Brain Tumour Classification

This document details the steps needed to run the code for carrying out the task of classifying Brain Tumour MRI images into Malignant and Benign.

Getting Started

Dependencies

The libraries can be installed in the Juypter Notebook.

Libraries needed:

- scikit-learn
- tensorflow
- opencv-python
- matplotlib
- pydot
- keras-tuner
- Pillow
- seaborn

Softwares needed:

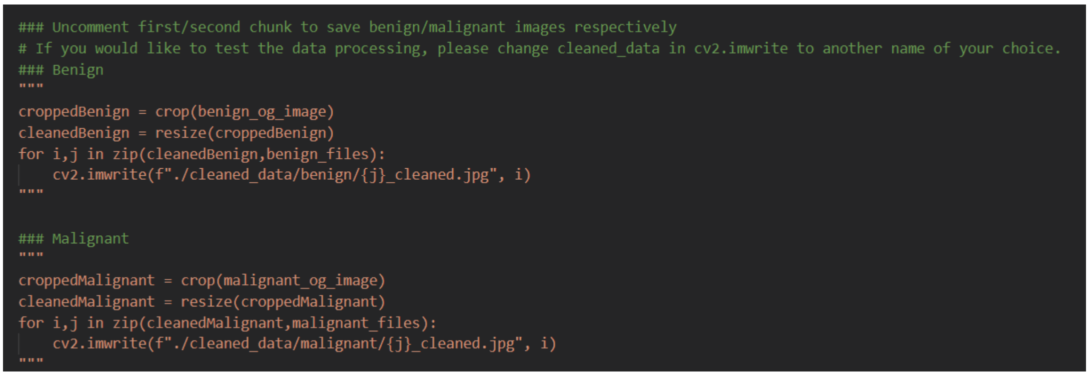
- Graphiz. Install Graphiz here: https://graphviz.org/download/

Installing

- Under the libraries code block, uncomment the code block to install the libraries. Once done, the code block can be commented again.
- Graphiz is optional to install. It used to visualise the Convolutional Neural Network (CNN) model layers.

Executing program

- The first few codes will be regarding the libraries. Uncomment to install libraries and comment them again once all the required libraries are installed. Run the import libraries code to ensure that all libraries are available for use.
- For the Data Processing code block, if you would like to test the data processing, please delete all the cleaned images in **cleaned_data/benign and cleaned_data/malignant** before testing.

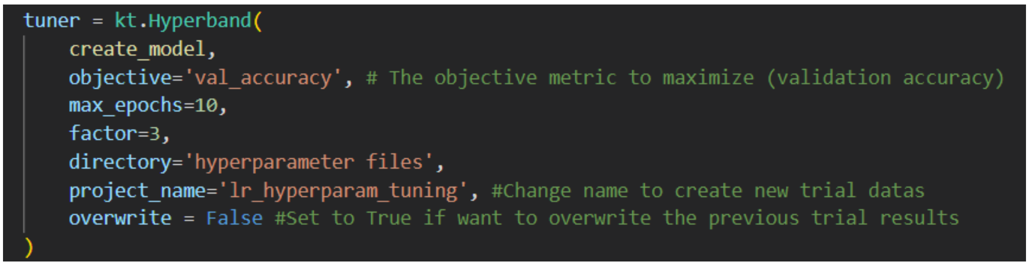


```
### Uncomment first/second chunk to save benign/malignant images respectively
# If you would like to test the data processing, please change cleaned_data in cv2.imwrite to another name of your choice.
### Benign
"""
croppedBenign = crop(benign_og_image)
cleanedBenign = resize(croppedBenign)
for i,j in zip(cleanedBenign,benign_files):
    cv2.imwrite(f"./cleaned_data/benign/{j}_cleaned.jpg", i)
"""


### Malignant
"""
croppedMalignant = crop(malignant_og_image)
cleanedMalignant = resize(croppedMalignant)
for i,j in zip(cleanedMalignant,malignant_files):
    cv2.imwrite(f"./cleaned_data/malignant/{j}_cleaned.jpg", i)
"""
```

- After that you can run each code block one by one, the code for logistic regression and KNN has no special exceptions.
- CNN code can be run one block at a time with some exceptions.

- Under the visualising the model section in the CNN code, if graphiz is not installed, leave the code commented and move on to the next block.
- Under the hyperparameter tuning in CNN, do not change overwrite to True. If you want to test the hyperparameter tuning, create a new file by making a new project_name.



```
tuner = kt.Hyperband(
    create_model,
    objective='val_accuracy', # The objective metric to maximize (validation accuracy)
    max_epochs=10,
    factor=3,
    directory='hyperparameter files',
    project_name='lr_hyperparam_tuning', #Change name to create new trial datas
    overwrite = False #Set to True if want to overwrite the previous trial results
)
```

- Other than that, all code can be run one by one in order.

Authors

IT1244 Group 41

- Lim Zhi Chao
- Ang Qi Wei Chester
- Faith Arielle Peh
- Tan Ze Qi Christian Lambayong

Acknowledgments

Codes for CNN were inspired from here:

- https://www.tensorflow.org/tutorials/images/transfer_learning#use_data_augmentation
- https://www.tensorflow.org/tutorials/keras/keras_tuner