screenshot of a cell phone

Description automatically generated

**What to Install:**

In order to run our codes, both the training model and the GUI, user has to install these packages: torch, torchvision, PIL, numpy, PyQt5, h5py, all to the latest version.

**Team Contributions:**

Hao Weining: model implementation;

Du Li: Data preparing and augmentation;

Wang Shunqi: Graphics User Interface;

Yin Yuanzhi: fine tuning and testing.